People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research
Kasdi Merbah University OUARGLA ALGERIA
Faculty of New Information and Communication Technologies
Department of Computer Science and Information Technology

Field: Mathematics and Computer Science
Track: Computer Science
Specialty: Theoretical Computer Science
Academic Master's Thesis

Presented by:

- Tadjine Rida
- Abdeljouad Mohammed Aimen

REAL-TIME TASKS ALLOCATIONS BASED ON HHO FOR FLYING AD HOC NETWORKS

Public debate on: 2023-2024

The committee members:

Dr. Benkaddour Kamel	Examiner	UKM Ouargla
Dr. Abdelatif Cheradid	Supervisor	UKM Ouargla
Dr. Kahlessenane Fares	Examiner	UKM Ouargla

ACADEMIC YEAR: 2023 / 2024

شكر و إهداء

بسم الله الرحمن الرحيم, والصلاة و السلام على اشرف المرسلين سيدنا وعظيمنا و قائدنا و قدوتنا محمد عليه افضل الصلاة و أزكى التسليم, اما بعد فنبدأ بشر الله عز و جل على ان وفقنا لهذا و سدد خطانا طيلة أعوام الدراسة و نسأله أن يعلمنا ما ينفعنا و ينفعنا بما علمنا و نسأله أن يزيدنا علما, الحمد لله ملئ السماوات و الأرض الحمد لله عدد خلقه ورضا نفسه وزنة عرشه الحمد لله دائما و ابدا, ونشكر والدينا و إخواننا الذين كافحو وإجتهدو وضحو بكل ما هو ثمين لأجلنا و لأجل ان يروننا على ما نحن عليه اليوم و أصدقائنا الذين كانو بمثابة العائلة, نشكر كل شخص ساندنا ولو بكلمة, كما نشكر اساتذتنا الكرام الذين قامو بتدريسنا ولا ننسى استاذنا الفاضل الدكتور شيراديد عبد اللطيف الذي قام بالإشراف علينا في إنجاز مذكرتنا و نبدي له كل الثناء على إرشاداته و توجيهاته لنا طيلة وقت إنجازنا لهذه المذكرة كما نشكر اعضاء لجنة التحكيم على قبولهم تقييم عملنا هذا, نسأل الله العظيم ان يبارك للجميع وان يثيت خطانا جميعا و إلى الله ترجع الأمور.

Abstract

In recent times, the use of drones has become powerful across various fields such as transportation, military applications, photography, and more. These unmanned aerial vehicles (UAVs) prompt us to consider enabling data exchange and communication among them to enhance task performance. This concept leads us to Flying Ad Hoc Networks (FANETs), which involve creating wireless networks among flying nodes like drones.

However, there are several challenges associated with this technology, including limited battery life and the distance between drones. To address these challenges, researchers propose leveraging the Harris Hawks Optimization (HHO) algorithm. Inspired by the collaborative flying strategy of Harris's Hawks, the goal of this research is to optimize FANET topology to achieve the maximum number of tasks performed by these drones while incorporating the behavioral characteristics of Harris Hawks through this optimization process.

Key Words: Drones, Wireless Networks, Flying Ad Hoc Networks (FANETs), Battery Life, Harris Hawks Optimization (HHO), Tasks.

الملخص

في الآونة الأخيرة، أصبح استخدام **الطائرات بدون طيار** قويًا في **مجالات مختلفة** مثل النقل والتطبيقات العسكرية والتصوير الفوتوغرافي وغير ذلك الكثير. تدفعنا هذه المركبات الجوية بدون طيار **(UAVs)** إلى التفكير في تمكين تبادل البيانات والتواصل فيما بينها لتعزيز أداء المهام. يقودنا هذا المفهوم إلى **الشبكات الطائرة المخصصة (FANETs)،** والتي تتضمن إنشاء شبكات لاسلكية بين العقد الطائرة مثل الطائرات بدون طيار.

ومع ذلك، هناك العديد من التحديات المرتبطة بهذه التكنولوجيا، بما في ذلك **عمر البطارية** المحدود والمسافة بين الطائرات بدون طيار. ولمواجهة هذه التحديات، يقترح الباحثون الاستفادة من **خوارزمية هاريس هوكس للتحسين (HHO).** مستوحاة من استراتيجية الطيران التعاونية لصقور هاريس، فإن الهدف من هذا البحث هو تحسين طوبولوجيا **FANET** لتحقيق أكبر عدد ممكن من **المهام** التي تؤديها هذه الطائرات بدون طيار مع دمج الخصائص السلوكية لصقور هاريس من خلال عملية التحسين هذه.

الكلمات المفتاحية: الطائرات بدون طيار، الشبكات اللاسلكية، الشبكات الطائرة المخصصة (FANETs)، عمر البطارية، خوارزمية هاريس هوكس للتحسين (HHO), المهام.

Résume

De nos jours, l'utilisation des **drones** est puissante dans différents **domaines** tels que le transport, l'armée, la photographie et bien d'autres. L'utilisation de ces drones incite les humains à réfléchir à la possibilité de les faire échanger des données et à communiquer entre eux pour accomplir des tâches plus efficacement. C'est ainsi que naissent **les réseaux ad hoc aériens (FANETs)**, qui consistent à créer des **réseaux sans fil** entre des nœuds volants tels que les drones.

Cependant, cette technologie présente des défis, notamment en termes **d'autonomie de la batterie** et de distance entre les drones. Pour relever ces défis, les chercheurs proposent d'utiliser **l'optimisation de Harris Hawks (HHO)**. Inspiré de la stratégie de vol collaborative des faucons de Harris, l'objectif de cette recherche est d'optimiser la topologie des **FANETs** afin d'accomplir le maximum de **tâches** avec ces drones, tout en intégrant les caractéristiques comportementales des faucons de Harris grâce à cette méthode d'optimisation.

Mots clé : drones, Réseaux sans fil, les réseaux ad hoc aériens (FANETs), autonomie de la batterie, l'optimisation de Harris Hawks (HHO), tâches.

Table of Contents

Table of Figures

List of Tables

Abbreviation Table

Abbreviation	Meaning
AODV	Ad-hoc On-Demand Distance Vector
APs	Access Points
DSR	Dynamic Source Routing
DSRC	Dedicated Short-Range Communication
FANETs	Flying Ad-hoc Networks
GCS	Ground Control Station
GPSR	Greedy Perimeter Stateless Routing
HHO	Harris Hawks Optimization
LAN	Local Aera Network
LAR	Location-Aided Routing
MAC	Media Access Control
MANETs	Mobile Ad Hoc Network
mmWave	Millimeter Wave
NED	Network Description file
NICs	Network Interface Cards
OLSR	Optimized Link State Routing
PHY	Physical Layer
QoS	Quality of Service
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
UAVs	Unmanned Aerial Vehicles
UDP	User Datagram Protocol
VANETs	Vehicular Ad-hoc Network

GENERAL INTRODUCTION

I General Introduction

1 General context:

Flying Ad Hoc Networks (FANETs), a cutting-edge network type, have emerged with the advancement of Unmanned Aerial Vehicles (UAVs), commonly known as drones. FANETs facilitate communication and collaboration among drones, forming a network in the sky. These networks are gaining popularity across various fields, including precision agriculture, delivery services, construction, environmental and climate monitoring, and military surveillance.

Drones in FANETs offer significant mobility and flexibility, enabling these networks to cover extensive geographical areas and adapt to changing collaborative tasks. The coordination and collaboration of drones enhance the range and efficiency of a FANET, allowing the network to cover more ground by dispersing during flight.

The development and enhancement in this field, particularly concerning the optimization of communication and task allocation, are of utmost importance. Efficient task allocation and decision-making are critical aspects of networks involving heterogeneous nodes operating in ad hoc mode, such as FANETs. Task allocation can be either mission-focused or based on the simple utilization of available resources. In networks involving mission-critical resources, cooperative task allocation and rendezvous are key factors that drive the mission and optimize performance.

A variety of optimization algorithms have been developed, focusing on the cooperative behavior of nodes and efficient resource management. One such algorithm is the Harris Hawks Optimization (HHO) algorithm, a novel and efficient optimization algorithm. The HHO algorithm is designed based on the cooperative behavior and chasing styles of Harris' hawks in nature. It has several effective features such as escaping energy parameter with a dynamic randomized time-varying nature, different exploration mechanisms, diverse patterns with short-length jumps, and a progressive selection scheme. These features make it an excellent choice for cooperative rendezvous and effective task allocation in FANETs.

Moreover, the significance of developing and enhancing FANETs is further highlighted by the integration of FANETs with various technologies, including Virtual Reality (VR), Internet of Things (IoT), flying edge computing, flying fog computing, cellular networks, and flying cloud computing.

In conclusion, the development and enhancement of FANETs, particularly in terms of communication optimization and task allocation, are crucial to ensure the best collaborative operation with drones. This will not only enhance the performance and range of drones but also open up new avenues for their use in various application domains. The use of the Harris Hawks Optimization algorithm plays a significant role in this process, ensuring optimal task allocation and efficient management of available resources in drones, such as battery and speed.

2 Problematique and Motivation for Applying Harris Hawks Optimization in FANET Task Allocation: A Simulation Approach

2.1 Background

Flying Ad Hoc Networks (FANETs) consist of a group of unmanned aerial vehicles (UAVs) or drones that communicate wirelessly to perform collaborative tasks. These networks are essential in scenarios where traditional infrastructure-based communication is impractical, such as disaster response, environmental monitoring, and surveillance.

2.2 Motivation

The motivation behind applying Harris Hawks Optimization (HHO) to FANET task allocation lies in its unique features:

- **Hawk-Inspired Behavior**: HHO mimics the collaborative hunting behavior of hawks. Similarly, drones in FANETs can collaborate to achieve common goals.
- **Exploration and Exploitation**: HHO balances exploration (searching for better solutions) and exploitation (utilizing known solutions) during optimization.
- **Robustness**: HHO has demonstrated robustness in solving various optimization problems.

2.3 Research Goals

Our research aims to:

1. **Adapt HHO for FANETs**: Investigate how HHO can be adapted to address task allocation challenges in FANETs.

2. **OMNeT++ Simulation**: Create an OMNeT++ simulation to model the FANET topology using INET, including a base control station and a group of drones.
3. **Packet-Based Communication**: Implement packet-based communication between the base control station and drones to allocate tasks effectively.

3 Thesis Structure and Contribution:

This Thesis is structured into three main chapters, each of which includes an introduction to the chapter's subject matter, a detailed exploration of the content, and a conclusion summarizing the chapter's main idea.

The first chapter, delves into wireless networks, discussing both wireless network infrastructure and autonomous Ad Hoc networks. It provides specific details about elements such as Access Points, Wireless Routers, and Wireless LAN Controllers for the wireless network infrastructure, and explores MANETs, VANETs, and FANETs for the autonomous Ad Hoc networks.

The second chapter, titled 'Harris Hawks Optimization and Tasks Allocation', explains the concept of Harris Hawks Optimization and its functioning. It also offers a preview of FANETs and discusses how task allocation should be handled. Furthermore, it explores the connection between Harris Hawks Optimization and FANETs task allocation, providing a research example for better understanding.

The final chapter, titled 'A Harris Hawks Optimization Approach for Fanet Task Scheduling', presents our study approach and its application in the OMNet++ simulator. It discusses the results of the applied approach and analyzes these results to determine their effectiveness.

In conclusion, the thesis provides a comprehensive summary of all the topics covered and suggests potential future approaches that could be applied within the context of Harris Hawks Optimization.

Chapter 01

Wireless Networks

II Wireless Networks

1 Introduction:

Wireless networks have revolutionized modern communication by enabling convenient and flexible connectivity without the constraints of physical cables[1]. They have become pervasive in our daily lives, supporting various applications ranging from internet access and mobile communications to IoT (Internet of Things) devices and smart home automation[1]. This is an overview of wireless networks, including their definition, types, key components, and applications[1].

2 Infrastructure:

Wireless network infrastructure refers to the physical and logical components that enable wireless communication between devices[2]. These networks provide connectivity without the need for physical cables, making them versatile and suitable for various applications[2]. Here's an overview of wireless network infrastructure components:

2.1 Access Points (APs):

- APs are devices that serve as central hubs for wireless communication within a network.
- They transmit and receive wireless signals to and from client devices such as smartphones, laptops, and IoT devices.
- APs may be standalone devices or integrated into other network equipment like routers or switches.

2.2 Wireless Routers:

- Wireless routers combine the functionality of a traditional wired router with wireless access points.
- They connect to a modem (or directly to an ISP) to provide internet access to wired and wireless devices in a network.
- Routers manage traffic between devices within the local network and between the local network and the internet.

2.3 Wireless Network Interface Cards (NICs):

- NICs are hardware components or integrated circuits that enable devices to connect to wireless networks.
- They facilitate the transmission and reception of wireless signals, allowing devices to communicate wirelessly.

2.4 Wireless LAN Controllers:

- WLAN controllers are centralized devices that manage multiple APs within a wireless network.
- They provide features such as centralized configuration, monitoring, and security management for APs and client devices.

2.5 Antennas:

- Antennas are essential components of wireless communication systems.
- They transmit and receive radio frequency signals, allowing devices to communicate over wireless networks.
- Antennas come in various types, including omni-directional antennas that transmit signals in all directions and directional antennas that focus signals in specific directions.

2.6 Wireless Protocols and Standards:

- Wireless networks operate based on various protocols and standards, such as Wi-Fi (IEEE 802.11), Bluetooth, Zigbee, and cellular standards like LTE and 5G.
- These standards define communication methods, data rates, security mechanisms, and compatibility requirements for wireless devices and networks.

2.7 Network Infrastructure Components:

- In addition to wireless-specific components, wireless networks often include traditional network infrastructure such as switches, routers, firewalls, and gateways.
- These components help manage traffic, enforce security policies, and provide connectivity between wireless and wired networks.

2.8 Wireless Security Mechanisms:

- Wireless networks implement security mechanisms to protect data and prevent unauthorized access.
- Encryption protocols like WPA2/WPA3, authentication methods such as WPA-Enterprise, and network segmentation techniques help secure wireless communications.

wireless network infrastructure comprises a range of components and technologies that enable wireless communication, connectivity, and management for various applications and environments.

3 Autonome Ad Hoc:

In the context of networks, "ad hoc" refers to a temporary, decentralized network formed on the spot without relying on pre-existing infrastructure[3]. These networks are often established by devices communicating directly with each other, without the need for a central access point or router[3].

Here are some key characteristics of ad hoc networks:

- **Decentralized:** There is no single point of control or central device managing the network.
- **Temporary:** Ad hoc networks are typically formed for a short period, such as for sharing files between two laptops or enabling temporary internet access during an event.
- **Self-configuring:** Devices automatically discover each other and establish communication channels without human intervention.
- **Dynamic:** The network topology can change as devices join or leave the network.

Here are some common applications of ad hoc networks:

- **File sharing:** Two or more devices can share files directly, bypassing the need for a central server.
- **Gaming:** A group of players can create a temporary network for multiplayer gaming.
- Emergency communication: Ad hoc networks can be established in disaster situations to facilitate communication when other infrastructure is damaged.
- **Sensor networks:** In wireless sensor networks, nodes can establish ad hoc connections to share data and collaborate.

While ad hoc networks offer flexibility and convenience, they also come with some limitations:

- **Limited range:** They typically have a shorter range compared to traditional networks with access points.

- **Security concerns:** Ad hoc networks are generally less secure than wired or managed wireless networks, as they lack centralized security controls.
- **Scalability concerns:** As the number of devices in the network increases, performance can degrade due to limited bandwidth and increased complexity in managing connections.

ad hoc networks offer a valuable tool for situations where temporary connectivity is needed. However, it's important to be aware of their limitations and consider alternatives like Wi-Fi hotspots or mobile data when security, performance, or larger-scale connectivity are crucial.

3.1 Mobile Ad Hoc Network (MANET):

A network formed by mobile devices such as laptops, smartphones, and tablets. These devices can communicate directly with each other without needing a central access point[4].

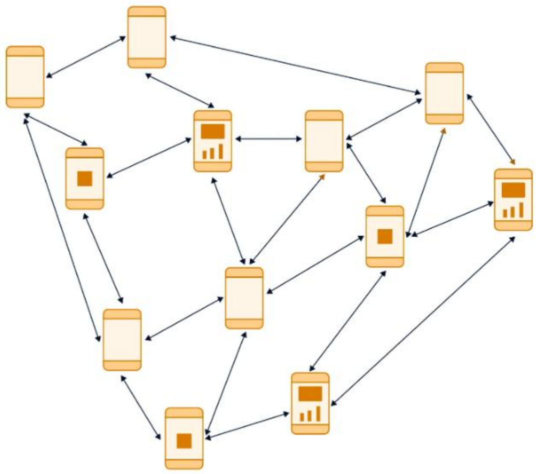

Figure 1 Mobile Ad-Hoc Network example

3.2 Vehicular Ad-hoc Network (VANET):

A network formed by vehicles equipped with wireless communication devices[5]. These devices can exchange information about traffic conditions, road hazards, and other relevant data to improve safety and efficiency[5].

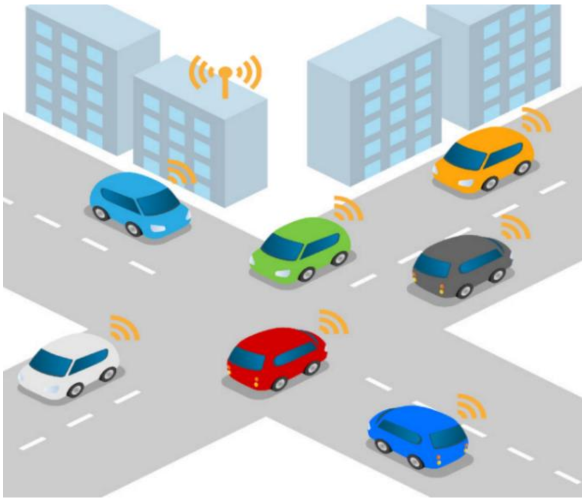

Figure 2 Vehicular Ad-Hoc Network example

3.3 Flying Ad-hoc Networks (FANETs):

FANETs, or Flying Ad-hoc Networks, are a specific type of mobile ad-hoc network (MANET) composed of Unmanned Aerial Vehicles (UAVs) that communicate and collaborate without relying on a pre-existing infrastructure[6]. They are becoming increasingly popular due to their unique capabilities and diverse potential applications[6].

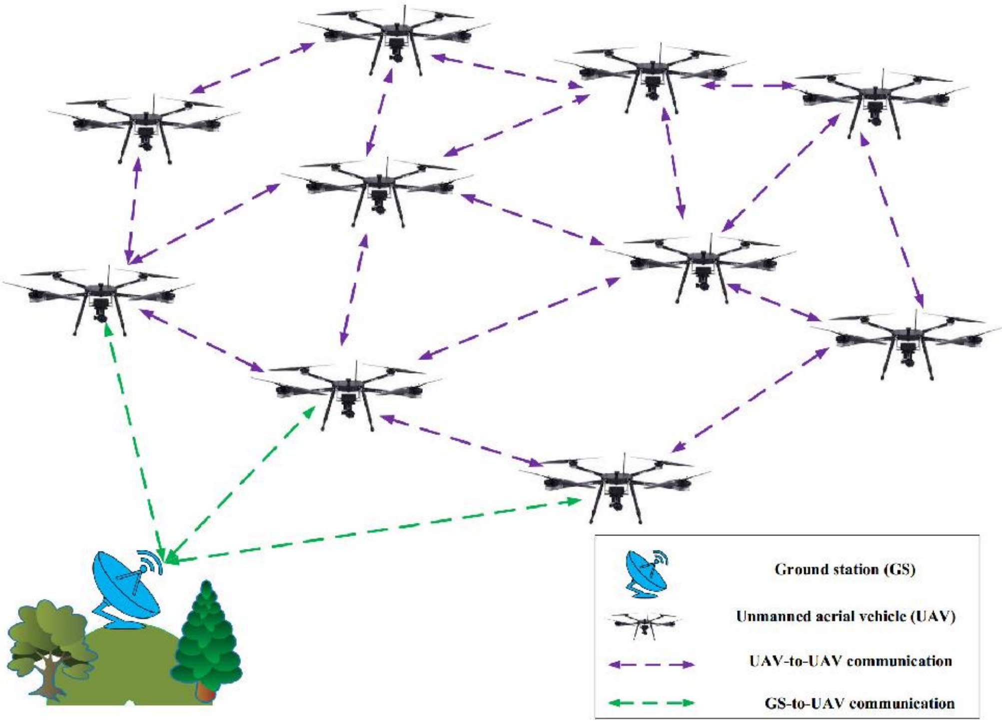

Figure 3 Flying Ad-hoc Network example

3.3.1 Characteristics:

- **High Mobility:** Unlike static nodes in traditional MANETs, UAVs can move freely in three dimensions, leading to frequent changes in network topology and challenging connectivity management.
- **Dynamic Network Formation:** UAVs can join or leave the network dynamically, requiring efficient mechanisms for handling network membership changes and maintaining connectivity.
- **Limited Resources:** UAVs are constrained by battery power, processing capacity, and communication bandwidth, necessitating resource-efficient network protocols and algorithms.
- **Varying Communication Range:** The communication range between UAVs depends on factors like altitude, terrain, and antenna characteristics, requiring adaptable protocols to cope with changing link quality.
- **Security Concerns:** Secure communication in FANETs is crucial to prevent unauthorized access or data interception, posing unique challenges due to the dynamic nature of the network and potential vulnerabilities of wireless communication.

3.3.2 Applications:

- **Search and Rescue:** FANETs can be deployed to search for missing persons or assess disaster zones, offering aerial coverage and rapid response capabilities.
- **Environmental Monitoring:** UAVs equipped with sensors can collect data on air quality, land use, wildlife populations, or environmental hazards, providing valuable insights for various purposes.
- **Precision Agriculture:** FANETs are used for crop monitoring, soil analysis, and targeted pesticide application, improving agricultural efficiency and resource management.
- **Delivery Services:** UAVs can deliver goods in remote areas, congested urban environments, or for last-mile delivery situations, offering faster and more flexible logistics solutions.
- **Traffic Management:** Real-time traffic monitoring and incident detection through aerial surveillance can improve traffic flow and management in cities or on highways.

3.3.3 Routing Protocols:

Traditional routing protocols designed for static networks require adaptations to cope with the dynamic nature of FANETs. Some common routing protocols in FANETs include:

- **Proactive Routing Protocols:** Proactively maintain routing information throughout the network, such as Optimized Link State Routing (OLSR) and AODV (Ad-hoc On-Demand Distance Vector).
- **Reactive Routing Protocols:** Establish routes on demand when needed, such as DSR (Dynamic Source Routing) and TORA (Temporally Ordered Routing Algorithm).

- **Position-based Routing Protocols:** Utilize location information of nodes for efficient routing, such as GPSR (Greedy Perimeter Stateless Routing) and LAR (Location-Aided Routing).

3.3.4 Mobility Models:

Mobility models are crucial for simulating and evaluating FANETs in the absence of real-world deployment. These models represent the movement patterns of UAVs within the network. Some common mobility models include:

- **Random Waypoint Model:** UAVs move randomly between predefined waypoints.
- **Random Walk Model:** UAVs take random steps in a specific direction.
- **Group Mobility Model:** UAVs move in groups with similar patterns.
- **Mission-Based Mobility Model:** UAVs follow pre-defined flight paths based on their specific mission objectives.

3.3.5 Communications Technology:

FANETs typically utilize wireless communication technologies to enable communication between UAVs and potentially with a Ground Control Station (GCS). Common communication technologies include:

- **Wi-Fi:** Widely used for short-range communication within limited areas.
- **Cellular Networks:** Provides wider coverage but may incur higher costs and potential latency issues.
- **Dedicated Short-Range Communication (DSRC):** Designed for vehicle-to-vehicle communication, but can be adapted for FANETs in specific scenarios.
- **Millimeter Wave (mmWave):** Offers high data rates but has limited range and can be affected by weather conditions.

3.3.6 Challenges and Future Directions:

Despite their potential, FANETs face several challenges:

- **Limited Battery Life:** Efficient energy management and resource allocation are crucial for extending the operational time of UAVs.
- **Security and Privacy:** Ensuring secure communication and protecting sensitive data in a dynamic and potentially vulnerable network environment requires robust security mechanisms.
- **Scalability and Interoperability:** Efficiently managing large-scale FANETs and ensuring interoperability between different FANET systems are ongoing areas of research and development.

As research and development in FANETs continues, advancements in areas like AI-powered network management, secure communication protocols, and energy-efficient technologies are expected to address existing challenges and pave the way for the wider adoption of FANETs in various applications.

3.3.7 Architecture:

The architecture of Fanet is as shown in the next Figure. It is divided into three main parts they are :

1. Enabling Technologies
2. Networking
3. Middleware and Application

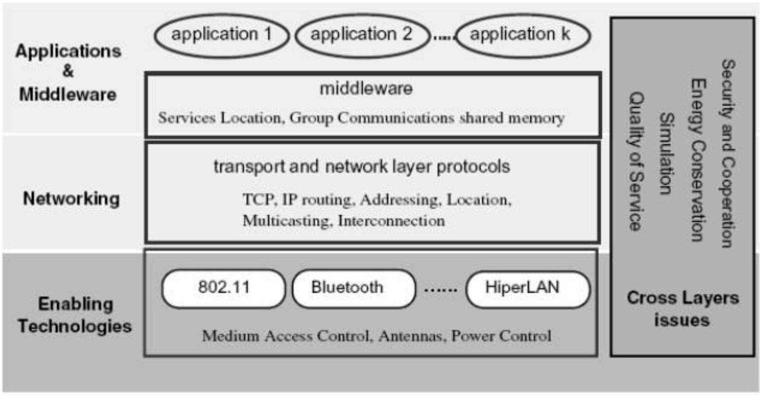

Figure 4 FANETs Architecture

❖ **Enabling Technologies**

Enabling technologies are further divided depending on their area of coverage.

1. BAN (Body Area Network): The communication range of BAN is 1 to 2 meters. It provides connectivity to the wearable computing devices.

2. PAN (Personal Area Network): The communication range of PAN is up to 10 meters. It provides
connectivity between mobile devices as well as stationary devices.

3. WLAN (Wireless Local Area Network): the communication range of WLANs is 100 to 500 meters.
It can connect a building or a group of buildings.

❖ **Networking**

In FANET architecture, most of the principal functionalities of the networking protocols need to be redesigned for self-configuring, dynamic, unstable, peer-to-peer communication environments. The initial target of networking protocols is to use the one-hop transmission services, given by the enabling technologies to evolve end-to-end reliable services, from a sender to the receiver. To inculcate an end-to-end communication the sender needs to find the receiver inside the network. The major task of a location service is to dynamically map the address of the receiver device to its present location in the network.

- **Middleware and Application**

The wireless technologies like WiFi, Bluetooth, IEEE 802.11, WiMAX and Hyper LAN remarkably encourages the deployment of ad hoc technology and new ad hoc networking applications mainly in specific fields like emergency services, disaster recovery and environment monitoring. The versatility of MANET makes this innovation adaptive for some practical situations like, in PAN, home networking, law enforcement operation, commercial and educational applications, and sensor network. Mobile ad hoc frameworks recently created adopt the methodology of not having a middleware, yet rather depend on every application to handle every one of the services it needs.

4 Conclusion

Wireless networks provide convenience and flexibility without requiring physical connections, they have completely changed the way we interact. Access Points (APs), among other infrastructure elements, are essential for maintaining constant connectivity for a variety of applications. For wireless networks to operate effectively and securely, a deep understanding of wireless protocols, standards, and security measures is required. On the other hand, ad hoc networks offer transient, decentralized connectivity, which makes them appropriate for exchanging sensor data or disaster response scenarios. Ad hoc networks are flexible, but they also have drawbacks. These include scalability problems and security concerns, which should be carefully taken into account before using them for a specific purpose.

Chapter 02

Harris Hawks Optimization and Tasks Allocation

III Harris Hawks Optimization and Tasks Allocation

1 Introduction:

Task allocation for drones is a pivotal aspect in optimizing unmanned aerial vehicle (UAV) operations across numerous applications. Drones, increasingly utilized for tasks like surveillance, package delivery, and agriculture, rely on efficient task allocation to maximize mission success and resource utilization. The strategic assignment of tasks among drones, considering mission requirements, capabilities, and environmental conditions, enhances efficiency and reduces operational costs. However, challenges such as task prioritization, coordination among multiple drones, and dynamic reassignment complicate this process. Robust algorithms, like Harris Hawks Optimization (HHO) Which depends on the cooperative behavior of falcons, which facilitates the process of distributing tasks at the expense of the characteristics of drones.

2 Harris Hawks Optimization:

2.1 Definition:

Harris Hawks Optimization (HHO) is a nature-inspired metaheuristic optimization algorithm that was proposed based on the social behavior of Harris's Hawks[7], [8], a species of bird of prey. The algorithm is designed to solve optimization problems by mimicking the hunting behavior of Harris's Hawks in nature[7], [8].

The basic principles of HHO are derived from the cooperative hunting behavior observed in these birds. In their natural habitat, Harris's Hawks exhibit a strategy where they collaborate in hunting prey, with some individuals taking on different roles to maximize hunting efficiency. This behavior involves cooperation, communication, and a division of tasks among group members[7], [8], [9].

In the context of optimization algorithms, HHO simulates this cooperative behavior to find optimal solutions to various types of optimization problems. The algorithm maintains a population of solutions, with each solution representing a potential solution to the optimization problem. Through a process of exploration and exploitation, solutions are iteratively improved over successive generations[7].

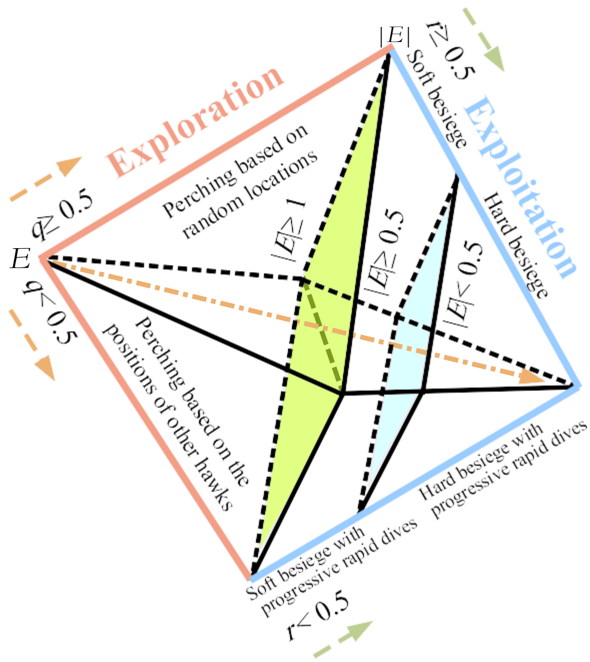

Figure 5 Harris Hawks Optimization Different Phases

The preceding figure provides a visual representation of the Harris Hawks Optimization (HHO) algorithm. This algorithm draws inspiration from the natural cooperative behavior and hunting tactics of Harris' hawks. The diagram showcases the equilibrium between the exploration and exploitation stages, which are fundamental elements of the HHO algorithm[7]. The following is an interpretation based on the figure:

- **Exploration Phase (α ≥ 0.5):** During this phase, hawks perch based on random locations or position themselves based on the position of other hawks. This represents the hawks scanning the environment and searching for prey[7].
- **Exploitation Phase (α < 0.5):** In this phase, hawks engage in "soft besiege" or "hard besiege" behaviors, with "soft/hard besiege with progressive rapid dives" being another tactic. This phase reflects the hawks' strategies for approaching and capturing prey through coordinated attacks and rapid dives[7].

The randomness parameter **(α)** plays a crucial role in determining the hawks' behavior, with different actions associated with values **less than 0.5** and **greater than or equal to 0.5**[7]. The algorithm's design allows for an adaptive balance between exploring the search space and exploiting the known good solutions, aiming for efficient task allocation and resource management in applications like FANETs. The HHO algorithm's ability to manage available resources, such as battery and speed in drones, makes it suitable for optimizing task allocation in FANET network topologies[7].

2.2 **Phases:**

2.2.1 Exploration:

This involves the exploration of the search space to discover potential solutions. In HHO, this is akin to the behavior of Harris's Hawks scouting for potential prey[10].

In HHO, the Harris' hawks perch randomly on some locations and wait to detect a prey based on two strategies[7].

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where $X(t+1)$ is the position vector of hawks in the next iteration (t), $X_{rabbit}(t)$ is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside $(0, 1)$, which are updated in each iteration, (UB, LB) show the upper and lower bounds of variables, $X_{rand}(t)$ is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks[7]. The average position of hawks is attained using the next equation:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \qquad (2)$$

where $X_i(t)$ indicates the location of each hawk in iteration t and N denotes the total number of hawks.

2.2.2 Transition from exploration to exploitation:

To model this step, the energy of a rabbit is modeled as:

$$E = 2E_0(1 - \frac{t}{T}) \qquad (3)$$

where E indicates the escaping energy of the prey, T is the maximum number of iterations, and E_0 is the initial state of its energy. The time-dependent behavior of E is also demonstrated in The Figure below[7].

$$E=2E_0(1-t/T)$$

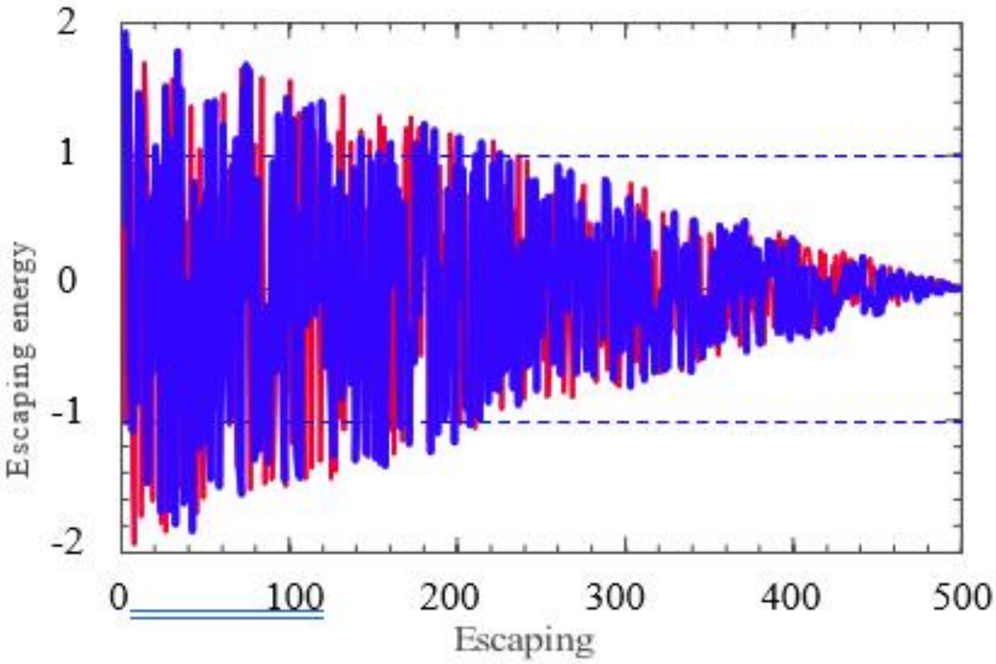

Figure 6 Behavior of E during two runs and 500 iterations

2.2.3 Exploitation:

Once promising solutions are identified, the algorithm focuses on exploiting these solutions to refine them further. This mirrors the cooperative hunting behavior of Harris's Hawks as they coordinate their efforts to capture prey[7], [10].

- **Soft besiege:** This behavior is modeled by the following rules:

$$X(t+1) = \Delta X(t) - E\ |JX_{rabbit}(t) - X(t)| \qquad (4)$$

And This one also :

$$\Delta X(t) = X_{rabbit}(t) - X(t) \qquad (5)$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in the iteration t, r_5 is a random number inside (0,1), and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure[7]. The J value changes randomly in each iteration to simulate the nature of rabbit motions[7].

- **Hard besiege:** In this situation, the current positions are updated using the next equation :

$$X(t + 1) = X_{rabbit}(t) - E\,|\Delta X(t)| \qquad (6)$$

A simple example of this step with one hawk is depicted in the next Figure:

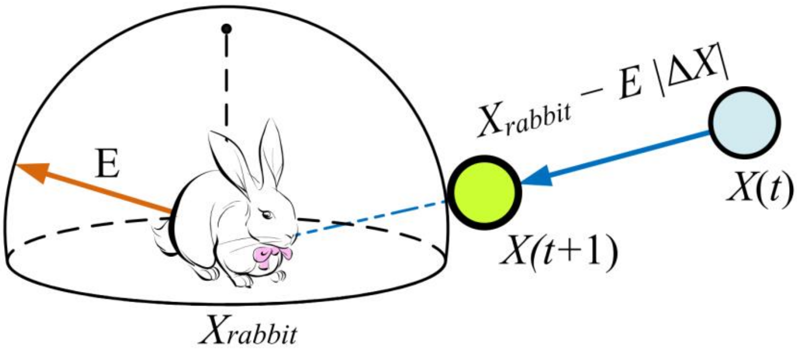

Figure 7 Example of overall vectors in the case of hard besiege

2.2.4 Communication:

HHO incorporates communication mechanisms among the individuals (solutions) in the population[7], [10]. This allows sharing of information and experiences, facilitating better cooperation and coordination in the search for optimal solutions.

2.2.5 Division of tasks:

Different individuals in the population may play different roles, akin to the division of tasks observed in Harris's Hawks during hunting. Some individuals may explore new areas of the search space, while others exploit promising solutions or share information with the group[7], [10].

2.3 Pseudo Code (HHO):

Here is a pseudo code that is represent the Harris Hawks Optimization

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population Xi(i = 1, 2, . . . , N)
while (stopping condition is not met) do
Calculate the fitness values of hawks
Set Xrabbit as the location of rabbit (best location)
for (each hawk (Xi)) **do**
    Update the initial energy E0 and jump strength J ▷ E0=2rand()-1, J=2(1-rand())
    Update the E using Eq. (3)
    if (|E| ≥ 1) **then** ▷ Exploration phase
        Update the location vector using Eq. (1)
    if (|E| < 1) **then** ▷ Exploitation phase
        if (r ≥0.5 and |E| ≥ 0.5) **then** ▷ Soft besiege
            Update the location vector using Eq. (4)
        else if (r ≥0.5 and |E| < 0.5) **then** ▷ Hard besiege
            Update the location vector using Eq. (6)
        else if (r <0.5 and |E| ≥ 0.5) **then** ▷ Soft besiege with progressive rapid dives
            Update the location vector using Eq. (10)
        else if (r <0.5 and |E| < 0.5) **then** ▷ Hard besiege with progressive rapid dives
            Update the location vector using Eq. (11)
Return Xrabbit

Figure 8 Harris Hawks Source Code

3 FANETs and Task Allocation:

FANETs (Flying Ad-hoc Networks) are a type of ad-hoc network that involves unmanned aerial vehicles (UAVs). These networks are particularly useful in situations where human intervention is dangerous or impractical, such as disaster management, military expeditions, and civilian applications[11]. One of the key challenges in FANETs is task allocation and cooperative rendezvous[12]. This involves assigning various tasks to different nodes in the network in a way that optimizes the overall performance of the network[12]. The task allocation can be either mission-based or based on the simple utilization of available resources[12].

Harris Hawks Optimization (HHO) is a swarm-based optimization method that mimics the action and reaction of Harris Hawks when they go hunting[8]. The HHO algorithm unfolds in two distinct phases: exploration and exploitation[13]. This intricate algorithm offers robust strategies to effectively navigate away from local optima, rendering it proficient at approximating and even converging upon global optima[13].

The relation between HHO and FANETs Tasks allocation is that the HHO algorithm can be used as a method to solve the task allocation problem in FANETs. By representing the task allocation problem as an optimization problem, the HHO algorithm can be used to find the optimal solution[13]. That is, it can find the best way to assign tasks to nodes in the network such that the overall performance of the network is maximized[13].

For example, in a FANET, We might have a set of tasks that need to be performed by the UAVs, such as monitoring a specific area, delivering a package, or collecting data. Each of these tasks could be assigned to a specific UAV. The HHO algorithm could be used to determine the optimal assignment of tasks to UAVs, taking into account factors such as the capabilities of each UAV, the location of the tasks, and the energy consumption of performing each task.

4 Applications of Harris Hawks optimization in mobile ad hoc networks

In [14] R.Sharma and S.Prakash propose a new model for positioning sensor nodes in a network. The model uses three fixed anchor nodes (represented by their coordinates) and a sensor node with an unknown location. A special algorithm called Harris Hawks Optimization (HHO) is used to analyze data from the network, along with relative signal strength in a wireless environment (RSSI), to estimate the position of the unknown sensor node.

For comparison, they use a traditional localization model that uses different optimization algorithms (Salp Swarm Algorithm (SSA), Grey Wolf Optimizer (GWO) and Equilibrium Optimizer (EO)) for positioning. Figure shown below illustrates the new and improved model that leverages the HHO algorithm for anchor-based localization. They show that their approach achieves better results in terms of accuracy, number of located nodes, and efficiency.

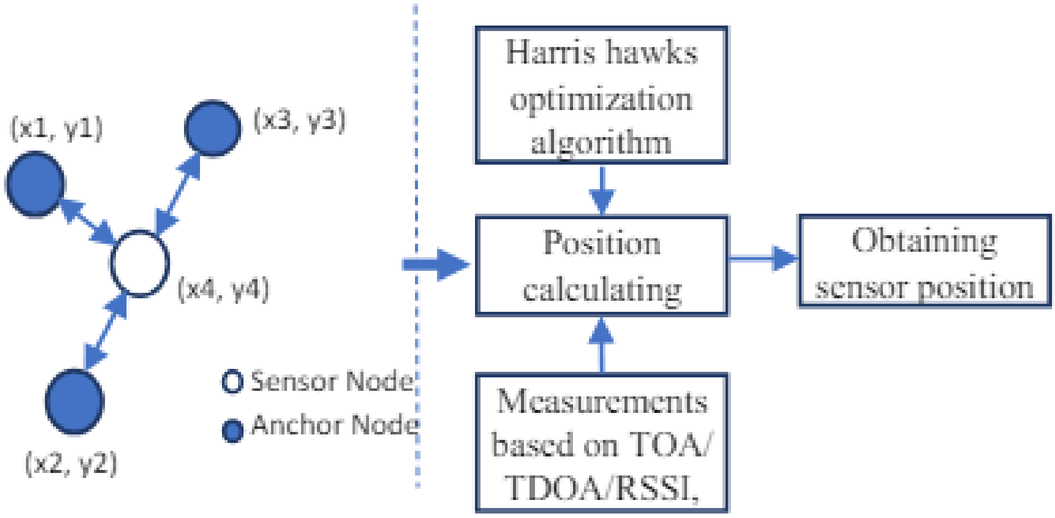

Figure 9 Anchor-based sensor nodes localization approach using harris hawks optimization algorithm WSNs

Wireless sensor networks (WSNs) struggle with creating efficient clusters and routing data. X.Xue and al [15] tackles this challenge by proposing a two-pronged approach:

Energy-efficient clustering: This is achieved using a novel method called k-medoids with improved artificial-bee-colony (K-IABC). This method creates clusters in a way that saves energy.

Cross-layer routing: To overcome uneven power distribution across sensor nodes (power asymmetry), an optimal routing solution is implemented. This routing protocol, called Cross-layer-based Harris-hawks-optimization-algorithm (CL-HHO), considers factors from various network layers to minimize delays and energy use.

The researchers built and tested CL-HHO in MATLAB. The results show significant improvement in network performance compared to existing techniques. Specifically, CL-HHO performs better in terms of:

- Packet loss ratio (PLR) - fewer lost data packets.
- Throughput - faster data transmission.
- End-to-end delay (E2E) - less time for data to travel from source to destination.
- Jitter - reduced variation in data arrival times.
- Network lifetime (NLT) - sensors last longer before needing battery replacement.
- Buffer occupancy - less congestion in data buffers.

Finally, the study compares CL-HHO to other routing strategies and finds that it outperforms them all, including HEED, EECRP, GWO, and CL-ALO.

A.Ali, FAadil, M.Fahed Khan and M.Maqsood [16] propose a Clustering Algorithm for Vanet based on Harris Hawks Optimization. The HHOCNET is inspired by how hawks work together to hunt (surprise attack) to create the best clusters of vehicles. The way the HHO algorithm works with randomness (stochastic operators) and balances searching everywhere (exploration) with focusing on promising areas (exploitation) helps it avoid getting stuck on solutions that aren't quite right (local optima) and find the absolute best solution (optimal number of vehicle clusters). They tested this system using computer simulations (MATLAB) and compared it to other advanced methods (like clustering algorithms based on Gray Wolf optimization, Multi-objective Particle Swarm Optimization, and Comprehensive Learning Particle Swarm Optimization). They looked at how well these methods performed in various ways. Their findings show that this new approach (HHOCNET) is a great way to cluster vehicles in a network (VANET) and outperforms the others in optimizing this clustering problem with multiple goals.

HHOCNET chooses only 36.04% of vehicles to be cluster heads, whereas existing methods pick much higher percentages (50.42%, 56.7%, and 60.89% for GWOCNET, CLPSO, and MOPSO respectively). This proposed method improves how the vehicle network works by up to 15%. As a result, it makes the network run more efficiently by using less wireless resources. It also reduces the number of times information needs to be passed between vehicles (hops) to reach its destination. This leads to shorter delays in communication (end-to-end communication latency).

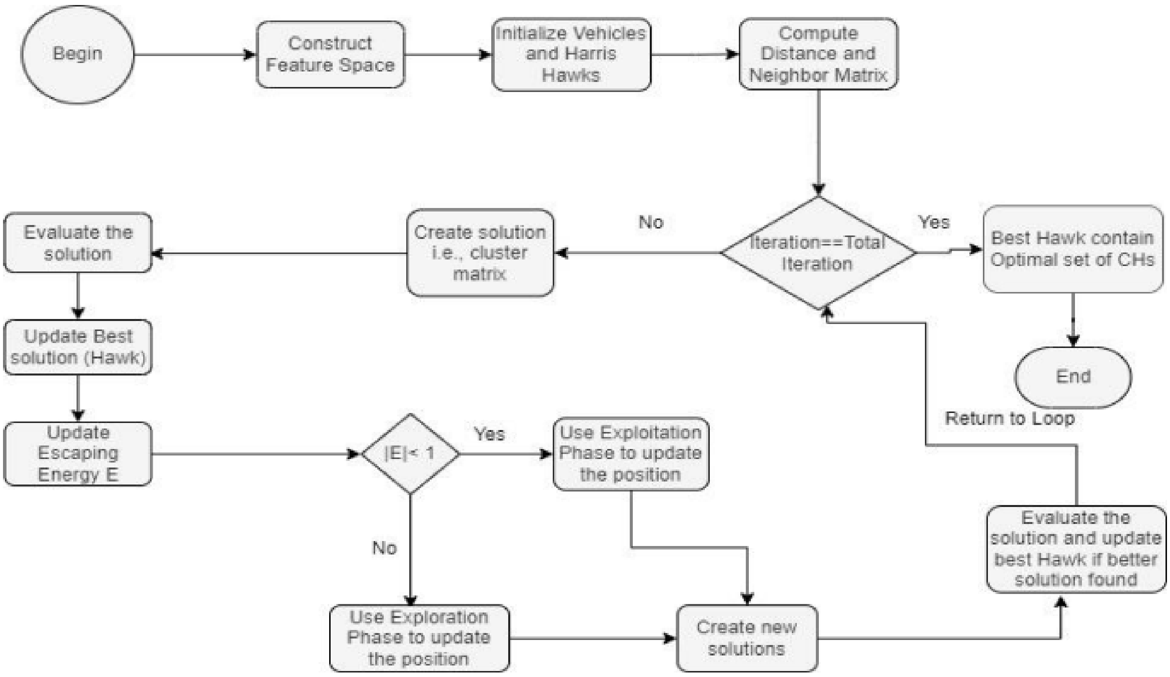

Figure 10 HHOCNET Flowchart

B. Farooq Ahmed and al [17] proposes a new method called Refined Adaptive Harris Hawks Optimization Algorithm (RAHHO) to improve security in mobile ad hoc networks (MANETs) used for Internet of Things (IoT) applications. RAHHO builds upon the strengths of the Harris Hawks Optimization (HHO) algorithm by adding a mechanism that updates dynamically. This improves how fast and accurately the algorithm finds the best settings for security features. RAHHO takes advantage of HHO's ability to adapt and changes where the "hawks" search and how they search. This allows for a more thorough exploration of the different security settings that are important for MANETs.

The study compares RAHHO to other optimization algorithms and finds that RAHHO is significantly less vulnerable to attacks. Compared to the original HHO, RAHHO is over 73% less vulnerable. It's also more secure than other variations like CHHSO and MU-HHO. Simulations show that RAHHO is very effective at improving network security, leading to better overall network performance. In conclusion, this research introduces RAHHO as the most secure and best-performing algorithm for this specific task. The improvements in security from

HHO to RAHHO show a clear link between how well the algorithm searches for optimal settings and the overall security of the network.

M.ASIF.HOSSAIN, R.MD.NOOR and al [18] propose a new routing algorithm for CR-VANETs. This 2-Hop routing algorithm leverages a technique called Multi-Objective Harris Hawks Optimization (2HMO-HHO). It focuses on finding the best intermediate vehicles (forwarders) to relay data between the source and destination vehicles. By choosing only two hops instead of the entire route, the algorithm aims for a more stable connection and ensures successful data transmission.

The study used simulations to evaluate the effectiveness of the proposed algorithm. These simulations, conducted using specialized software (OMNeT++ and SUMO), show promising results. The 2HMO-HHO routing algorithm achieves improvements in several key metrics, including data transfer rate (throughput), delivery time (delay), message success rate (packet delivery ratio), message loss rate (packet loss rate), and communication overhead. Overall, the research suggests that this approach offers a significant advancement in routing data within CR-VANETs.

Figure 11 The 2HMO-HHO routing algorithm

5 Conclusion:

A metaheuristic algorithm inspired by nature, the Harris Hawks Optimization (HHO) model models the cooperative hunting style of Harris's hawks. It is essential for maximizing the distribution of tasks across different domains, particularly in Flying Ad-hoc Networks (FANETs). By making use of HHO's characteristics, tasks can be distributed across drones in an efficient manner, enhancing mission performance and maximizing resource usage. HHO offers a solid framework for effective task distribution in dynamic circumstances, despite difficulties with task priority and coordination.

Chapter 03

A Harris Hawks Optimization Approach for Fanet Task Scheduling

IVA Harris Hawks Optimization Approach for Fanet Task Scheduling

1 Introduction:

Our project revolves around implementing Harris Hawks Optimization in FANETs to foster collaborative work among drones. The process involves the virtual creation of topology to observe the outcomes, which is the primary reason for this simulation. Within this simulation, we will explore the application of Harris Hawks behavior to drones and examine how this technology aids in energy conservation by assigning the appropriate task to the suitable drone.

2 A Harris Hawks Optimization Approach for Fanet Task Scheduling:

The goal of this approach is to Apply the first Step or Phase of Harris Hawks Optimization (Exploration Phase) to our FANET Topology, the Network has a Control Base Station that is responsible for the Selection and Sending of exploring orders to Drones UAVs. We also aim to optimize this step to be more precise in Drones selection to achieve optimal solutions to **reduce energy consumption**, reach the **largest number of explored areas**, and also **extend the life of the exploration** process.

The Exploration phase in this simulation will be divided to three steps:

- **The first step:** The Base Control Station temporarily every period of time sends a random exploration Area coordination to all Drones in the network.
- **The second step:** The Drones receive the coordination and Calculate power Consumption statistics according to the distance between the Actual position of the Drone and the Exploration area, after that the drone resends a Report with these Statistics to the Base Control Station.
- **The third step:** The Base Control Station after receiving all Drones Reports will make the decision about witch of the Drones should go to explore the area.

The next figure shows how the Exploration phase of Harris Hawks Optimization steps are done:

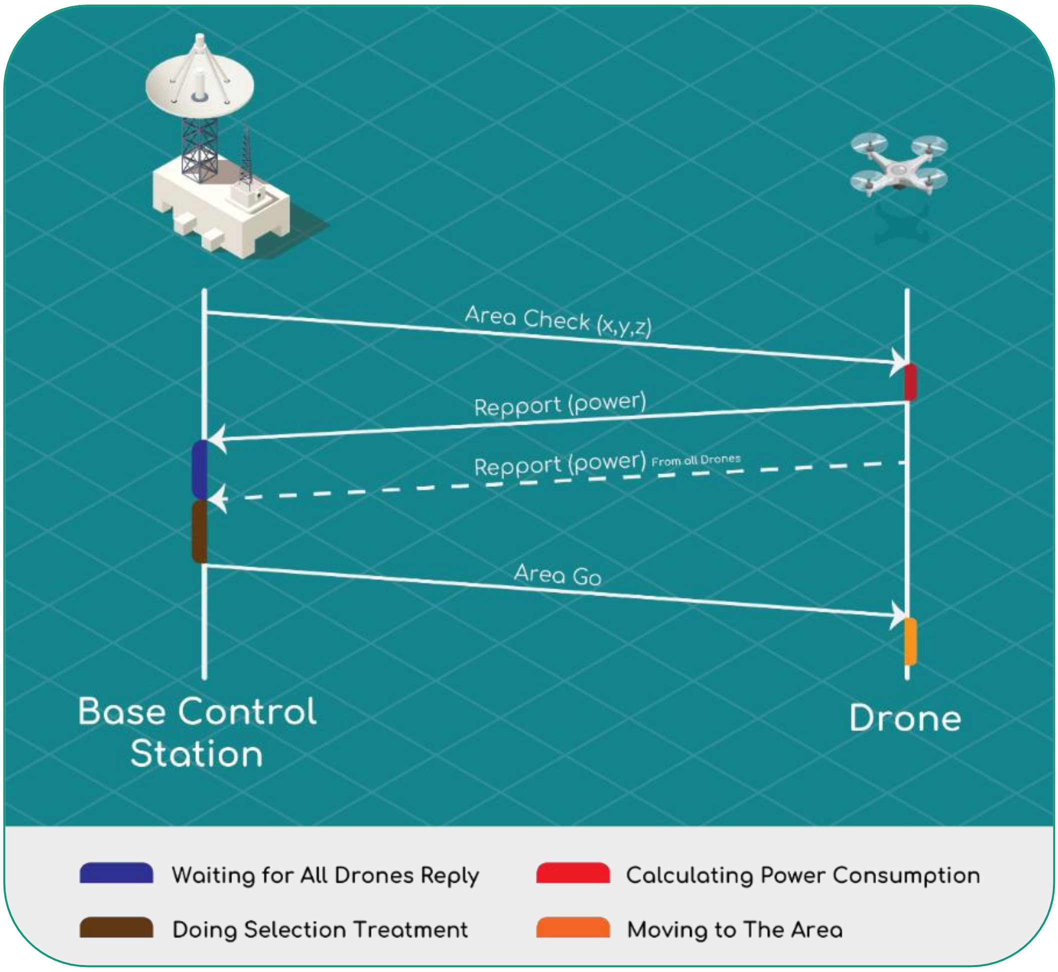

Figure 12 Exploration steps

Note: The base Control Station should wait for all the drones to Reply to decide on selecting which group of UAVs is going to explore the area, if there is a long delay the Base Control Station should make the decision with those who sends replies only.

3 Simulation:

Simulation is a process that consists of designing a model of the (real) system studied, carrying out experiments on this model (and not calculations), interpreting the observations provided by the model's progress, and formulating decisions relating to the system. The goal can be to understand the system's dynamic behavior, compare configurations, evaluate different control strategies, and evaluate and optimize performance.

There are two categories of computer simulation, discrete simulations and continuous simulations. In continuous simulations, quantities are represented by continuous variables, while in discrete simulation systems, the quantities of interest are represented by values of discrete variables [19].

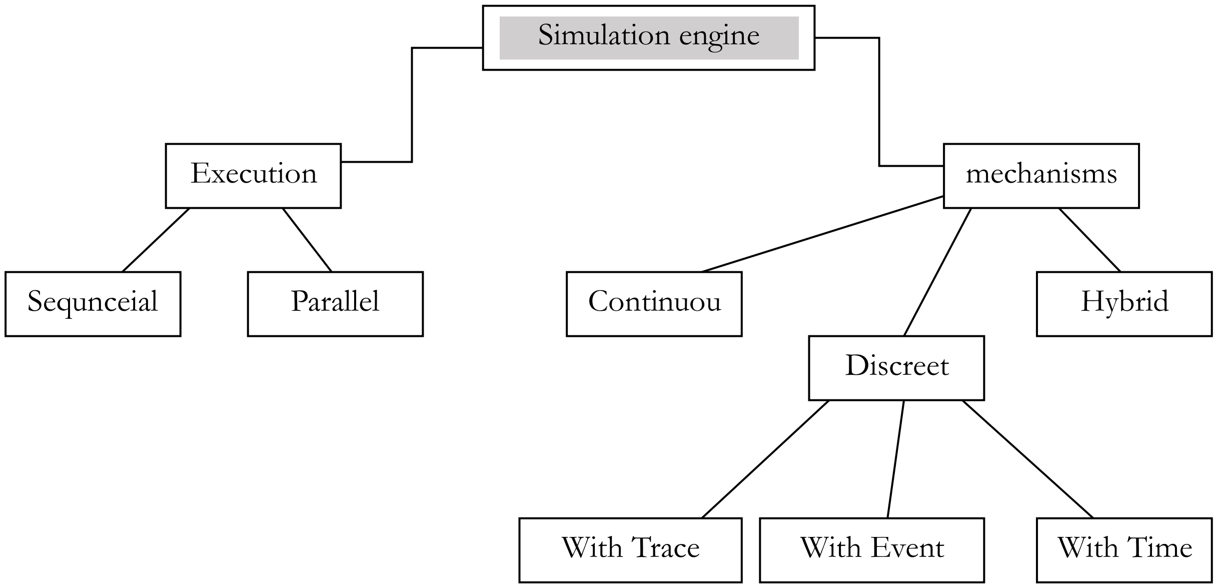

Figure 13 Simulation Types

The dynamics of a discrete simulation can be thought of as a sequence of events at discrete times. As a third type of simulation, Monte Carlo simulation is related to discrete event simulation, which is commonly used to model stochastic systems. "Monte Carlo methods are a widely used class of computational algorithms for simulating the behavior of various physical and mathematical systems, and for other calculations", Monte Carlo simulators usually make use of random numbers to model nondeterministic parts in order to simulate the system. They calculate their results by repeating random sampling a large number of times. The law of large numbers is often used as a justification for the accuracy of results.

Continuous simulations are implemented as a set of equations. The simulation program solves all equations periodically by numerical evaluation.

For large-scale simulations, hybrid approaches have emerged as viable solutions, where large parts are implemented as discrete simulations and other lesser parts as continuous simulations. Hybrid simulation strategies except significant amounts of computing resources are compared to discrete simulations.

The different types of simulation can be run in parallel or in series. Parallel execution of simulations provides shorter execution times but increases the complexity of running the simulation model, as well as verifying it.

Simulating a computer network in general is a difficult task and not easy to handle. The main problem is that the computer network is composed of many nodes such as routers, switches, and hosts, making the modeling part of the simulation process a trivial task. There are certain decisions to make at the start of the simulation process:

- What facts does the simulation need to show or prove?
- What are the important elements that should be studied?
- Which simulator provides the best possibilities for modeling the system?
- What simulation is accurate enough to use the results for research?

There are different methods for simulating computer networks,

3.1 Network Simulators

Commonly used network simulators support multiple protocols and therefore provide considerable benefits such as:

- better validation of existing protocols
- an infrastructure for the development of new protocols
- easier comparison of results

for example**: GloMoSim**., **QualNet, SSFNet, OPNET**, **NS-2, and OMNet++**:

3.1.1 OMNet++:

OMNet++ is an event simulator based on the C++ language, primarily intended to simulate network protocols and distributed systems. It is completely programmable, configurable and modular. It is an open source application under the GNU license, developed by Andras Varga, researcher at the University of Budapest. OMNet++ is intended primarily for academic use and is the intermediary between simulation software like NS, intended primarily for research, and OPNET which is a commercial alternative to OMNet++. The operation of OMNet++ relies entirely on the use of modules which communicate with each other through messages. These modules are organized hierarchically. The basic modules are called simple modules. These are grouped into compound modules. These modules can themselves be grouped into compound modules. The number of hierarchical levels is not limited. They are coded in C++ and are instances of the module base type. The architecture is built such that simple modules are both senders and recipients of messages. Compound modules simply relay messages to simple modules transparently. Different parameters can be assigned to the connections between the modules: propagation delays, data rates, error rates, etc.

OMNeT++ offers an Eclipse-based Integrated Development Environment (IDE), a graphical runtime environment, and a wide range of tools, with extensions available for functions like real-time simulation, network emulation, database integration, and SystemC integration. The IDE, built on the Eclipse platform, is enhanced with new editors, views, wizards, and additional features, introducing functionalities for creating and configuring models (NED and ini files), executing batch operations, and analyzing simulation outcomes.

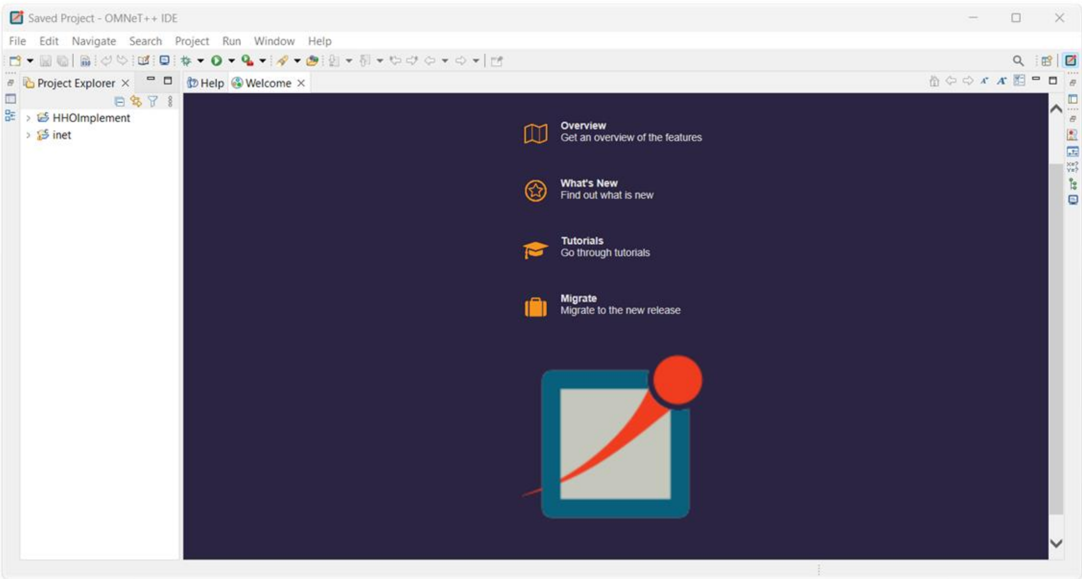

Figure 14 Omnet++ 6.0.2 Interface

3.1.1.1 **INET 4.5.2:**

OMNeT++ offers the INET Framework as a provision. This framework is a standard protocol model library that encompasses models for the Internet stack and a multitude of other protocols and components. It's noteworthy that several other simulation frameworks use INET as a foundational base and extend it in diverse directions.

Figure 15 INET Framework Icon

The INET Framework is a key component of the OMNeT++ simulation environment, providing a wide range of protocols, agents, and models for communication networks. It supports numerous protocols, application models, and link layer models, and facilitates mobile and wireless network simulations. The framework's structure is based on modules that communicate through message passing and are organized into OSI layer-based packages. It also offers interfaces for IEEE 802.11 and includes network interface tables and routing tables for IPv4 and IPv6. It's a valuable tool for designing and validating new protocols or exploring unique scenarios.

4 Experimental Frameworks:

4.1 FANETs Topology:

Our topology is composed of two principal Nodes '**Base Control Station**' and several number '**Drones**', they exchange **data** wirelessly using **Radio** signals, and this data is used to do Drones **Movement** behavior and doing tasks and reach the main objective.

4.1.1 Network Area Creation:

In OMNeT++ we used the INET framework to define nodes and their properties.

The **"Base Control Station"** and the **"Drones"** should have basic Wireless Properties so we chose WirelessHost Node type for both to ensure the communication is well done, The WirelessHost module applies the OSI model by implementing various layers of the model within its structure[20]. Here's how it corresponds to the OSI model:

- **Physical Layer (Layer 1):** The WirelessHost uses the Ieee80211Radio module to represent the physical layer. This module handles the transmission and reception of signals over the wireless medium[20].
- **Data Link Layer (Layer 2):** The Ieee80211Mac module represents the data link layer. It manages access to the shared medium using the CSMA/CA protocol, handles frame retransmissions, acknowledgments, and more[20].
- **Network Layer (Layer 3):** The Ipv4 and Ipv6 modules represent the network layer. They handle IP packet forwarding, routing, and fragmentation[20]. The Tcp, Udp, and Sctp modules represent the transport layer. They provide end-to-end data transfer services[20].
- **Session, Presentation, and Application Layers (Layers 5-7):** These layers are represented by various application modules like UdpBasicApp, TcpSessionApp, etc., that generate network traffic[20].

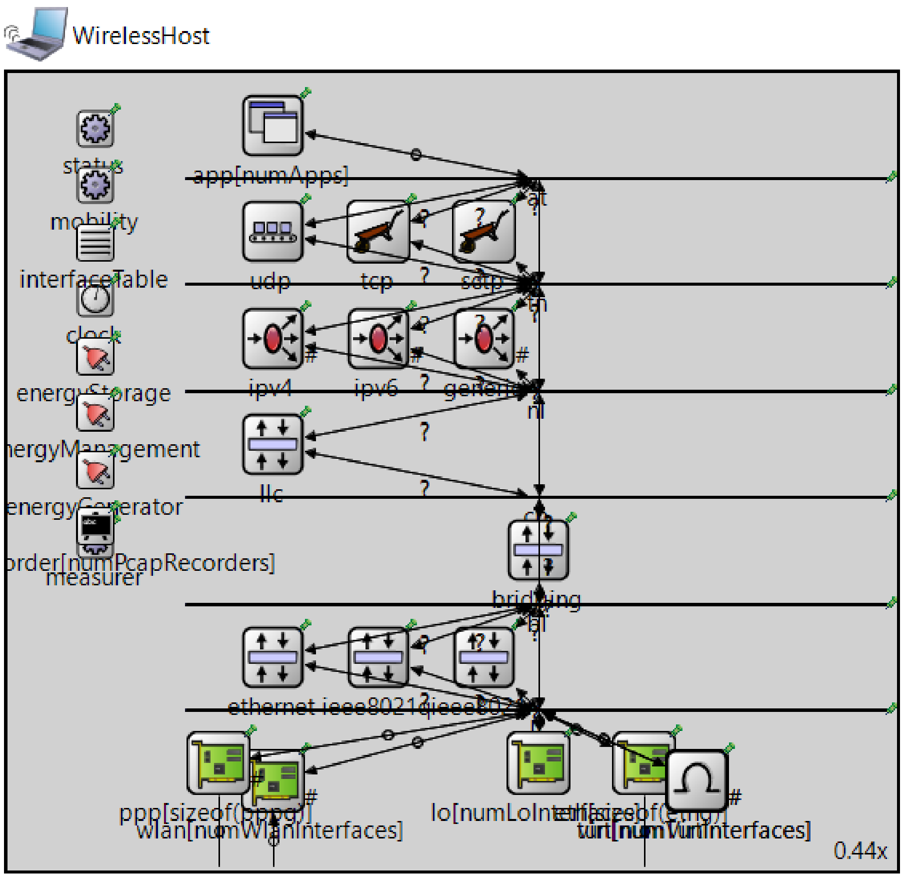

Figure 16 WirelessHost Architecture

4.1.2 Wireless Communication Configuration:

To do this step in OMNeT++ we should:

- Create RadioMedium that represents the physical layer, it is necessary to make WirlessHost do its job, the RadioMedium is a component that models the shared physical medium for wireless communication[21]. It is responsible for **Tracking**, **Computation**, **Interference Management**[22], and more.
- We also should add a visualizer to display sent and received packets and all submodules changes, The IntegratedCanvasVisualizer in OMNeT++ is a comprehensive module that amalgamates all canvas visualizers into a single entity[23], so that's why we use it.
- Also to make Hosts Addressing automatically managed in our simulation we Create Configurator with Ipv4NetworkConfigurator type, It is responsible of **Address Assignment**, **Routing Setup**[24].

The hosts ip address range starts with 10.0.0.1 with the network 10.0.0.0 to all devices in the network, the visualizer is responsible of tracing sending signals.

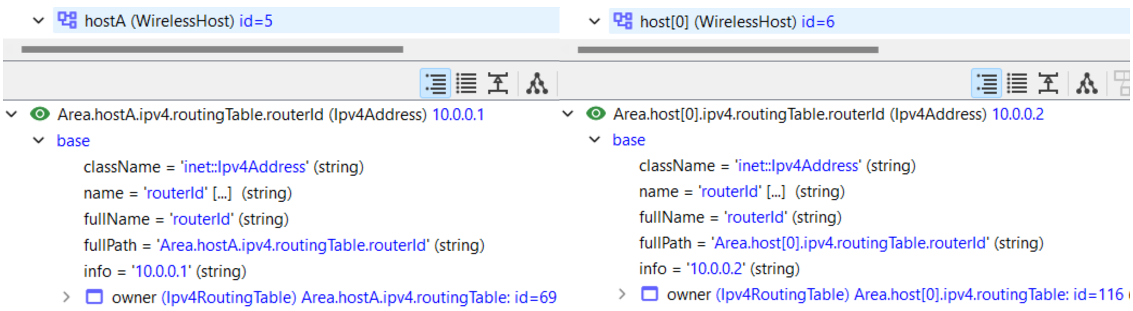

Figure 17 Addressing Informations

As we see in the previous figure that addresses start with 10.0.0.1 to the base station "HostA" and 10.0.0.2 to the Host[0] …

- To send packets from the Base Control Station we will use "UdpBasicApp", UdpBasicApp is a simple module in OMNeT++ used for sending UDP packets to a specified IP address at a given interval[25]. It is compatible with both IPv4 and IPv6[25]. The sending interval can be a constant or a random value (e.g., exponential (1))[25].
- To Receive packets in Drones we will use "UdpSink", "UdpSink" is a simple module in OMNeT++ that consumes and prints packets received from the UDP module[26]. It's

typically used in conjunction with applications that send UDP packets, such as UdpBasicApp[25].

The next figure will show the communication between the hosts.

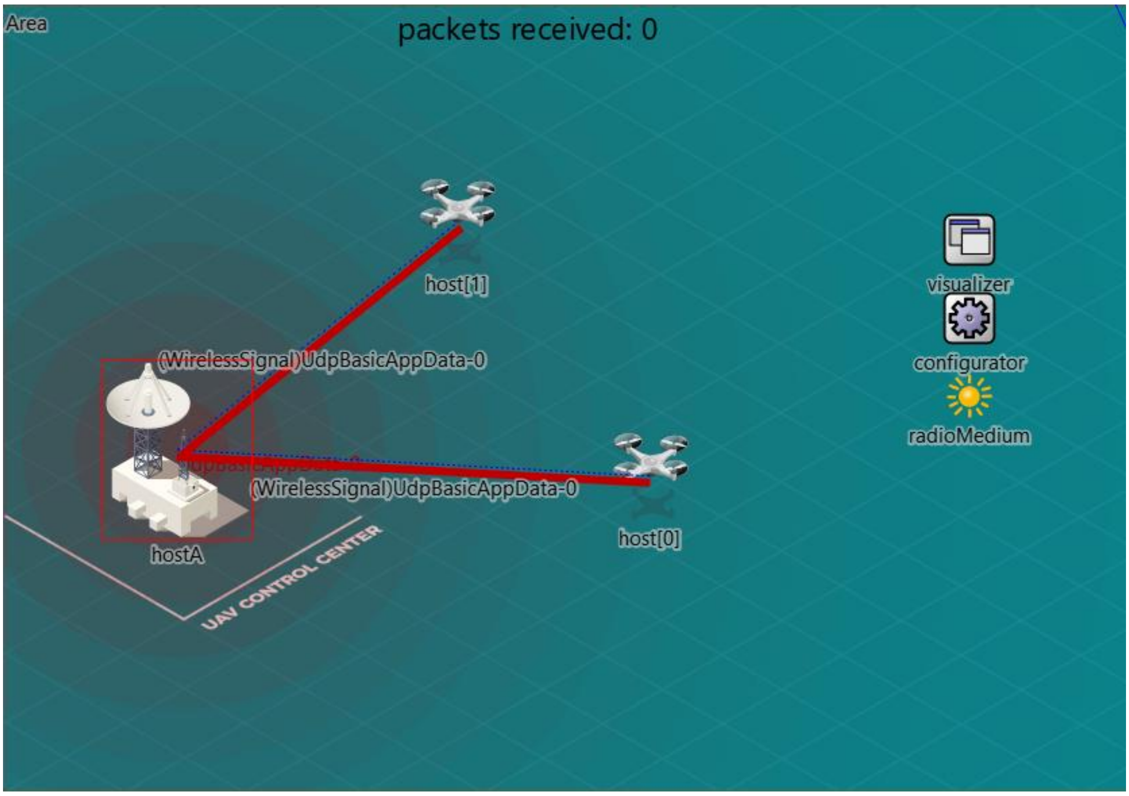

Figure 18 Wireless Communication

4.2 Nodes properties Configuration:

4.2.1 Setting Drones Battery Capacities:

To set Drones battery properties we use this tables power settings:

- **Drones Energy Properties:**

We choose SimpleEpEnergyStorage for the Drones power module to set the battery properties, SimpleEpEnergyStorage is a module in OMNeT++ that models an energy storage system[27]. It maintains a residual energy capacity by integrating the difference between the total consumed power and the total generated power over time[27], [28].

The settings are in the table below:

Power Property	Value
Battery Capacity	0.9 joule
Battery initial State	Fully Charged
Shutdown Percentage Capacity	10 %
Mobility Power Consumption	0.1 (j) every 100 (m)

Table 1 Drones Energy Properties

❖ **Radio Transmission Energy Properties:**

We have chosen StateBasedEpEnergyConsumer to provide transmission power properties, The StateBasedEpEnergyConsumer is a simple module in OMNeT++ that provides a radio power consumer model[29]. The power consumption is determined by the radio mode, the transmitter state, and the receiver state using constant parameters[29].

The table below shows the default power properties [29] of the radio:

Power Property	value
Transmitter Transmitting Power Consumption	100mW
Transmitter Idle Power Consumption	2mW
Receiver Receiving Power Consumption	10mW
Receiver Idle Power Consumption	2mW
Receiver Busy Power Consumption	5mW
Switching Power Consumption	1mW
Sleep Power Consumption	1mW

Table 2 Radio Transmission Energy Properties

To show the battery level in the simulation we use Visualizer next Figure:

Figure 19 Visualizing Power Level

4.2.2 Setting Drones Mobility and Movement Behavior:

As we say, the drones should receive a target area coordination to explore it, we see that "RandomWayPointMobility" with some additional modifications will do the movement of drones.

"RandomWaypointMobility" is a mobility model used in OMNeT++ to simulate the movement of nodes in a network, and is especially useful in wireless ad hoc networks[30]. In this model, each node moves from one waypoint or target to another at a speed that is chosen from a predetermined range in *.mobility.speed parameter [30], [31]. When a waypoint is reached, the node pauses for a specified waiting period in *.mobility.waitTime parameter before moving on to the next randomly chosen waypoint [31]. Waypoints are uniformly distributed within a defined restriction zone[31].

In our work, the waypoints should be specified by the Control Base Station so that's why we will extend the "RandomWayPointMobility" module behavior and add some C++ source code modifications to make the target position settable by specific parameters.

We created a new mobility module in the path " inet\src\inet\mobility\single" with the name "WaypointMobility" with the same parameters of "RandomWayPointMobility" adding to it { mobility.targetX, mobility.targetY, mobility.targetZ } parameters that select the waypoint position and make the node reach it and we also modified the C++ code to use these Coordination's.

```
simple WaypointMobility extends MovingMobilityBase
{
    parameters:
        double initialX @unit(m) = default(uniform(this.constraintAreaMinX, this.constraintAreaMaxX));
        double initialY @unit(m) = default(uniform(this.constraintAreaMinY, this.constraintAreaMaxY));
        double initialZ @unit(m) = default(nanToZero(uniform(this.constraintAreaMinZ, this.constraintAreaMaxZ)));
        bool initFromDisplayString = default(true);
        volatile double speed @unit(mps)@mutable = default(mps);
        volatile double waitTime @unit(s)@mutable = default(0s);
        double targetX @mutable @unit(m) = default(m);
        double targetY @mutable @unit(m) = default(m);
        double targetZ @mutable @unit(m) = default(m);
        @class(WaypointMobility);
}
```

Figure 20 Mobility Source Code Modification

The next tow figures shows the host mobility before and after moving using "WaypointMobility" that we have created.

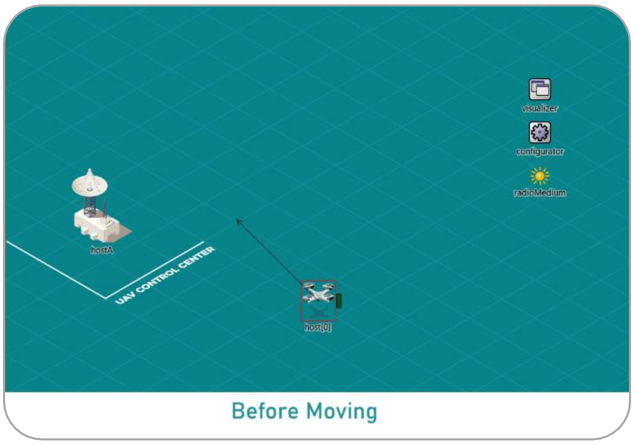

Figure 21 Host Position Before Moving

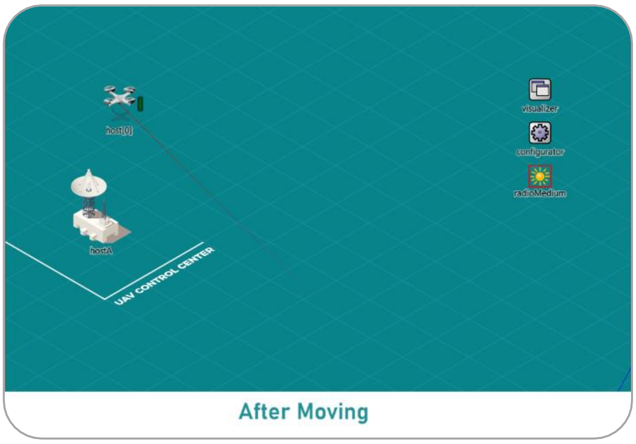

Figure 22 Host Position After Moving

4.3 Drones Selection Behavior and Exploration Orders Sending:

4.3.1 Drones Selection Methods:

By applying the Approach We decided to create three methods that will be the Base Control Station Drones selection behavior, After all of that we see the simulation Results and track the power consumption in each method.

❖ **Method 01: HarrisHowksOptimization**

"HarrisHowksOptimization" is our optimization Base Control Application, we developed this module to get the best available Host that consumes less power to arrive at the exploration area.

The idea is to order the received reports according to the lower power consumption that confirms the reachability to the exploration area by the drone and the highest power level if there is equality in consumption values, taking into account that each drone will be available as long as it does not reach the minimum power level.

❖ **Method 02: RandomSelection**

"RandomSelection" is a random Selection method of Drones for the Base Control Application that we created, it selects Drones to go to the exploration area Randomly.

The idea is to select any available drone that can reach the exploration area without seeing the power consumption or battery level.

❖ **Method 03: HighBatteryLevel**

"HighBatteryLevel" is a module that selects the Drones with the highest power level available and sends them to the exploration area.

The idea is to order the received reports according to the highest battery power Level that confirms the reachability to the exploration area by the drone.

"Action", this module is to receive the Base Control Station orders and to process the movement action, it is based on "UdpSink" module but we added some changes to resend the Power Consumption Rapport to the Base Control Station.

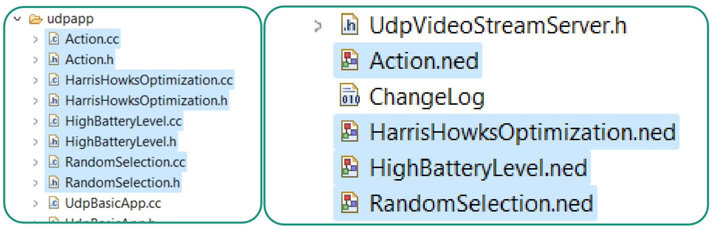

Figure 23 Comunication Modules Files

4.3.2 Creation of Messages Packets and Types Definitions:

To make the Base Control Station Send exploration orders to drones according to simulation consept we create new Message that contain three types (**Area Check, Power Rapport, Area Go**).

- **Area Check:** it is with the Index = 1, this message type contain the coordination of the exploration area that generated randomly, it is sent by the Base Control Station to tell the drone to calculate the Statistics about power consumption according to distance to the area.
- **Power Rapport:** it is with the Index = 2, this message type contain the drone statistics and information's, it's sent by the Drone to the Base Control Station, this rapport used to make decision witch Drone is going to explore the area generated.
- **Area Go:** it is with the Index = 3, this message type is sent after making decision of selection, it's sent by the Base Control Station to the selected Drones only that been chosen to explore the area.

```
class TargetMsg extends FieldsChunk
{
    int MsgType;
    Repport myRepport;
    int x;
    int y;
    int z;

}
```

Figure 24 Message Creating

We create also the Rapport as a class that contain:

- HostName: is the drone Definition.
- HostIndex: is the index of the drone.
- IpAddres: is the drone Ipv4 adderss.
- Consomation: is the drone power consumption To arrive in the Exploration area.
- Speed: is the speed that sited to go to the area.
- ActualPower: is the drone Actual Battery Level .
- Distence: is the distance between the exploration area and the drone actual Coordination's.

- FullPower: is the size of battery.
- ReservedPower: is the power Level if the drone moved to the Exploration area.
- Availability: it is false if the drone's battery level is less then 10%.

The next figure is the Rapport Class definition:

```
class Repport {
protected:
    std::string HostName;
    int HostIndex;
    inet::L3Address IpAdress;
    double Consomation;
    double Speed;
    double ActualPower;
    double Distance;
    double FullPower;
    double ReservedPower;
    bool Avalability;
```

Figure 25 Drones Power Rapport

The files we created to generate massage are represented in the next figure, note that the file with the name Repporting.msg is used to convert the class of Repport to a type in TtargetMsg.msg file:

```
> ApplicationBase.h
> ApplicationPacket_m.cc
> ApplicationPacket_m.h
> ApplicationPacketSerializer.cc
> ApplicationPacketSerializer.h
> Repport.cc
> Repport.h
> Repporting_m.cc
> Repporting_m.h
> TargetMsg_m.cc
> TargetMsg_m.h
  ApplicationPacket.msg
  ChangeLog
  Repporting.msg
  TargetMsg.msg
```

Figure 26 Message Files

the Data part in the message datagram is represented in the next Figure:

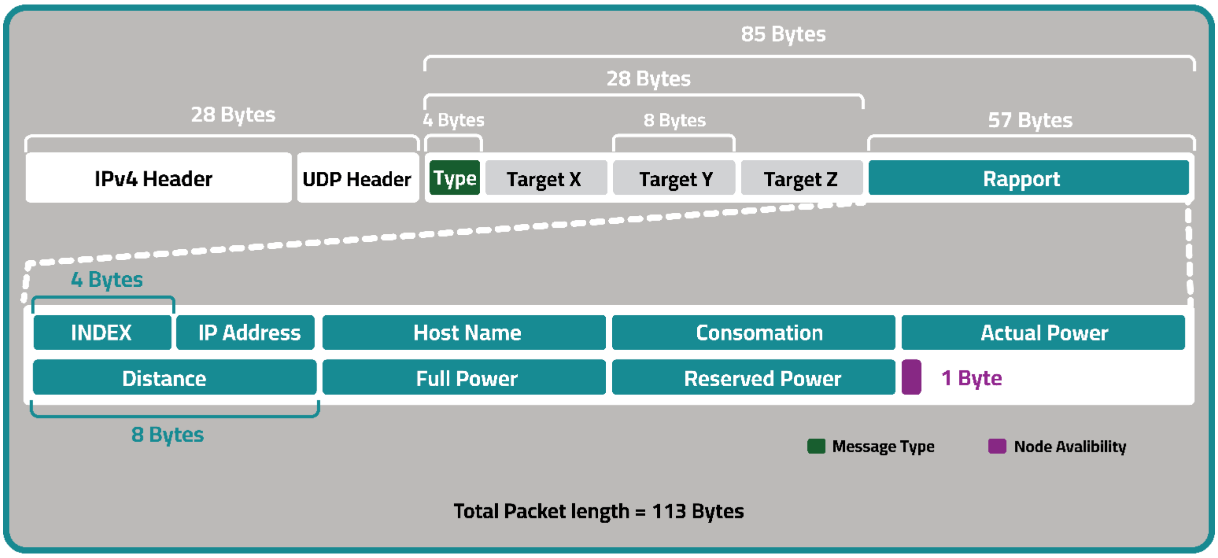

Figure 27 message Details

Here we see the message components and the length of each part of the packet, since we are using IPv4 and UDP protocols The packet should contain the Header of each protocol. The length of the IPv4 Header is 20 bytes and the UDP header is 8 bytes.

The Data part is composed of 5 Parts:

- The first one is the Message Type identifier, it is an integer variable that is represented with 4 Bytes, the values of this identifier are (1= check Area message, 2 = Power Rapport Reply, 3 = Area Go).
- The next three Parts are (Target X, Target Y, Target Z), these are the exploration area Coordination that are generated by the Base Control Station, they are Double Values represented by 8 Bytes.
- The last part Contains the UAVs Power Rapport information, the index and the Ip address is represented by 4 Bytes for each one, availability is a Boolean that is represented by 1 Byte, and the last five informations (Consomation, Actual Power, distance, full power, reserved power) are also double values that represented with 8 Bytes, adding to them the host name that is a string value that does not pass 8 character it is represented by 8 Bytes too.

The total packet length is:

L = length(IPv4 Header + UDP Header) + length(Type) + 3 * length(Target X) + length(Rapport)

L = 28 + 4 + 3*8 + (4*2 + 6*8 + 1) = 28 + 28 + 57 = 113 Bytes

4.4 Results and Data Analyse:

To decide if our Module {Harris Hawks Optimization} is optimized we put the same settings to all three modules { HarrisHowksOptimization, RandomSelection, HighBatteryLevel } and we compare between them.

The comparison will contain the **Power Consumption percentage**, **Reached Aera's Number** during the time lifecycle of the simulation, The **number of Drones Affected in each area**, the **Total Power Consumed**, the **Total Explored Area Number,** and the **largest lifecycle time of** each module.

We put the settings in the next table to each selection Module {HarrisHowksOptimization, RandomSelection, HighBatteryLevel } in simulation:

Settings	Value
Total Drones Number	54
Maximum Number of Drones Sent for Each Area	12
Random Exploration Area Generating Interval	5 sec
Waiting for All Nodes Reply	1 sec
Ending Exploration Phase Lifecycle	When All Drones Battery levels reach 10%

Table 3 Simulation Settings

4.4.1 Power Consumption percentage during the Exploration Phase lifecycle:

for the three modules, the results of power consumption are presented in the next Pivot.

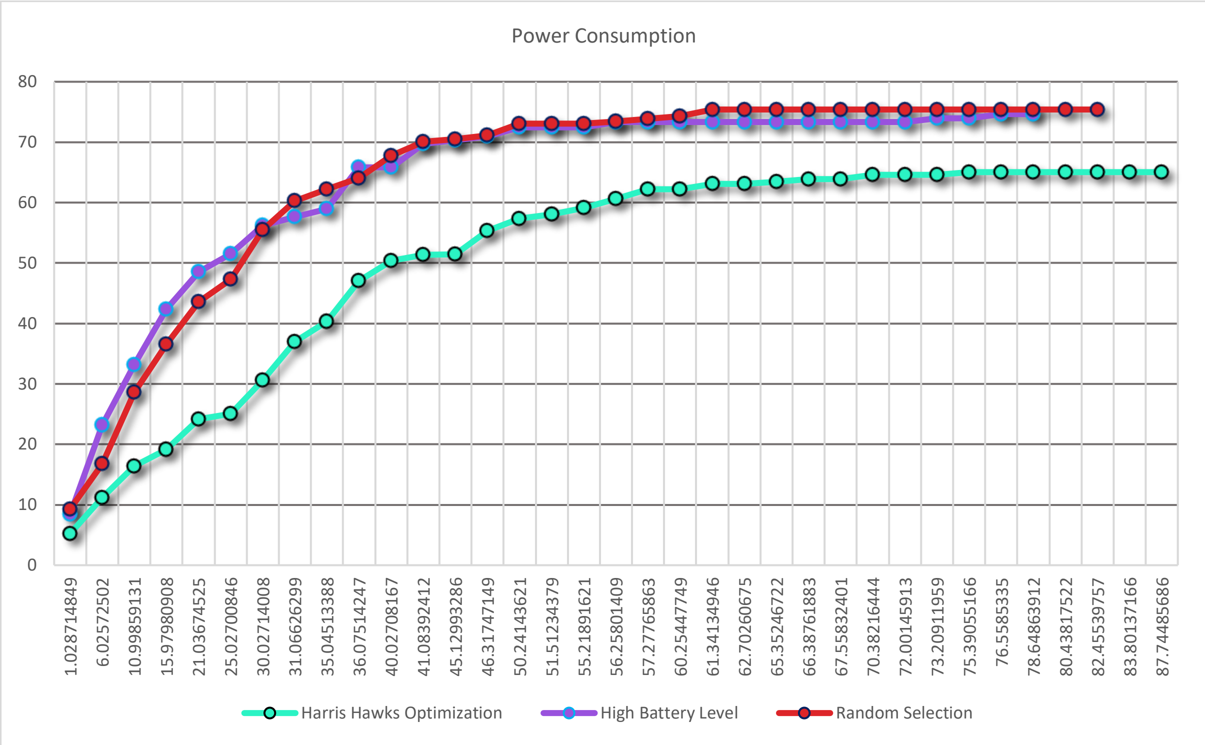

Figure 28 Power Consumption During Exploration Phase Lifecycle

As we see in the power consumption pivot the Harris Hawks Optimization selection module is the lowest one that consumes power during the exploration phase, the Random selection was lower than the High Battery Level module from the beginning until the second 30, after the second 36 it consumes more than the other two modules during the Exploration time lifecycle.

4.4.2 **Reached Aera's Number during the Exploration Phase lifecycle:**

for the three modules, the results of Reached Aera's Number are presented in the next Pivot.

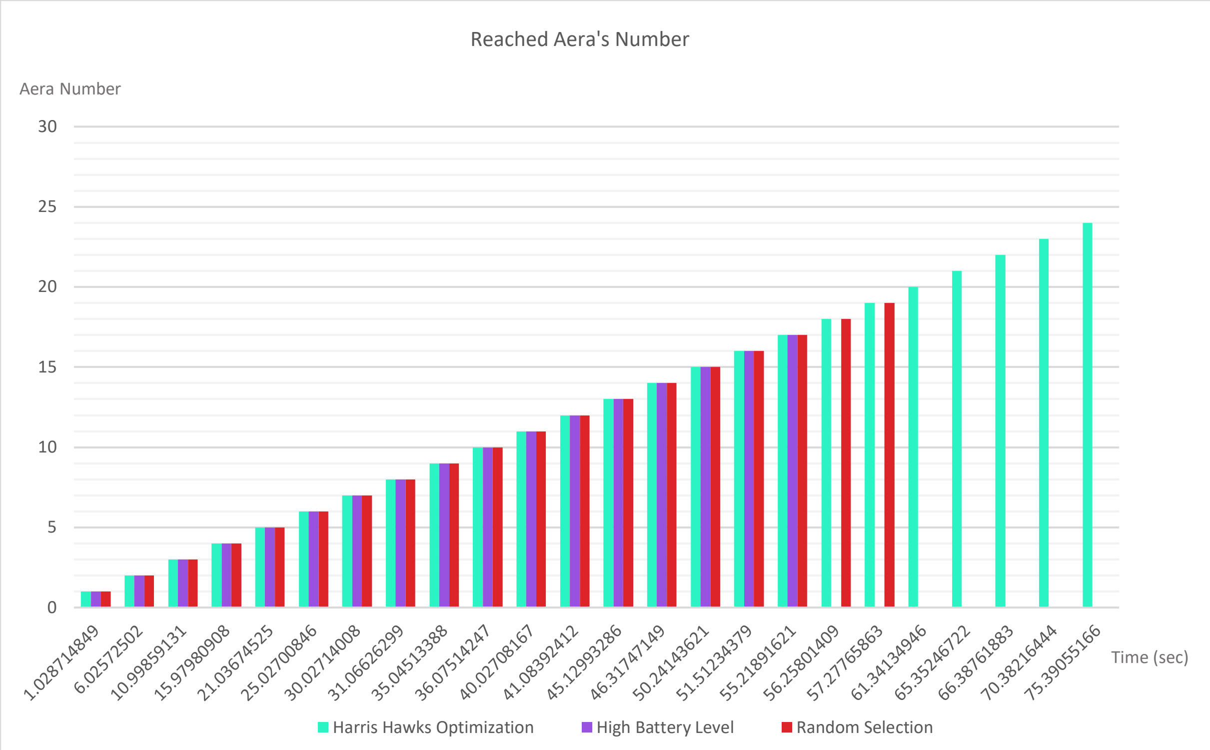

Figure 29 Reached Area's Number During Exploration Phase Time Lifecycle

As we see the three modules have the same Explored area number from the beginning until the second 55, the High Battery level has no more areas to explore, and after two seconds (57) the random Selection module has no more areas to explore too, the Harris Hawks Optimization Selection module stopped to find exploration areas until second 75.

4.4.3 Number of Drones Affected in each area during the Exploration Phase lifecycle:

for the three modules, the results of Number of Drones Affected in each area are presented in the next Pivot.

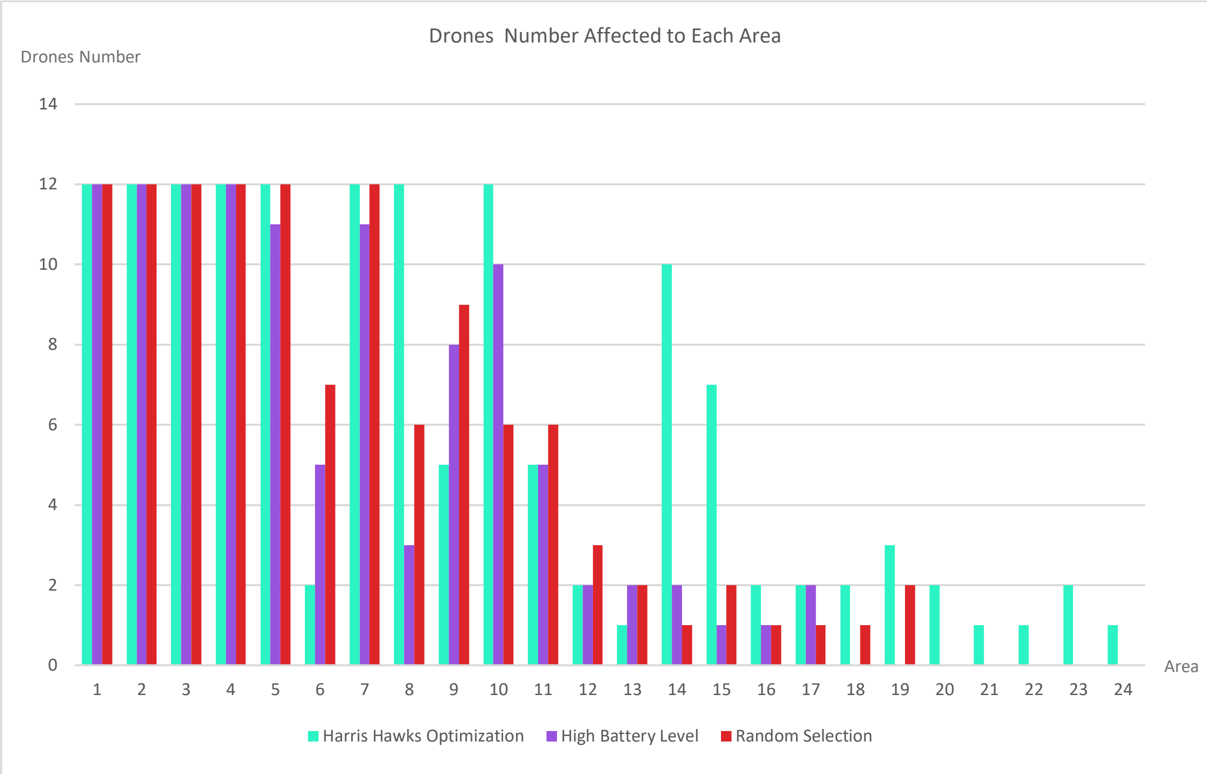

Figure 30 Number of Drones Affected in each area

As we have said in the simulation Settings the maximum number of Drones to explore one area is 12 Drones, we see that the Harris Hawks Optimization Selection module sent the highest number of drones to 8 areas in order to Random selection sends to 6 areas, the High Battery Level Module sends to 4 areas.

4.4.4 Maximum Numbers of Area's Operated:

for the three modules, the results of the maximum number of exploration areas operated are presented in the next Pivot.

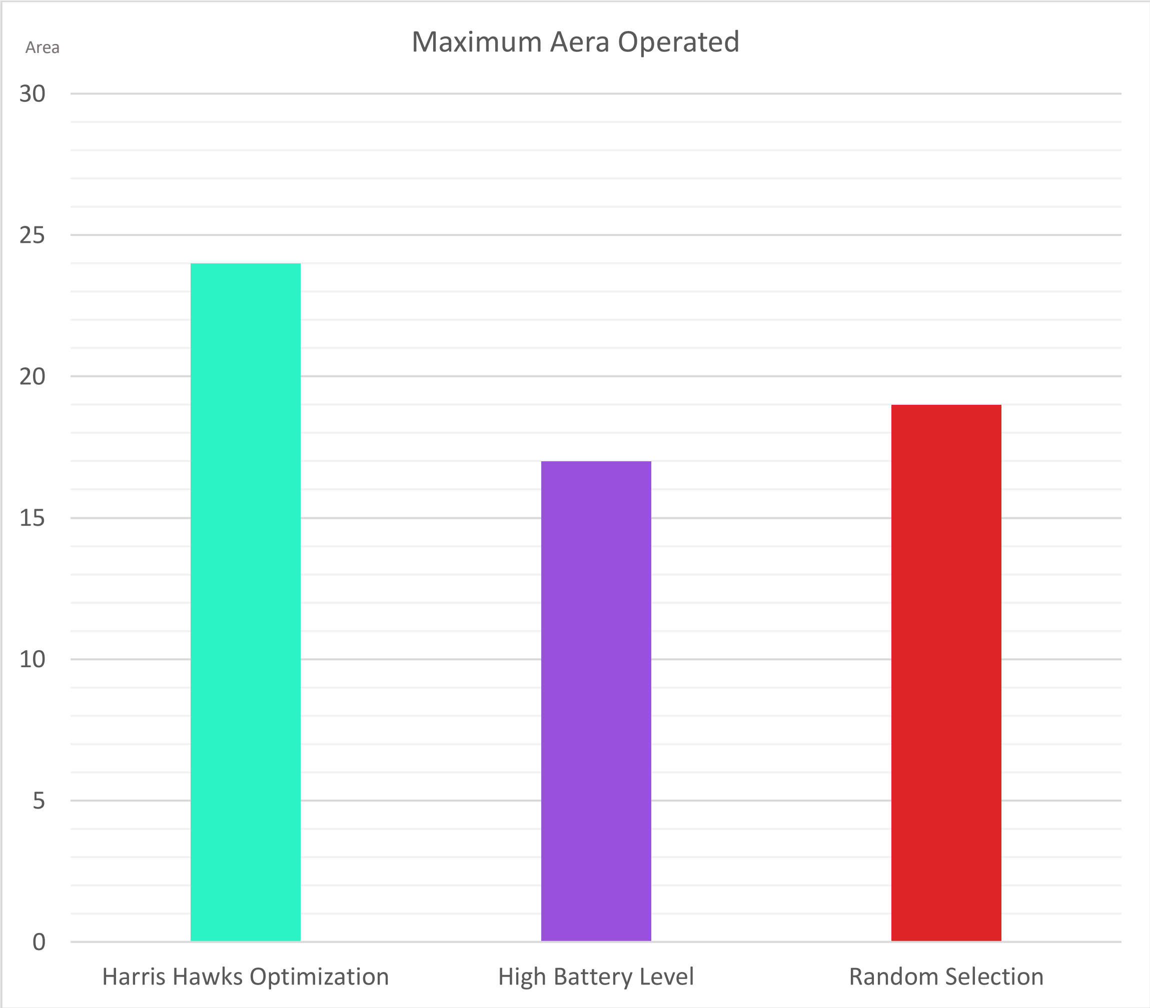

Figure 31 Maximum Exploration areas Operated

We see that the Harris Hawks Optimization Selection module has the highest number of Explored areas 24, after that the Random Selection with 19 areas, after that the High Battery Level selection module with 17 areas.

4.4.5 **Total Power Consomption:**

for the three modules, the results of the Total Power Consumption are presented in the next Pivot.

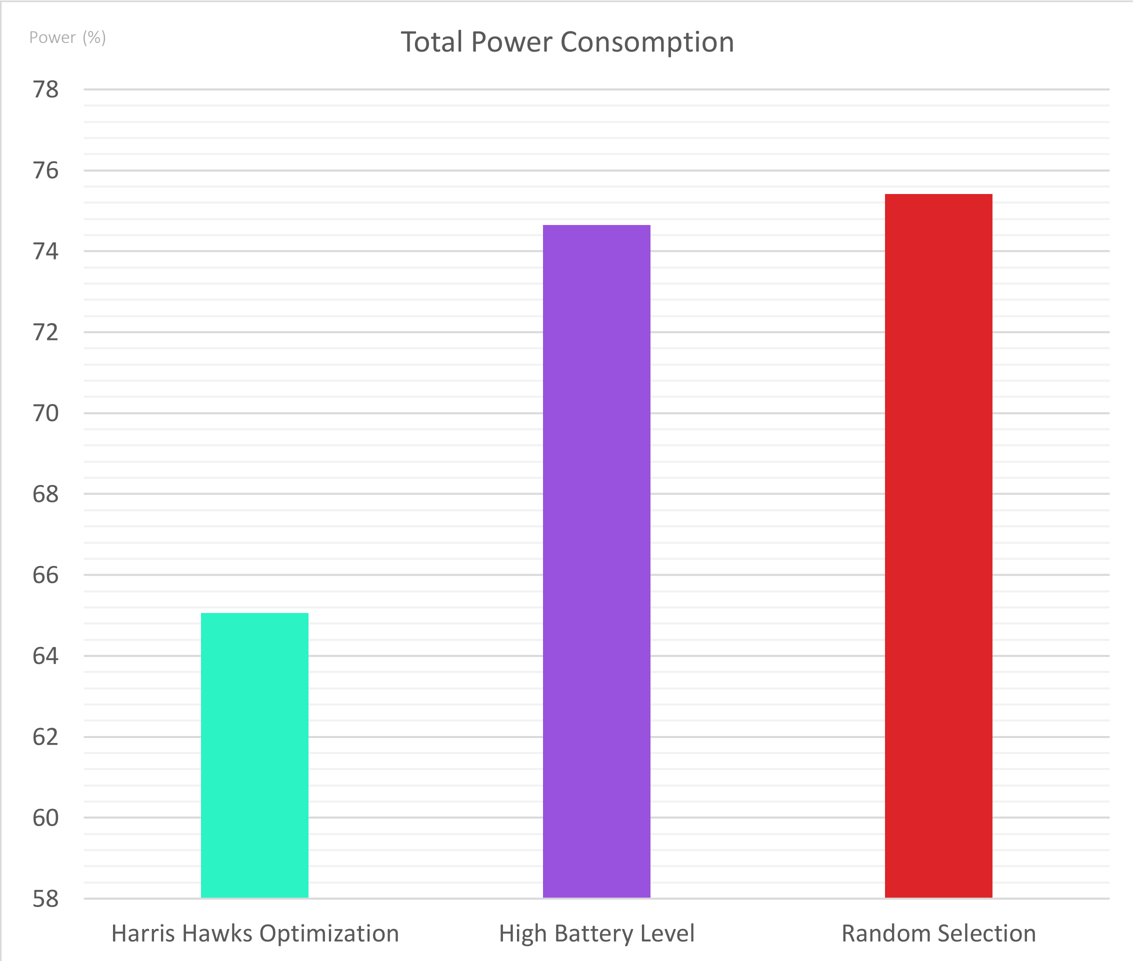

Figure 32 Total Power Consomption

We see that the Harris Hawks Optimization Selection module is the lower one in the total Power consumption with 65%, after it the High Battery Level Selection with 74.6%, and then the Random Selection module with 75.4% of power consumption.

4.4.6 Total Time Lifecycle:

for the three modules, the results of the Total Time Lifecycle are presented in the next Pivot.

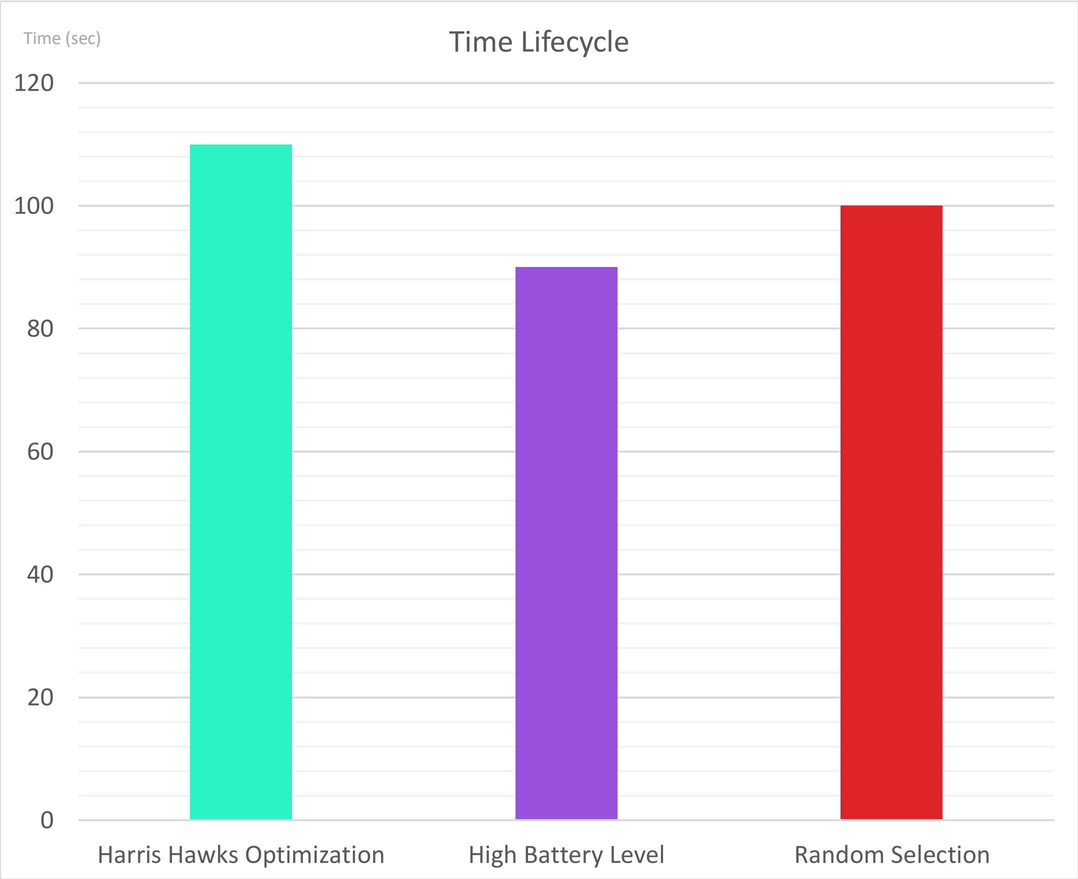

Figure 33 Exploration Phase Time Lifecycle

As we see that the Time Lifecycle of Harris Hawks Optimization selection is the Bigger one with 110 seconds, then the Random Selection with 100 seconds, after that the High Battery Level selection with 90 seconds.

5 Conclusion:

The experimental setups discussed in the previous chapter provide important new information about the topology, node property setting, drone selection behaviors, and exploration order dispatch in Flying Ad-hoc Networks (FANETs). These trials yielded data that allowed for a comprehensive understanding of drone behavior and performance in FANETs. By using this data, researchers can make informed decisions to increase FANETs' operational efficacy and efficiency in a variety of applications.

GENERAL CONCLUSION

V General Conclusion

Flying Ad Hoc Networks (FANETs) have emerged as a cutting-edge network type, enabling communication and collaboration among drones in various fields. The optimization of communication and task allocation in FANETs is crucial for enhancing operational efficiency and maximizing mission success. Through the utilization of different selection methods like RandomSelection, HighBatteryLevel, and the Harris Hawks Optimization algorithm, we aim to determine the most effective approach for drone selection in FANET task scheduling.

Based on the analysis of the methods, the Harris Hawks Optimization Method stands out as the most promising approach for drone selection in FANET task scheduling. By mimicking optimizes ensuring efficient and effective operations like reducing energy consumption, reaching the largest number of explored areas, and also extending the life of the exploration process...

Harris Hawks Optimization Method provides the best results in the simulation which means that the integration of this method in the exploration phase of Harris Hawks Optimization algorithm in FANETs in real life can significantly enhance performance, range, and operational capabilities, opening up new possibilities for drone applications across various domains.

1 Challenges

Several challenges arise in FANETs:

- **Dynamic Topology**: The network topology changes rapidly due to UAV mobility, affecting communication links and task allocation.
- **Energy Constraints**: Drones operate on limited battery power, necessitating efficient energy management.
- **Task Allocation**: Assigning tasks to drones optimally while considering their capabilities, energy levels, and communication range is a complex problem.

2 Future Directions

Potential future directions include:

- **Variant HHO Algorithms**: Explore new variants of HHO tailored specifically for FANETs.

- **Real-World Deployment**: Apply the optimized task allocation approach to real-world FANET scenarios.
- **Benchmarking**: Compare HHO's performance with other optimization techniques in FANETs.

BIBLIOGRAPHY

VI Bibliography

[1] "What Is a Wireless Network? Types of Wireless Network," Fortinet. Accessed: May 16, 2024. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/wireless-network

[2] "Wireless Network Infrastructure | GlobalSpec." Accessed: May 16, 2024. [Online]. Available: https://www.globalspec.com/reference/81402/203279/wireless-network-infrastructure

[3] "Ad Hoc Networks." Accessed: May 16, 2024. [Online]. Available: https://www.scimagojr.com/journalsearch.php?q=26799&tip=sid

[4] "Introduction of Mobile Ad hoc Network (MANET)," GeeksforGeeks. Accessed: May 16, 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-of-mobile-ad-hoc-network-manet/

[5] "Vehicular ad hoc networks (VANETS): status, results, and challenges | Telecommunication Systems." Accessed: May 16, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11235-010-9400-5

[6] J. Wang and C. Jiang, "Introduction of Flying Ad Hoc Networks," in *Flying Ad Hoc Networks: Cooperative Networking and Resource Allocation*, J. Wang and C. Jiang, Eds., Singapore: Springer, 2022, pp. 1–10. doi: 10.1007/978-981-16-8850-8_1.

[7] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[8] H. M. Alabool, D. Alarabiat, L. Abualigah, and A. A. Heidari, "Harris hawks optimization: a comprehensive review of recent variants and applications," *Neural Comput. Appl.*, vol. 33, no. 15, pp. 8939–8980, Aug. 2021, doi: 10.1007/s00521-021-05720-5.

[9] M. Shehab *et al.*, "Harris Hawks Optimization Algorithm: Variants and Applications," *Arch. Comput. Methods Eng.*, vol. 29, no. 7, pp. 5579–5603, Nov. 2022, doi: 10.1007/s11831-022-09780-1.

[10] L. Chen, C. Feng, and Y. Ma, "Improved Harris Hawks optimization for global optimization and engineering design," *Clust. Comput.*, vol. 27, no. 2, pp. 2003–2027, Apr. 2024, doi: 10.1007/s10586-023-04020-y.

[11] "Nature Inspired Optimization Techniques, A review for FANETs," *Sukkur IBA J. Emerg. Technol.*, vol. 3, no. 2, Dec. 2020, doi: 10.30537/sjet.v3i2.716.

[12] V. Sharma, D. G. Reina, and R. Kumar, "HMADSO: a novel hill Myna and desert Sparrow optimization algorithm for cooperative rendezvous and task allocation in FANETs," *Soft Comput.*, vol. 22, no. 18, pp. 6191–6214, Sep. 2018, doi: 10.1007/s00500-017-2686-4.

[13] M. Osmanpoor, A. Shameli-Sendi, and F. Faraji Daneshgar, "Convergence of the Harris hawks optimization algorithm and fuzzy system for cloud-based task scheduling enhancement," *Clust. Comput.*, Jan. 2024, doi: 10.1007/s10586-023-04225-1.

[14] R. Sharma and S. Prakash, "HHO-LPWSN: Harris Hawks Optimization Algorithm for Sensor Nodes Localization Problem in Wireless Sensor Networks," *ICST Trans. Scalable Inf. Syst.*, vol. 8, p. 168807, Feb. 2021, doi: 10.4108/eai.25-2-2021.168807.

[15] X. Xue, R. Shanmugam, S. Palanisamy, O. I. Khalaf, D. Selvaraj, and G. M. Abdulsahib, "A Hybrid Cross Layer with Harris-Hawk-Optimization-Based Efficient Routing for Wireless Sensor Networks," 2023.

[16] A. Ali, F. Aadil, M. F. Khan, M. Maqsood, and S.-K. Lim, "Harris Hawks Optimization-Based Clustering Algorithm for Vehicular Ad-Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, 2023, Accessed: Jun. 12, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Harris-Hawks-Optimization-Based-Clustering-for-Ali-Aadil/2e464d81288123a2d544a88b31be1fff4d7bff09

[17] P. Satyanarayana, U. Sofi, B. Ahmed, N. Saranya, G. Rao, and D. V Gokula Krishnan, "Enhancement of Network Security in MANET Using Refined Adaptive Harris Hawks Optimization Algorithm (RAHHO) for IoT Applications," Mar. 2024, pp. 1–5. doi: 10.1109/ESCI59607.2024.10497254.

[18] M. A. Hossain *et al.*, "Multi-Objective Harris Hawks Optimization Algorithm Based 2-Hop Routing Algorithm for CR-VANET," *IEEE Access*, vol. 9, pp. 58230–58242, 2021, doi: 10.1109/ACCESS.2021.3072922.

[19] "End-to-End Quality of Service Over Heterogeneous Networks Braun, Torsten," DLS liefert Lehrmittel. Accessed: Jun. 12, 2024. [Online]. Available: https://gibs-solothurn.webdls.ch/en/detail/ISBN-9783540791195/Braun-Torsten/End-to-End-Quality-of-Service-Over-Heterogeneous-Networks

[20] "WirelessHost." Accessed: May 10, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.node.inet.WirelessHost.html

[21] "RadioMedium." Accessed: May 10, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.physicallayer.wireless.common.medium.RadioMedium.html

[22] "The Transmission Medium — INET 4.5.0 documentation." Accessed: May 10, 2024. [Online]. Available: https://inet.omnetpp.org/docs/users-guide/ch-transmission-medium.html

[23] "IntegratedCanvasVisualizer." Accessed: May 10, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.visualizer.canvas.integrated.IntegratedCanvasVisualizer.html

[24] "IPv4 Network Configurator Tutorial — INET 4.5.0 documentation." Accessed: May 10, 2024. [Online]. Available: https://inet.omnetpp.org/docs/tutorials/configurator/doc/index.html

[25] "UdpBasicApp." Accessed: May 12, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.applications.udpapp.UdpBasicApp.html

[26] "UdpSink." Accessed: May 12, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.applications.udpapp.UdpSink.html

[27] "SimpleEpEnergyStorage." Accessed: May 11, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.power.storage.SimpleEpEnergyStorage.html

[28] "Power Consumption — INET 4.5.0 documentation." Accessed: May 11, 2024. [Online]. Available: https://inet.omnetpp.org/docs/showcases/wireless/power/doc/index.html

[29] "StateBasedEpEnergyConsumer." Accessed: May 11, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.physicallayer.wireless.common.energyconsumer.StateBasedEpEnergyConsumer.html

[30] "Node Mobility — INET 4.5.0 documentation." Accessed: May 12, 2024. [Online]. Available: https://inet.omnetpp.org/docs/users-guide/ch-mobility.html

[31] "RandomWaypointMobility." Accessed: May 12, 2024. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/inet.mobility.single.RandomWaypointMobility.html