

الجمهورية الجزائرية الديمقراطية
الشعبية

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي والبحث
العلمي

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

جامعة سعيدة - د. الطاهر مولاي

UNIVERSITY OF SAIDA - Dr. MOULAY TAHAR

Faculty of Mathematics, Computer Science and Telecommunications



A Dissertation Submitted in Partial Fulfilment of the Requirements for a master's Degree in
Telecommunications

Specialization: Networks and Telecommunications

Presented by: M. AOUIMEUR Abdallah
M. AKKAL Azel-Islem

Drone Placement and Energy Optimization for Flying Ad Hoc Networks Coverage via Deep Reinforcement Learning

Defended on 13th June, 2026 in front of the jury composed of:

M. Boudkhil Abdelhakim	MCA	President
Ms. Ouessai Asmâa	MCB	Supervisor
M. Ouardi Aissa	MCA	Examiner

2025 / 2026

Acknowledgements

We wish to express our deep gratitude to all those who contributed to the completion of this final year project.

A special thank you to our final year project supervisor, Dr. Ouessai Asmâa, for her invaluable guidance, unwavering support, and expertise.

We thank Dr. Boudkhil and Dr. Ouardi for agreeing to serve on the jury and for the time they dedicated to evaluating our work.

We would like to express our sincere gratitude to Dr. Boudkhil Abdelhakim, Head of the Telecommunications Department, for his valuable support and interest in our work.

We are also grateful to the teaching staff at Dr. Moulay Tahar University, as well as our classmates for their collaborative spirit.

Finally, we wish to express our gratitude to our families and friends for their moral support, patience, and encouragement during the most intense moments of this project.

This final year project has been an enriching and formative experience, and it is thanks to everyone's involvement that we were able to successfully complete this important stage of our academic journey.

Dedication

I bow before Almighty God who opened the door of knowledge for me and helped me to pass through it.

I dedicate this modest work:

To my dear parents and grandparents, a source of affection, courage, and inspiration, who sacrificed so much to see me reach this day.

A special dedication to our final year project supervisor, Ms. Ouessai Asmâa

To the entire AKKAL and AOUIMER families

To my sisters

My brothers

My nieces

My nephews

All my friends

ISLEM

Dedication

I bow before Almighty God who opened the door of knowledge for me and helped me to pass through it.

I dedicate this modest work:

To my dear parents and grandparents, a source of affection, courage, and inspiration, who sacrificed so much to see me reach this day.

A special dedication to our final year project supervisor, Ms. Ouessai Asmâa

To the entire AOUIMER and AKKAL families

To my sisters

My brothers

My nieces

My nephews

All my friends

ABDALLAH

Abstract

The rapid evolution of wireless communication technologies has driven the emergence of Flying Ad Hoc Networks (FANETs), where multiple Unmanned Aerial Vehicles (UAVs) cooperate to provide coverage in infrastructure-deprived environments. A critical challenge in FANETs is the strict energy constraint imposed by limited battery capacity, which directly affects network lifetime and mission success. This thesis investigates the use of Deep Reinforcement Learning (DRL), specifically Deep Q-Network (DQN) architectures, to jointly optimize UAV placement and energy consumption in a three-dimensional FANET environment. Three DQN variants (DQN-3L, DQN-4L, DQN-5L) are trained over 12,000 episodes within a custom Gymnasium-based simulation using an energy-aware reward function defined as the ratio of navigation progress to energy consumed. A K-Means clustering algorithm initializes UAV positions at user-density centroids to accelerate convergence. Results demonstrate that DQN-5L achieves the best performance: 84.3% mission success rate, 22.3% reduction in energy consumption, and a 36.2% improvement in energy efficiency compared to the shallowest architecture. Deterministic inference testing confirms 100% goal-reaching accuracy with an energy efficiency metric of 0.375. Multi-UAV validation with $K=2$ drones confirms the scalability of the proposed framework for cooperative FANET deployments.

Keywords: FANET, UAV, Deep Reinforcement Learning, DQN, Energy Optimization, K-Means Clustering, 3D Placement.

Résumé

L'évolution rapide des technologies de communication sans fil a favorisé l'émergence des Réseaux Ad Hoc Volants (FANETs), où plusieurs véhicules aériens sans pilote (UAVs) coopèrent pour fournir une couverture dans des environnements dépourvus d'infrastructure. L'un des défis majeurs des FANETs est la contrainte énergétique stricte imposée par la capacité limitée des batteries, qui affecte directement la durée de vie du réseau et le succès des missions. Ce mémoire étudie l'utilisation de l'Apprentissage par Renforcement Profond (DRL), et plus particulièrement des architectures Deep Q-Network (DQN), pour optimiser conjointement le placement des UAVs et leur consommation énergétique dans un environnement FANET tridimensionnel. Trois variantes DQN (DQN-3L, DQN-4L, DQN-5L) sont entraînées sur 12000 épisodes dans un environnement de simulation basé sur Gymnasium, avec une fonction de récompense énergétique définie comme le rapport entre la progression de navigation et l'énergie consommée. Un algorithme de clustering K-Means initialise les positions des UAVs aux centroïdes de densité d'utilisateurs pour accélérer la convergence. Les résultats montrent que DQN-5L atteint les meilleures performances : un taux de succès de 84,3 %, une réduction de 22,3 % de la consommation énergétique, et une amélioration de 36,2 % de l'efficacité énergétique par rapport à l'architecture la plus simple. La validation multi-UAV avec $K=2$ drones confirme la scalabilité du cadre proposé.

Mots-clés : FANET, UAV, Apprentissage par Renforcement Profond, DQN, Optimisation Énergétique, K-Means, Placement 3D.

ملخص

أبرز التطور السريع في تقنيات الاتصالات اللاسلكية ظهور الشبكات المتنقلة الجوية (FANETs)، حيث تتعاون مجموعة من الطائرات غير المأهولة (UAVs) لتوفير تغطية الإتصال في البيئات المحرومة من البنية التحتية. ويتمثل أبرز تحديات هذه الشبكات في القيود الطاقية الصارمة الناجمة عن محدودية طاقة البطارية، مما يؤثر مباشرة على عمر الشبكة ونجاح المهام. تستلزم هذه الرسالة استخدام التعلم معزز العميق (DRL)، وتحديدًا بنية شبكة Q العميقة (DQN)، لتحسين توزيع الطائرات غير المأهولة واستهلاكها الطاقوي في بيئة ثلاثية الأبعاد. تم تدريب ثلاثة نماذج DQN (DQN-3L, DQN-4L, DQN-5L) عبر 12,000 حلقة باستخدام دالة مكافأة واعية بالطاقة، مع تهيئة مواضع أولية باستخدام خوارزمية K-Means. أثبتت النتائج تفوق نموذج DQN-5L بتحقيقه نسبة نجاح 84.3%، وتخفيض استهلاك الطاقة بنسبة 22.3%، وتحسين كفاءة الطاقة بنسبة 36.2%. كما أثبت اختبار الاستدلال الحتمي دقة 100% في الوصول إلى الهدف، مما يؤكد فعالية الإطار المقترح لنشر طائرات ثير مأهولة في شبكات FANET متعددة العملاء.

الكلمات المفتاحية: FANET، UAV، التعلم معزز العميق، DQN، تحسين استهلاك الطاقة، تجميع K-Means، توزيع ثلاثي الأبعاد.

Summary

<i>General Introduction</i>	18
<i>Chapter 1: Flying Ad Hoc Networks</i>	20
1.1 Introduction	21
1.2 Overview of Ad Hoc Networks	22
1.3 Types of Ad Hoc Network	23
1.3.1. MANET	23
A. MANET Applications	23
B. MANET Challenges	24
1.3.2. VANET	24
A. VANET Applications	25
A.1 Safety Applications	25
A.2 Traffic efficiency applications	26
A.3 Infotainment and others	26
B. VANET Challenges	26
1.3.3. Sea Ad Hoc Networks (SANETs)	27
A. Applications of SANETs	28
B. Challenges and Issues in SANETs	28
1.3.4 Flying Ad Hoc Networks (FANETs)	28
A. Characteristics of FANETs	30
A.1. Node Density	30
A.2 Node Mobility	30
A.3 Changing network topology	30
A.4. Radio propagation model	30
A.5. Localization	31
A.6. Power consumption and network lifetime	31
A.7. Frequency band	31
B. Applications of FANETs	31
B.1 Military Operations	31
B.2 Environmental Monitoring	31
B.3. Post-Disaster Communication	32
C. Research issues and challenges	32
C.1. Routing	32
C.2. Path Planning	33

C.3. Quality of Service (QoS).....	33
1.4. Conclusion.....	33
<i>Chapter 2: UAV-assisted wireless communication networks</i>	34
2.1 Introduction	35
2.2 Role of UAVs in Wireless Communications	35
2.2.1 UAVs as Aerial Base Stations (UAV-BS).....	36
2.2.2 UAVs as Wireless Relay Nodes (UAV-RS)	36
2.2.3 UAVs as User Equipment (UAV-UE)	36
2.3 UAV Deployment and Coverage Models	37
2.3.1 Air-to-Ground LoS Probability and Optimal Altitude.....	37
2.3.2 Single UAV Deployment.....	38
2.3.3 Multiple UAV Deployment.....	38
2.4 Air-to-Ground Communication.....	39
2.4.1 Large-Scale Fading: Path Loss Models	39
2.4.2 Small-Scale Fading.....	40
2.4.3 Doppler Effects and Channel Non-Stationarity.....	40
2.5 Energy Constraints in UAV Networks.....	41
2.5.1 Propulsion Energy Model.....	41
2.5.3 Energy-Aware Trajectory and Resource Optimization.....	42
2.5.4 Energy Replenishment Strategies	42
2.6 Conclusion.....	43
<i>Chapter 3: Deep Reinforcement learning</i>	45
3.1 Introduction	44
3.2 Reinforcement Learning	46
3.3 Deep Learning.....	49
3.4 Deep reinforcement Learning (DRL) Models for UAV Placement optimization	50
3.4.1 Deep Q-Network (DQN).....	50
3.4.2 Deep Deterministic Policy Gradient (DDPG)	51
3.4.3 Proximal Policy Optimization (PPO).....	51
3.4.4 Soft Actor-Critic (SAC).....	51
3.4.5 Multi-Agent DRL: MADDPG and MAPPO	52
3.4.6 Comparative Analysis	52
3.4.7 Related work	53
3.5 Conclusion.....	53
<i>Chapter 4: Simulation and results</i>	55
4.1 Introduction	56
4.2 Simulation Environment	56

4.2.1 Development Environment	56
a. Google Colaboratory.....	56
b. Visual Studio Code and Local Workstation	56
4.2.2 Programming Language	56
a. Python	56
b. ReactJS	57
4.2.3 Libraries Used	57
a. NumPy.....	57
b. Gymnasium	57
c. PyTorch	57
d. PyTorch Neural Network Module.....	57
e. PyTorch Optimization Module.....	57
f. Random	58
g. Deque	58
h. Matplotlib	58
i. Axes3D.....	58
4. 3 Scenario description	58
4.4 Performance Metrics.....	59
4.4.1 UAV Power Consumption Model.....	59
4.4.2 Energy Consumption	61
4.4.3 Euclidean Distance	61
4.4.4 Energy Efficiency Metric	61
4.4.5 Reward Function	62
4.4.6 Deep Q-Learning Update Equation	62
4.4.7 Spatial Optimization via K-Means Clustering for Initial Drone Placement	63
4.4.7.1 Mathematical Framework and Cluster Anchor Selection	63
4.4.7.2 Integration with the DQN-5 State Space	64
4.5 Results and Analysis	64
Part I – DQN Architectures Comparison	64
4.5.1 Experimental Setup and Architecture Configurations	64
4.5.2 Comparative Performance Results	66
4.5.3 Reward Convergence Analysis	67
4.5.4 Energy Consumption Comparison	68
4.5.5 Success Rate and Navigation Efficiency	68
4.5.6 Energy Efficiency Metric η Comparison.....	69
4.5.7 Summary and Model Selection.....	70

Part II – Detailed DQN-5 Analysis (12 000 Episodes)	70
4.5.8 Training Phases Over 12 000 Episodes	70
4.5.9 Aerodynamic Energy Consumption & Kinematic Allocation	72
4.5.10 Inference Testing and Policy Evaluation	73
4.5.11 Optimized 3D Trajectory Analysis	74
4.5.12 Multi-UAV Deployment and K-Means Target Allocation Validation	75
4.5.13 Discussion	77
4.6 Conclusion	78
<i>General Conclusion</i>	80
REFERENCES	81

List of figures

Figure 1: Organization of the dissertation	19
Figure 1-1: Infrastructure-based and infrastructure-less networks.....	21
Figure 1-2 : Types of Ad Hoc Network	22
Figure 1-3 : Creating an Ad-hoc Network using Vehicles (VANETs)	25
Figure 1-4 : Taxonomy of VANET applications	25
Figure 1-5 : Sea Ad Hoc Network (SANET)	27
Figure 1-6 : Flying Ad Hoc Network	29
Figure 1-7 : Flying Ad Hoc Network	29
Figure 2-1 : UAV assisted wireless communication	35
Figure 3-1 : Reinforcement Learning	46
Figure 3-2 : Markov decision process	46
Figure 3-3 : Feedforward Deep Learning Model	50
Figure 4-1 : The simulation scenario used in this work, where the UAV acts as an aerial base station providing wireless coverage to multiple ground users distributed within the communication area.	59
Figure 4-2 : Kmeans clustering principle.....	63
Figure 4-3 : Training reward evolution with 2000-episode moving average for three DQN architectures.....	67
Figure 4-4 : Average energy consumption per episode (2000-episode moving average) for the three DQN architectures.....	68
Figure 4-5 : Success rate (%) and average steps per episode for the three DQN architectures.....	69
Figure 4-6 : Energy efficiency η (100-episode moving average) for the three DQN architectures	70
Figure 4-7 : DQN-5 training dashboard: reward, energy, steps, success rate, η, loss, and ϵ over 12 000 episodes.....	72
Figure 4-8 : Quadrotor power model components vs. flight speed (energy_breakdown)	73
Figure 4-9 : Best episode 3D UAV trajectory produced by DQN-5; color encodes step progression from start (lime) to goal (red)	75
Figure 4-10 : Multi-UAV Kinematic Convergence and Energy Dissipation Profiles Over Simulation Steps	76

List of tables

Table 1.1 : WIFI 802.11n standards.....	29
Table 2.1 : Comparative Summary of UAV Roles in Wireless Communications	37
Table 2.2 : Single vs. Multiple UAV Static Deployment	39
Table 2.3 : A2G Channel Models Summary.....	41
Table 2.4 : UAV Energy Optimization Strategies	43
Table 3.1 : Comparative analysis of DRL algorithms for UAV placement in FANET.	53
Table 3.2 : Summary of related works on DRL-based UAV placement	53
Table 4.1 : DQN Network Topologies : Layer Configuration and Parameter Count.	65
Table 4.2 : Training Hyperparameters Common to All DQN Architectures	65
Table 4.3 : Comparative Performance Summary of DQN Architectures (averaged over last 100 episodes of 2 000-episode runs)	66
Table 4.4 : Energy consumption model parameters.....	67
Table 4.5 : DQN-5 Training Phase Analysis Over 12 000 Episodes	71
Table 4.6 : DQN-5 Energy Breakdown by Movement Type (averaged over last 100 episodes)	72
Table 4.7 : DQN-5 Inference Test Results Across Three Independent Episodes	73
Table 4.8 : Final Deployment Summary for K-Means Target Allocation and DQN-5 Navigation	76

List of Abbreviations

2D: Two-Dimensional

3D: Three-Dimensional

4G: Fourth Generation Cellular Network

5G: Fifth Generation Cellular Network

A2G: Air-to-Ground

AC: Actor-Critic

AF: Amplify-and-Forward

AGPS: Assisted Global Positioning System

AI: Artificial Intelligence

AIS: Automatic Identification System

AWGN: Additive White Gaussian Noise

BS: Base Station

C2 : Command and Control

CI: Close-In (*Path Loss Model*)

CNN : Convolutional Neural Network

CPU: Central Processing Unit

DDPG : Deep Deterministic Policy Gradient

DGPS: Differential Global Positioning System

DF: Decode-and-Forward

DGPS: Differential Global Positioning System

DQN: Deep Q-Network

DRL: Deep Reinforcement Learning

EE: Energy Efficiency

EMTAD: Energy-Efficient Multi-UAV Traffic-Aware Deployment

FANET: Flying Ad Hoc Network

FSPL: Free-Space Path Loss

GCS : Ground Control Station

GPU : Graphics Processing Unit

GPS: Global Positioning System

GSM: Global System for Mobile Communications

GU: Ground User

IEEE: Institute of Electrical and Electronics Engineers

IoT: Internet of Things

IoV: Internet of Vehicles (*referenced in Figure 2*)

ITU-R: International Telecommunication Union – Radiocommunication Sector

LiPo: Lithium-Polymer (*Battery technology*)

LoS: Line-of-Sight

MAD-DDQN: Multi-Agent Deep Double Q-Network

MADDPG: Multi-Agent Deep Deterministic Policy Gradient

MANET: Mobile Ad Hoc Network

MAPPO: Multi-Agent Proximal Policy Optimization

MDP: Markov Decision Process

MILP: Mixed-Integer Linear Programming

ML : Machine Learning

MLP: Multilayer Perceptron

MOS: Mean Opinion Score

NLoS: Non-Line-of-Sight

PAN: Personal Area Network

PG: Policy Gradient

PGW: Packet Data Network Gateway (*referenced in Figure 2*)

PLE: Path Loss Exponent

POMDP: Partially Observable Markov Decision Process

PPO: Proximal Policy Optimization

QoE: Quality of Experience

QoS: Quality of Service

RBM: Restricted Boltzmann Machine

RF: Radio Frequency

RIS: Reconfigurable Intelligent Surface

RL: Reinforcement Learning

RNN: Recurrent Neural Network

RSU: Road Side Unit

SAC: Soft Actor-Critic

SANET: Sea Ad Hoc Network / Sensor Ad Hoc Network

SCA: Successive Convex Approximation

SINR: Signal-to-Interference-plus-Noise Ratio

SINR: Signal-to-Interference-plus-Noise Ratio

SVM : Support Vector Machine

TR-PPO-RB: Trust Region-based Proximal Policy Optimization with Rollback

U2G: UAV-to-Ground

U2U: UAV-to-UAV

UAV: Unmanned Aerial Vehicle

UAV-BS: Unmanned Aerial Vehicle - Aerial Base Station

UAV-RS: Unmanned Aerial Vehicle - Wireless Relay Node

UAV-UE: Unmanned Aerial Vehicle - User Equipment

UE: User Equipment

UMTS: Universal Mobile Telecommunications System

USV: Unmanned Surface Vehicle

V2I: Vehicle-to-Infrastructure

V2V: Vehicle-to-Vehicle

VANET: Vehicular Ad Hoc Network

VS Code : Visual Studio Code

WCSS: Within-Cluster Sum of Squares

WiFi: Wireless Fidelity

WLAN: Wireless Local Area Network

WPT: Wireless Power Transfer

WSN: Wireless Sensor Network

WSS: Wide-Sense Stationary

General Introduction

General Introduction

The fast evolution of wireless communication technologies has led to the emergence of very dynamic and intelligent networking paradigms that can work in environments where the traditional communication infrastructure is absent or insufficient. Of these paradigms, the formation of Flying Ad Hoc Networks (FANETs) based on multiple Unmanned Aerial Vehicles (UAVs) to communicate cooperatively via wireless links has received considerable attention from academia and industry. FANETs have unique benefits including fast deployment, flexible mobility, extended coverage, and the capability to support communication services in disaster-stricken areas, remote areas, military operations, environmental monitoring, and emergency response scenarios.

FANETs have many advantages but also many critical challenges. UAVs have high mobility, leading to frequent topology changes and unstable communication links and complex routing requirements. Moreover, limited battery capacity of UAVs imposes strict energy constraints, which directly affects the network lifetime, communication reliability, and mission success. Since propulsion energy and communication energy are the dominant sources of power consumption, efficient energy management has emerged as one of the most important research issues in FANET design and operation.

Recent advances in Artificial Intelligence (AI), particularly in Machine Learning (ML) and Reinforcement Learning (RL), have provided new solutions for tackling complex optimization problems in wireless networks. Deep Reinforcement Learning (DRL), which combines the decision-making capability of Reinforcement Learning with the representational capacity of Deep Neural Networks, has achieved remarkable success in dynamic and high-dimensional environments. DRL permits UAVs to learn the optimal policy by themselves through interaction with the environment without requiring explicit mathematical models.

In the context of FANETs, DRL offers an efficient framework for solving energy optimization issues through the combined optimization of UAV placement, trajectory planning, transmission power allocation and resource management. By constantly learning and adapting, DRL-based approaches can significantly improve network coverage, quality of service, and energy efficiency, decrease operating costs, and extend the length of UAV flights.

The primary objective of this thesis is to investigate the use of Deep Reinforcement Learning techniques for placement and energy optimization in Flying Ad Hoc Networks. The purpose of this work is to investigate the architecture and challenges of FANET, analyze the UAV assisted wireless communication system, and investigates the DRL algorithms capable of optimizing the UAV deployment and the energy consumption in the dynamic environment.

Organization of the dissertation:

Our work is divided into four chapters:



Figure 1: Organization of the dissertation

Chapter 1: Describes the basics of Ad Hoc Networks, including MANETs, VANETs, SANETs and FANETs, with their characteristics, applications and challenges.

Chapter 2: Covers UAV-enabled wireless communication networks, UAV placement and coverage models, air-to-ground communication channels, and energy limitations in UAV operations.

Chapter 3: Introduces Reinforcement Learning and Deep Reinforcement learning concepts and reviews of the state-of-the-art DRL algorithms applied to UAV placement and energy optimization problems.

Chapter 4: Introduces the proposed DRL-based framework for energy optimization in FANETs, including the system modeling, the simulation setup, the performance evaluation, and the analysis of the obtained results.

In this work we aim to demonstrate how Deep Reinforcement Learning can be used to develop smarter, more energy efficient and autonomous FANET systems that will be able to meet the demands of future wireless communication networks.

Chapter 1: Flying Ad Hoc Networks

1.1 Introduction

Wireless networks are being increasingly used for communication among devices of varied types and sizes. Personal computers, handheld devices, telephones, appliances, industrial machines, and sensors are widely deployed in diverse environments such as residences, buildings, cities, forests, and battlefields. In recent years, various wireless network standards and technologies have emerged to facilitate the easy deployment of applications.

However, in environments where fixed infrastructure cannot be deployed or reliably maintained, conventional wireless networking approaches become inadequate. This limitation has motivated the development of alternative communication paradigms that can operate autonomously and adapt to dynamic conditions. An ad hoc network is an autonomous system of wireless nodes that communicate directly with each other without relying on any fixed infrastructure [1]. The nodes dynamically form a network whenever communication is required and exchange data using wireless radio links. In such networks, each node acts not only as an end system but also as a router, forwarding data for other nodes. The network operates in a fully distributed manner without the presence of a centralized access point or base station. Ad hoc networks enable automatic and flexible connectivity among participating nodes [2].

Wireless network architectures can be broadly classified into two categories: infrastructure-based and infrastructure-less networks. In infrastructure-based architectures, nodes communicate through fixed physical entities such as access points or base stations, as illustrated in Fig. 1.1. Common examples of such networks include GSM, UMTS, and Wireless Local Area Networks (WLANs).

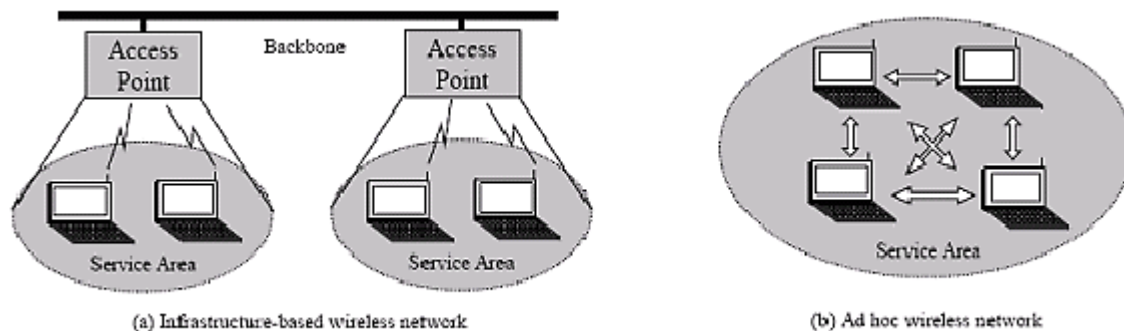


Figure 1-1: Infrastructure-based and infrastructure-less networks

In contrast, infrastructure-less networks operate without any fixed physical infrastructure, and communication is established directly among nodes. This category is particularly suitable for highly dynamic or unpredictable environments. Ad hoc networks fall under this classification, where terminals are connected through a multi-hop, distributed architecture [1]. Due to the absence of a centralized control structure, nodes in an ad hoc network function not only as end devices but also as routers, forwarding data to enable end-to-end communication [3]. Furthermore, the dynamic nature of ad hoc networks eliminates single points of failure, thereby enhancing network robustness.

1.2 Overview of Ad Hoc Networks

Ad hoc networks represent one of the most active and rapidly evolving research areas in wireless communications. Research in this domain has been conducted for nearly three decades, initially under terms such as packet radio and multi-hop networks. In recent years, increased availability of low-cost wireless devices and growing interest in mobile computing have significantly accelerated research and development in this field.

An ad hoc network is a collection of wireless communication devices, referred to as nodes, that dynamically establish communication without relying on any fixed infrastructure or centralized control. Nodes autonomously discover neighboring nodes and collaborate to forward data through multi-hop routing, as direct communication between all nodes is generally not feasible. This cooperative behavior enables communication over extended distances beyond the direct transmission range of individual nodes. Due to node mobility and power constraints, ad hoc networks experience frequent changes in topology and link characteristics.

Ad hoc networking is inherently a cross-layer challenge, as rapid variations in connectivity affect all layers of the communication stack. Consequently, efficient coordination among physical, medium access, network, transport, and application layers is required to ensure reliable communication. Owing to their infrastructure-less and self-organizing nature, ad hoc networks are particularly suitable for environments where traditional communication infrastructure is unavailable, unreliable, or impractical, such as military operations, emergency response and environmental monitoring.

Based on node mobility and application requirements, ad hoc networks can be classified into several categories (figure 1.2), including Mobile Ad Hoc Networks (MANETs), Vehicular Ad Hoc Networks (VANETs), Sensor Ad Hoc Networks (SANETs), and Flying Ad Hoc Networks (FANETs).

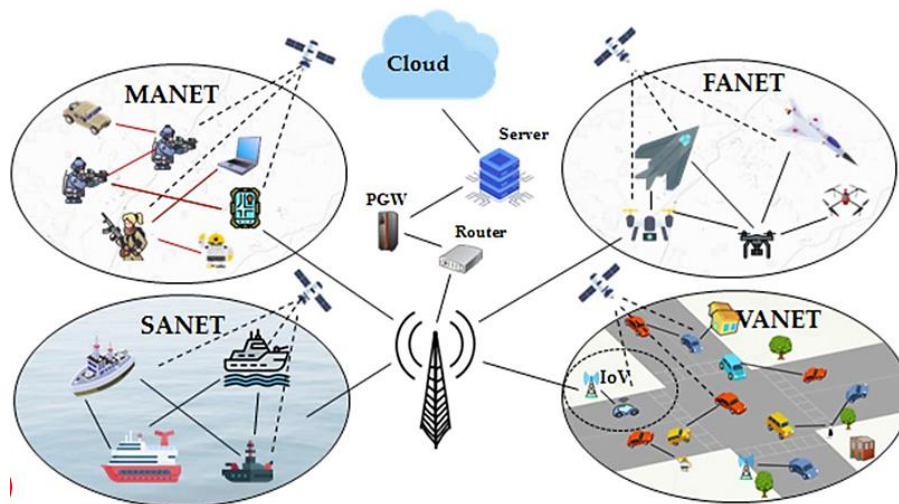


Figure 1-2: Types of Ad Hoc Network

1.3 Types of Ad Hoc Network

1.3.1. MANET

Mobile Ad Hoc Networks (MANETs) represent one of the most extensively studied areas in wireless networking research due to the numerous challenges they introduce to communication protocols. A MANET is an infrastructure-less wireless network that enables mobile devices to communicate with each other without relying on any fixed infrastructure, regardless of their geographical location. For this reason, MANETs are often referred to as self-organizing and decentralized networks.

The rapid proliferation of low-cost, compact, and increasingly powerful wireless devices has significantly contributed to the growth and popularity of MANETs. In such networks, each device must be capable of discovering the presence of neighboring nodes and performing the necessary configurations to establish communication and share data or services. Nodes can join or leave the network dynamically, which allows high flexibility but also introduces additional complexity.

Due to node mobility, the network topology in MANETs may change rapidly and unpredictably over time. As a result, routing becomes a critical challenge in this decentralized environment, where each node is responsible not only for data transmission but also for forwarding packets on behalf of other nodes. Unlike static networks, determining and maintaining optimal routing paths in MANETs is difficult because links may frequently break or be newly established.

MANETs support a wide range of applications, from large-scale, highly dynamic mobile networks to small, relatively static networks with limited power resources. In addition to traditional applications adapted from infrastructure-based networks, MANETs also enable the development of novel services tailored to dynamic and decentralized environments.

However, MANETs are more vulnerable than wired networks due to several factors, including node mobility, lack of centralized management, limited physical security, scalability issues, and the presence of potentially compromised nodes within the network. These characteristics make MANETs more susceptible to various security threats and malicious attacks.

A. MANET Applications

With the increase of portable devices as well as progress in wireless communication, ad-hoc networking is gaining importance with the increasing number of widespread applications. Ad-hoc networking can be applied anywhere where there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use. Ad hoc networking allows the devices to maintain connections to the network as well as easily adding and removing devices to and from the network. The set of applications for MANET is diverse, ranging from large-scale, mobile, highly dynamic networks, to small, static networks that are constrained by power sources. Besides the legacy applications that move from traditional infrastructure environment into the ad hoc context, a great deal of new services can and will be generated for the new environment. Typical applications include [2, 4]:

- Military Battlefield
- Commercial Sector
- Local Level
- Personal Area Network (PAN)

B. MANET Challenges

Regardless of the attractive applications, the features of MANET introduce several challenges that must be studied carefully before a wide commercial deployment can be expected. These include [5, 4]:

- Routing
- Security and Reliability
- Quality of Service (QoS)
- Inter-networking
- Multicast
- Location-aided Routing

1.3.2. VANET

Although Vehicular Ad Hoc Networks (VANETs) are not a new research topic, they continue to introduce significant challenges and attract ongoing research interest. VANETs can be considered a specialized form of Mobile Ad Hoc Networks (MANETs), where vehicles act as mobile nodes and establish communication without relying on a centralized base station or controller.

The primary objective of VANETs is to enable vehicles to dynamically form and maintain a communication network that supports information exchange in real time. One of the most critical applications of VANETs arises in emergency situations, such as medical or accident scenarios, where communication infrastructure may be unavailable and timely information dissemination is essential for saving human lives.

Due to the absence of fixed infrastructure, VANETs impose additional responsibilities on vehicles. Each vehicle participates as both a communication endpoint and a network node that assists in managing and forwarding data. Vehicular ad hoc networks support communication among moving vehicles within a given environment. A vehicle may communicate directly with another vehicle through Vehicle-to-Vehicle (V2V) communication or interact with infrastructure components such as Road Side Units (RSUs) via Vehicle-to-Infrastructure (V2I) communication. A typical VANET communication scenario is illustrated in Figure 1.3.

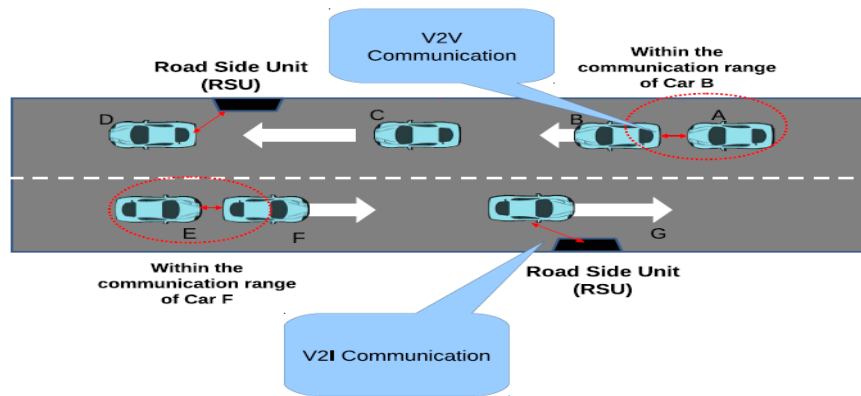


Figure 1-3: Creating an Ad-hoc Network using Vehicles (VANETs)

A. VANET Applications

According to [6], VANET applications can be classified into three main categories: safety applications, traffic efficiency applications, and infotainment applications. Figure 1.4 presents a taxonomy of VANET applications along with representative use cases.

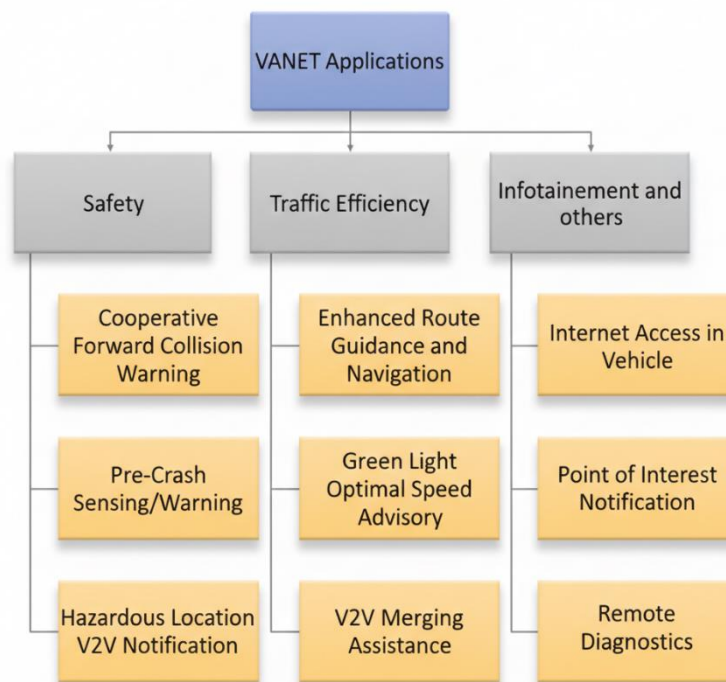


Figure 1-4 : Taxonomy of VANET applications

A.1 Safety Applications

Safety applications aim to improve road safety by providing timely warnings to drivers about potentially dangerous situations. These applications are designed to reduce accidents and enhance situational awareness. Several safety-related use cases and their requirements are discussed in [7]. Key examples include:

- Cooperative Forward Collision Warning
- Pre-Crash Sensing and Warning
- Hazardous Location V2V Notification

A.2 Traffic efficiency applications

Traffic efficiency applications focus on improving transportation system performance by delivering real-time traffic information to drivers and road operators. To achieve this objective, traffic-related data must be exchanged efficiently through VANETs. These applications help reduce travel time, traffic congestion, and road infrastructure costs. Some representative use cases introduced by the CAR-2-CAR Communication Consortium [7] include:

- Enhanced Route Guidance and Navigation
- Green Light Optimal Speed Advisory
- V2V Merging Assistance

A.3 Infotainment and others

Applications that do not directly relate to safety or traffic efficiency fall into this category. These applications provide entertainment, information services, or support vehicle maintenance and operation. Some of these applications are transparent to drivers but play an important role in enhancing vehicle functionality. Typical examples include:

- Internet Access in Vehicles
- Point of Interest Notifications
- Remote Diagnostics

B. VANET Challenges

Despite the numerous advantages offered by VANETs, several challenges must be addressed to enable their widespread adoption. These challenges represent open research issues that require further investigation and development. The main challenges include:

- High node mobility and rapidly changing network topology
- Data administration and storage
- Security and privacy concerns
- Quality of Service (QoS) provisioning
- Heterogeneity and standardization
- Efficient routing protocol design

1.3.3. Sea Ad Hoc Networks (SANETs)

Sea Ad Hoc Networks (SANETs) consist of maritime nodes such as ships, boats, vessels, unmanned surface vehicles (USVs), and underwater vehicles that cooperate to form a large-scale wireless ad hoc network. The primary objective of SANETs is to extend connectivity in aquatic and maritime environments where conventional communication infrastructure is unavailable or impractical [8].

In SANETs (figure 1.5), node density is considered moderate, as nodes are typically dispersed over large oceanic or water-body areas. Compared to other ad hoc networks, the mobility of SANET nodes is higher than that of Mobile Ad Hoc Networks (MANETs) but lower than that of Vehicular Ad Hoc Networks (VANETs) and Flying Ad Hoc Networks (FANETs). The movement of underwater and surface vehicles is often random and unpredictable, resulting in highly dynamic mobility models and frequent topology changes.

Due to the operation of nodes on the water surface, line-of-sight communication among nodes is generally high, which can improve link quality. However, SANETs experience relatively high energy consumption because of long communication distances and continuous mobility. SANETs typically operate in the frequency band between 5 and 8 GHz.

Accurate localization in SANETs presents a significant challenge. Traditional GPS systems may not provide sufficient accuracy in maritime environments; therefore, enhanced localization technologies such as Assisted GPS (AGPS), Differential GPS (DGPS), and the Automatic Identification System (AIS) are commonly employed. Given the wide range of supported applications and the inherent communication delays in aquatic environments, SANETs require stable and efficient multi-hop synchronization mechanisms to ensure reliable data transmission among nodes [9].

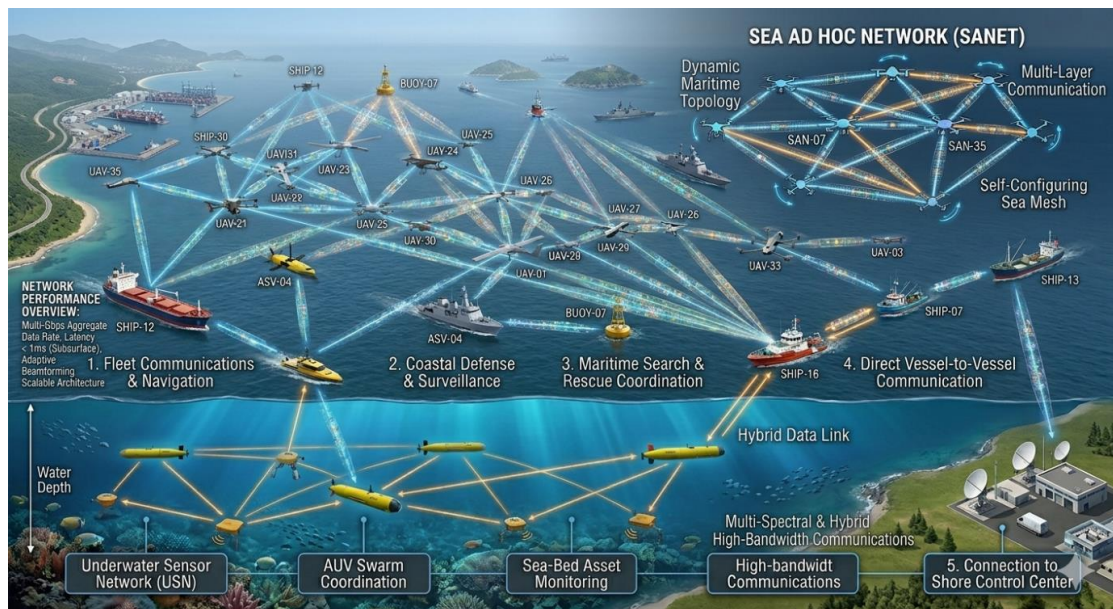


Figure 1-5: Sea Ad Hoc Network (SANET)

A. Applications of SANETs

SANETs support a wide range of applications in maritime and underwater environments, including:

- Catastrophe and disaster management
- Seismic monitoring
- Environmental and oceanographic monitoring
- Military and naval operations
- Underwater robotic systems

B. Challenges and Issues in SANETs

Despite their advantages, SANETs face several technical challenges that must be addressed to ensure efficient and reliable operation:

- Accurate GPS localization and positioning [10]
- Security and data protection
- Peer-to-peer communication reliability
- Energy supply and power efficiency
- Routing challenges in dynamic maritime environments

1.3.4 Flying Ad Hoc Networks (FANETs)

Unmanned Aerial Vehicles (UAVs), also known as drones, are being extensively utilized for wireless communication purposes. These drones can operate autonomously either under the control of a ground operator or through onboard computers. Unmanned Aerial Vehicles are utilized for a wide array of missions, encompassing both military operations and agricultural uses. A flying ad hoc network (FANET) of UAVs can be established by utilizing a cluster of airborne UAVs [11]. FANET refers to a group of mobile drones that continuously modify their network structure without relying on any permanent infrastructure. These drones communicate with a central base station from various altitudes [12], as illustrated in Figures 1.6 and 1.7. The categorization of UAVs used in different FANET scenarios can be determined by various factors, including size, range weight, endurance, altitude, application, flying mechanism, ownership, airspace class, level of control (autonomy), and type of engine [11]. In order to establish communication between drones, it is necessary to utilize an IEEE 802.11 wireless adapter capable of transmitting on many frequencies, including the 2.4 GHz, 5 GHz, and 6 GHz frequency bands. The Wireless Fidelity (WiFi) Alliance employed a numbering system to designate different generations of WiFi protocols, namely 802.11b, 802.11a, 802.11g, 802.11n, 802.11ac, and 802.11ax (Table 1). Some of these versions have the capability to operate on both the 2.4 GHz and 5 GHz frequency bands simultaneously [13].

In this section, we provide a review of FANET communication networks, applications, and limitations, with the aim of elucidating the issues related to connecting multiple UAVs within

ad hoc networks. Since FANETs are dynamic and decentralized in nature, their deployment requires a high level of coordination due to the complexity involved.

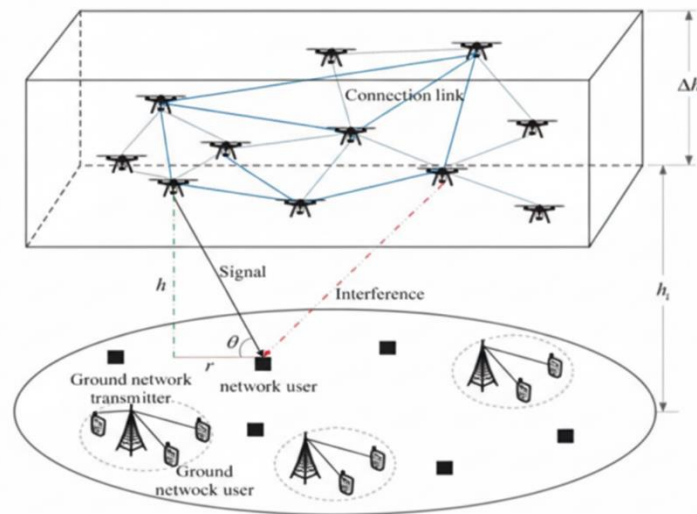


Figure 1-6 : Flying Ad Hoc Network

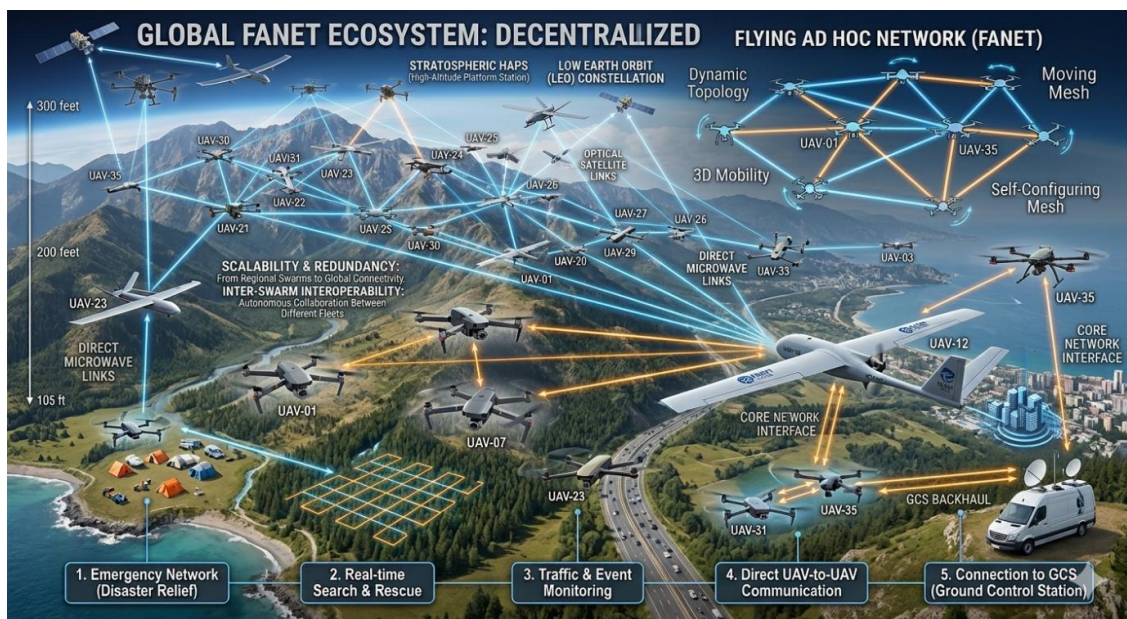


Figure 1-7 : Flying Ad Hoc Network

Table 1.1 WIFI 802.11n standards

IEEE standard	Frequency/ medium	Speed	Transmission range	Transmission Range
802.11	2.4 GHz RF	1-2 Mbps	20 feet	
802.11a	5 GHz	Up to 54 Mbps	25—75 feet; range can be affected by building materials	
802.11b	2.4 GHz	Up to 11 Mbps	Up to 150 feet; range can be affected by building materials	
802.11g	2.4 GHz	Up to 54 Mbps	Up to 150 feet; range can be affected by building	
802.11n	2.4 GHz/5 GHz	Up to 600 Mbps	175+feet; range can be affected by building materials	
802.11ac	5 GHz	Up to 600 Mbps	175+feet; range can be affected by building materials	

A. Characteristics of FANETs

As shown in Figure 5, the main characteristics of FANETs are node density, node mobility, changing network topology, communication range, radio propagation model, localization, power consumption, frequency band, cost efficiency, versatility, agility, and network lifetime [14-17], as detailed below.

A.1. Node Density

The average number of UAVs per unit volume is the node density. The node density of UAVs in FANETs is less than in other ad hoc networks such as MANET and VANET. The node density can be varied according to the objective of the UAVs mission.

A.2 Node Mobility

Node mobility is one of the main features of FANETs; it is very high compared to VANET and MANET. UAV speed varies from 30 to 460 km/h depending on the type of UAV. This can cause issues, including disruptions, link failure, and more [18].

A.3 Changing network topology

The network topology in FANETs undergoes frequent changes due to the rapid movements of UAVs. Possible FANET topologies under conditions of frequent topological fluctuation include star topology, in which all UAVs directly communicate with the ground control station (GCS), and mesh topology, in which dynamic routing is necessary. Both the star and mesh network topologies have advantages and disadvantages; for example, with star network topologies, the dedicated link between each UAV and GCS fluctuates due to the high speed of UAVs, which can affect data exchange [15,19-20].

A.4. Radio propagation model

A crucial element when designing and simulating any communications system is the radio propagation model employed in the network. The simplest and most popular propagation model used in simulation tests is the Friis free space model. This model only uses the distance and frequency of the signal, which has corresponding limitations [21]. A UAV-to-Ground (U2G) communication channel is a widely used channel model in the literature. The different types of propagation models can be categorized [22] as follows:

- Theoretical models: These models provide a detailed propagation model of U2G or U2U channels for UAV network scenarios.
- Empirical models: These models are obtained from a series of measurements made in various rural or urban scenarios.
- Semi-empirical models: These models are initiated as theoretical models and then varied according to a set of measures to match reality.
- Well-known models: These models attempt to verify the sufficiency of already-known propagation models in UAV network scenarios.

A.5. Localization

FANETs take advantage of low-latency global positioning (GPS) to locate UAVs to compensate for their high speed and mobility and the resulting network topology changes. Localization in FANETs can be based on network positioning, height, assisted GPS (AGPS), or differential GPS (DGPS) [15,23].

A.6. Power consumption and network lifetime

Energy constraints represent a critical issue in ad hoc networks. Power consumption in FANETs depends on the size of the UAVs, the distances involved, the communication hardware of the FANET and the link, and other hindrances. Sensor and actuator nodes play a vital role in the power consumption of FANETs; lowering the requirements of power-sensitive devices in FANETs can directly improve network lifetime and reduce network breakdowns [15,23].

A.7. Frequency band

ISM bands such as 900 MHz and 2.4 GHz are widely used in UAV communication systems; however, their shared nature can lead to congestion and interference. The 5 GHz band, supported by IEEE 802.11a, has shown favorable performance for UAV-to-Ground links. Meanwhile, the 5.9 GHz band with IEEE 802.11p offers better interference avoidance, making it suitable for UAV communications in environments with dense radio traffic [15].

B. Applications of FANETs

Flying Ad Hoc Networks (FANETs) are being increasingly used in fields offering advantage and facing specific challenges in each application [13].

B.1 Military Operations

FANETs enhance communication in crisis management, border surveillance, search and destroy missions. They provide a communication network when traditional infrastructure is compromised, FANETs can be quickly Self-configured to connect with rescue teams during disaster management.

B.2 Environmental Monitoring

FANETs play a role in applications such as wildfire management and wind estimation by enabling real-time data collection and sharing due to their mobility and flexibility make it easier to cover areas efficiently and improving environmental monitoring processes and decision-making. Disaster management plays a role, in dealing with crisis situations. In scenarios, search and rescue operations and the use of damaged network systems (FANETs) are vital more over they not enable real time data sharing and situational awareness. Also assist in making critical decisions during emergencies [11].

FANETs also have applications in traffic monitoring, Unmanned aerial vehicles (UAVs) within FANETs contribute by providing observations of crop conditions and aiding in traffic analysis for flow management. However, it's important to note that effectively operating and maintaining UAVs over areas is essential for these applications. When it comes to search and rescue

operations FANETs prove their worth by facilitating target detection efforts and aiding in rescue missions. Ensuring a connection for real time communication is the challenge that needs to be addressed. Additionally, FANETs find applications, in healthcare, traffic control and public safety fields. They have the ability to quickly adapt to changing needs within these domains. However, there are challenges associated with ensuring network stability and coverage across environments.

B.3. Post-Disaster Communication

A FANET can play a significant role in post-disaster scenarios where ground users (GUs) lose communication with the outside world due to the destruction of conventional communication infrastructure. In such situations, UAVs cooperate to rapidly construct an aerial ad hoc network capable of restoring communication services. The primary objective of the FANET is to provide wireless coverage to the maximum number of affected users and to transmit critical disaster information to the rescue centre as quickly as possible.

C. Research issues and challenges

Considering few similarities between FANET, MANET and VANET; FANET inherits some of the issues and challenges from MANET and VANET. However, due to additional characteristics, such as, very high speed of UAVs in FANET, the routing protocols require major review and investigation. So, the data routing between UAVs undergoes a serious challenge. The routing communications protocol must be able to update routing table or cache dynamically according to modification in the topology. Previous protocol does not provide a reliable communication. So, there is a need of new communication protocol to provide a flexible and reliable communication. There are also various issues related to transmission like, security overheads, loss of data packets, and use of energy. To some extent, a FANET is not quite the same as customary MANETs and VANETs; but the main idea is the same: having dynamic nodes in an ad hoc manner. Consequently, in a FANET, a few difficulties are substantial as in a VANET while confronting with extra difficulties. Many inquiries have been performed to build the productivity of system with flying nodes, there are as yet numerous unsolved issues, which ought to be investigated.

C.1. Routing

In a FANET, routing of data between UAVs is a serious challenge, which is not same as the MANETs with low portability mobility conditions. As indicated by topology changes, routing tables must be updated dynamically. Most of existing routing algorithms' metric calculations are neglected in FANET to give a dependable correspondence between UAVs. Reliability of routes is also a real challenge due to the very high mobility in FANET. Authors in [24] have suggested reliable routing protocol, however suitability of such routing techniques in FANET is required. Along these lines, there is a need of new research direction to calculate routing metrics and develop efficient routing algorithms and network models for developing an adaptable and responsive ad hoc model.

C.2. Path Planning

In a large-scale mission territory and multi-UAV operations, participation and coordination between UAVs are one of the most important factors to increase the efficiency of a FANET. In such cases, each UAV needs to change its pathway, and new ones ought to be re-computed progressively. Accordingly, new methods/algorithms are required to arrange the FANET nodes for organizing the clusters of UAVs.

C.3. Quality of Service (QoS)

A FANET can be utilized for transporting various types of goods to the customer residences such as the delivery drones from Amazons. They incorporate GPS maps, streaming video/voice, images, straightforward instant messages etc. Provision of certain qualities to service [25] parameter such as delay, bandwidth and packet loss are essential for FANET applications. Characterizing an exhaustive system for QoS-empowered middleware is a serious challenge that ought to be overcome because of the exceedingly mobile and dynamic structure of FANET. Misbehaviors [26] of nodes in FANET could also affect the quality of services in their operations. Coordination between UAVs and manned aircrafts: It is an inescapable fact that, later on with increased number of UAVs, the flights of UAVs must be coordinated with the manned aircraft. Similarly, in a military aspect, destruction of enemy aircraft may be achieved by utilizing a FANET or these UAVs can be utilized as electronic jammers. Furthermore, these UAVs can be used as surveillance in foe zones.

1.4. Conclusion

This chapter provided an overview of wireless ad hoc networks and highlighted their role in enabling communication in environments lacking reliable infrastructure. The distinction between infrastructure-based and infrastructure-less wireless networks was introduced, emphasizing the self-organizing and decentralized nature of ad hoc networks. Furthermore, the main types of ad hoc networks—MANETs, VANETs, SANETs, and FANETs—were discussed in terms of their characteristics, applications, and challenges. Among these, Flying Ad Hoc Networks (FANETs) were identified as a particularly complex and promising research area due to their high mobility, three-dimensional movement, and strict energy constraints.

The discussion presented in this chapter establishes the necessary background for understanding FANETs and their unique design challenges. This foundation supports the motivation for the subsequent chapters, which focus on advanced communication strategies and optimization techniques for UAV-based networks.

Chapter 2: UAV-Assisted Wireless Communication Networks

2.1 Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have emerged as one of the most transformative technologies in next-generation wireless communication systems. Driven by their inherent mobility, flexible deployment, and ability to establish reliable line-of-sight (LoS) communication links, UAVs are increasingly integrated into wireless network architectures to augment coverage, capacity, and resilience. This chapter provides a comprehensive review of the key aspects of UAV-assisted wireless communication networks, structured around four fundamental themes: the role of UAVs in wireless communications, UAV deployment and coverage models, air-to-ground communication channels, and energy constraints in UAV networks. Together, these themes form the theoretical and technical foundation upon which modern UAV-assisted network systems are designed, optimized, and evaluated [27][28].

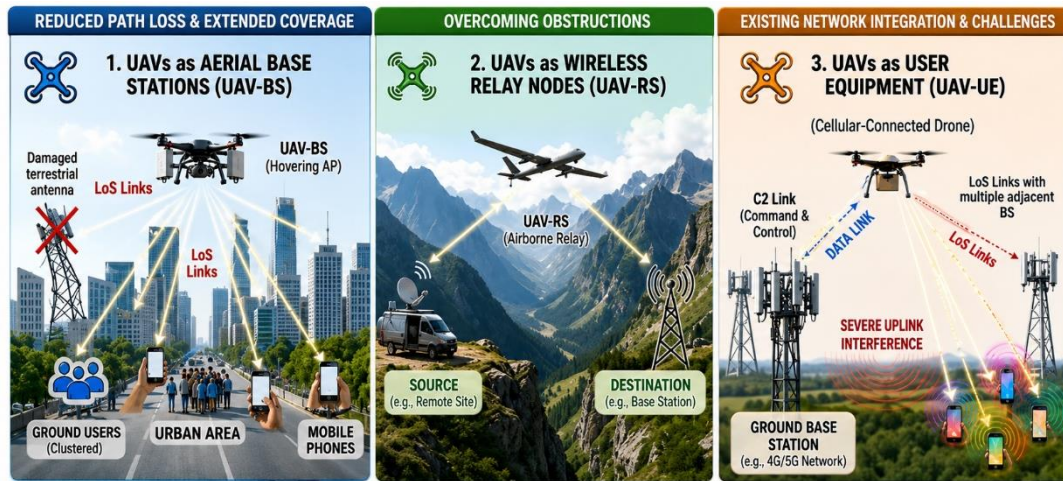


Figure 2-1 : UAV assisted wireless communication

2.2 Role of UAVs in Wireless Communications

UAVs occupy a unique and multifaceted role in wireless communication networks, serving simultaneously as infrastructure providers, mobile users, and signal relays. Their elevated position above ground level confers a fundamental propagation advantage: the probability of establishing LoS links with ground users is substantially higher than for terrestrial base stations, particularly in urban and semi-urban environments where buildings obstruct ground-level signals. This LoS advantage directly translates into reduced path loss, improved signal quality, and extended coverage range [27][29]. The literature broadly classifies UAV roles into three fundamental categories, each corresponding to a distinct network architecture and design challenge.

2.2.1 UAVs as Aerial Base Stations (UAV-BS)

In this configuration, the UAV replaces or supplements a ground base station (GBS), hovering at an optimized altitude to provide downlink and uplink services to a cluster of ground users. UAV-BSs are particularly valuable in post-disaster scenarios where terrestrial infrastructure has been damaged or destroyed, as well as for temporary coverage at large-scale public events, remote area connectivity, and military communications [27]. The UAV-BS operates as an access point, managing radio resource allocation, interference coordination, and user scheduling in the same manner as a conventional GBS, but with the added degrees of freedom of altitude and 2D horizontal positioning. Hayajneh et al. [30] demonstrated that a properly deployed UAV-BS can serve up to 40% more users than an equivalently powered GBS in dense urban scenarios, by exploiting altitude to overcome LoS blockage.

2.2.2 UAVs as Wireless Relay Nodes (UAV-RS)

When a direct link between a source and a destination is infeasible due to excessive path loss, physical obstruction, or distance, a UAV can be deployed as an airborne relay station (UAV-RS). The UAV relay receives signals from the source, amplifies or decodes them, and retransmits them toward the destination, effectively extending the communication range. UAV relays are particularly effective in mountainous terrain, maritime environments, and disaster zones where ground-level relay deployment is impractical [28]. Zeng et al. [31] showed that a properly positioned UAV relay can increase the end-to-end channel capacity by more than 3 dB compared to a static ground relay, by simultaneously optimizing relay position and power allocation. The decode-and-forward (DF) protocol is most commonly adopted for UAV relays due to its noise suppression capabilities, while amplify-and-forward (AF) is preferred in latency-sensitive applications [29].

2.2.3 UAVs as User Equipment (UAV-UE)

In cellular-connected UAV deployments, the UAV itself functions as a user equipment (UE) within an existing 4G/5G network, receiving control and communication services from ground base stations. This is common in commercial drone delivery, surveillance, and inspection applications where real-time command and control (C2) links are required. UAV-UEs introduce unique interference challenges: because UAVs fly at elevated altitudes, they establish LoS links with a large number of surrounding base stations simultaneously, creating severe uplink interference that degrades service quality for ground users [32]. The 3rd Generation Partnership Project (3GPP) has addressed this in Release 15 and Release 17, introducing enhanced interference mitigation techniques and UAV identification mechanisms specifically for cellular-connected UAV-UEs [32].

Table 2.1 Comparative Summary of UAV Roles in Wireless Communications

UAV Role	Network Function	Key Advantage	Main Challenge	Typical Use Case
Aerial Base Station	Access Point / BS	On-demand coverage	3D placement optimization	Disaster recovery
Relay Node	Signal forwarding	Range extension	Trajectory & power control	Remote connectivity
User Equipment	Network subscriber	Real-time C2 link	Uplink interference	Drone delivery / surveillance

2.3 UAV Deployment and Coverage Models

The deployment of UAVs in wireless networks is a three-dimensional (3D) optimization problem: the horizontal position (x, y) and altitude h of each UAV must be jointly determined to maximize coverage, minimize interference, or satisfy user-specific quality-of-service (QoS) requirements. UAV deployment models are broadly classified into two paradigms: static deployment, where UAVs hover at fixed positions, and dynamic deployment, which involves trajectory-based movement. This section focuses on static deployment and coverage modeling, which constitutes the foundational framework for UAV network design [27][33].

2.3.1 Air-to-Ground LoS Probability and Optimal Altitude

The coverage performance of a deployed UAV is fundamentally determined by the air-to-ground (A2G) channel characteristics, particularly the probability of establishing a LoS link with a ground user. The ITU-R (International Telecommunication Union – Radiocommunication Sector) and the widely-cited model of Al-Hourani et al. [34] express the LoS probability as a function of the elevation angle θ between the UAV and the ground user:

$$P_{LoS}(\theta) = 1 / [1 + a \cdot \exp(-b(\theta - a))] \quad (2.1)$$

where $\theta = \arctan(h / r)$, h is the UAV altitude, r is the horizontal ground distance, and a, b are environment-dependent constants (e.g., for urban environments: $a = 9.61$, $b = 0.16$ [34]). The aggregate path loss incorporating both LoS and NLoS components is then:

$$L(h, r) = P_{LoS} \cdot L_{LoS} + (1 - P_{LoS}) \cdot L_{NLoS} \quad (2.2)$$

where L_{LoS} and L_{NLoS} are the free-space path loss values for LoS and NLoS propagation, respectively, each incorporating an additional attenuation factor specific to the environment type. This model reveals a fundamental tradeoff: increasing altitude improves LoS probability but also increases free-space propagation distance, resulting in an optimal altitude h^* that minimizes total path loss for a given horizontal distance r . Al-Hourani et al. [34] derived a closed-form expression for h^* as:

$$h^* = r \cdot \tan\left[\frac{1}{b} \cdot \ln\left(\frac{b}{a} \cdot \left(\sqrt{1 + \frac{ab}{4\pi^2 r^2}} - 1\right)\right)\right]^{1/2} \quad (2.3)$$

This optimal altitude directly determines the maximum coverage radius R^* , which is the key design parameter for single-UAV deployment. Alzenad et al. [35] extended this result to a 3D placement framework, decoupling the problem into an altitude sub-problem followed by a 2D horizontal positioning sub-problem, resulting in a computationally tractable algorithm applicable in real-time network planning scenarios.

2.3.2 Single UAV Deployment

In single-UAV deployment, the network relies on one aerial node to provide coverage over a target area. The core design problem is formally stated as a constrained optimization:

$$\begin{aligned} & \text{maximize} \quad |U(x, y, h)| \\ & \text{subject to:} \quad h_{\min} \leq h \leq h_{\max}, \quad L(h, r_k) \leq L_{th} \quad \forall k \in U \end{aligned} \quad (2.4)$$

where $|U(x, y, h)|$ is the number of ground users covered, L_{th} is the maximum tolerable path loss threshold, and r_k is the distance to the k -th user. A critical limitation of single-UAV networks is their vulnerability to failure: a single link interruption renders the entire network unavailable, motivating the adoption of redundant multi-UAV architectures for mission-critical applications [28][36].

2.3.3 Multiple UAV Deployment

Multi-UAV deployment overcomes the limitations of single-UAV networks by distributing coverage across multiple aerial nodes. When configured as a mesh network, multiple UAVs can self-reorganize upon link disruption, rerouting information through alternative paths to maintain service continuity [28]. Mozaffari et al. [36] applied circle packing theory to the multi-UAV deployment problem, modeling each UAV coverage footprint as a circular disc and minimizing the number of UAVs N required to cover a target area of radius R_c :

$$N^* = \lceil R_c^2 / r^*(h^*)^2 \rceil \quad (2.5)$$

where $r^*(h^*)$ is the optimal coverage radius at the optimal altitude. More recent traffic-aware approaches, such as the EMTAD algorithm of Zhang et al. [37], jointly minimize UAV count while ensuring per-user traffic demands are satisfied, formulating the problem as a Mixed-Integer Linear Programming (MILP) problem solved via iterative k-means clustering and linear relaxation. The key challenge in multi-UAV deployment lies in inter-UAV interference management and backhaul link design, particularly when UAVs must coordinate to serve overlapping coverage regions [33].

Table 2.2 Single vs. Multiple UAV Static Deployment

Criterion	Single UAV	Multiple UAVs
Coverage Area	Limited (one footprint)	Extended (tessellated)
Network Resilience	None — single failure = outage	High — mesh rerouting
Optimization	3D placement (tractable)	MILP / k-means (NP-hard)
Interference	No inter-UAV interference	Requires coordination
Use Case	Hotspot, small events	Wide-area disasters, IoT

2.4 Air-to-Ground Communication

The air-to-ground (A2G) communication channel is the cornerstone of UAV-assisted wireless network design. Unlike conventional terrestrial channels, the A2G channel exhibits distinct propagation characteristics that are governed by the UAV altitude, velocity, elevation angle, and surrounding environment type. A thorough understanding of A2G channel behavior is essential for accurate system-level modeling, coverage prediction, and interference analysis in UAV-assisted networks [38][39].

2.4.1 Large-Scale Fading: Path Loss Models

Large-scale fading in A2G channels is primarily characterized by path loss, which quantifies the average signal attenuation over distance. Three widely adopted path loss models are used in the UAV communication literature:

a. Free-Space Path Loss (FSPL) Model: Applicable when a clear LoS exists between the UAV and ground user, the FSPL model gives:

$$L_{FS} = 20 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f) + 20 \cdot \log_{10}(4\pi/c) \quad (2.6)$$

where d is the 3D Euclidean distance, f is the carrier frequency, and c is the speed of light. As altitude increases and propagation approaches free-space conditions, the path loss exponent (PLE) approaches 2, consistent with LoS propagation [38].

b. Log-Distance Path Loss Model: For environments where terrain, buildings, or foliage partially obstruct the signal, the close-in (CI) log-distance model is preferred:

$$L(d) = L_{FS}(d_0) + 10 \cdot n \cdot \log_{10}(d/d_0) + X_\sigma \quad (2.7)$$

where d_0 is a reference distance (typically 1 m), n is the PLE, and X_σ is a zero-mean Gaussian random variable representing shadow fading with standard deviation σ (in dB). Experimental measurements across multiple UAV altitudes (1.5–120 m) have shown that PLE decreases from 3.7 to 2.0 as altitude increases, confirming the progressive transition toward free-space propagation at higher altitudes [39].

c. ITU-R Statistical LoS/NLoS Model: This model, also known as the Al-Hourani model, it is the most widely adopted in UAV network design. It computes the mean path loss as a probabilistic combination of LoS and NLoS components, as described in Section 2.3.1. The additional attenuation factors η_{LoS} and η_{NLoS} are environment-specific constants that capture scattering and diffraction losses beyond free-space propagation [34].

2.4.2 Small-Scale Fading

Small-scale fading arises from multipath propagation, where multiple reflected, scattered, and diffracted signal components combine at the receiver with random phases, causing rapid amplitude fluctuations over short distances. In A2G channels, the dominant fading distribution is the **Rician distribution**, characterized by the K-factor — the ratio of the power in the dominant LoS component to the total scattered power. In strong LoS conditions (high altitude, low obstruction), the K-factor is large ($K \gg 1$) and the channel behaves approximately as an AWGN channel. In NLoS conditions, $K \approx 0$ and the fading follows a Rayleigh distribution [38]. The Nakagami-m distribution is also used as a generalized fading model that encompasses both Rayleigh ($m = 1$) and near-AWGN ($m \rightarrow \infty$) cases [30].

2.4.3 Doppler Effects and Channel Non-Stationarity

Unlike static base stations, UAVs are mobile platforms whose velocity introduces **Doppler shifts** into the received signal. For a UAV moving at velocity v at an angle φ relative to the signal direction, the Doppler shift is:

$$f_D = (v \cdot \cos \varphi) / \lambda \quad (2.8)$$

where λ is the carrier wavelength. At typical UAV speeds (5–30 m/s) and 5G carrier frequencies (3.5 GHz), Doppler shifts can reach several hundred Hz, necessitating Doppler compensation in the receiver design [38][14]. Furthermore, because UAVs traverse diverse propagation environments during flight, the A2G channel is **non-stationary**: its statistical properties (mean path loss, K-factor, shadow fading standard deviation) vary along the flight trajectory, requiring time-varying channel models rather than wide-sense stationary (WSS) assumptions [38].

Table 2.3 : A2G Channel Models Summary

Model	Type	Key Parameter	Best Suited For
Free-Space (FSPL)	Deterministic	Distance, frequency	Clear LoS, high altitude
Log-Distance CI	Statistical	PLE n , shadow std σ	Mixed LoS/NLoS environments
ITU-R LoS/NLoS	Probabilistic	a , b constants, η_{LoS} , η_{NLoS}	UAV-BS deployment planning
Rician Fading	Small-scale	K-factor	LoS-dominant A2G links
Rayleigh Fading	Small-scale	σ^2 scatter power	NLoS A2G links

2.5 Energy Constraints in UAV Networks

Energy limitation is arguably the most critical practical constraint in UAV-assisted wireless networks. Commercial UAVs are powered by lithium-polymer (LiPo) batteries with a typical flight endurance of 15–30 minutes, severely restricting their operational duration and mission range [41][16]. Unlike terrestrial base stations connected to the power grid, UAVs must balance two competing energy demands: **propulsion energy** required to keep the UAV airborne and mobile, and **communication energy** consumed by signal transmission, reception, and baseband processing. Understanding and optimizing these energy components is fundamental to extending UAV network lifetime and enabling practical deployments [17].

2.5.1 Propulsion Energy Model

Propulsion energy consumption constitutes the dominant energy expenditure for UAVs, typically accounting for 80–85% of total power consumption [41]. For rotary-wing UAVs (the most common platform for wireless network applications), the propulsion power model by Zeng et al. [31] is widely adopted:

$$P_{\text{prop}}(v) = P_0(1 + 3v^2/U_{\text{tip}}^2) + P_i(\sqrt{(1 + v^4/4v_h^4) - v^2/2v_h^2})^{1/2} + (1/2) \cdot d_o \cdot \rho \cdot s \cdot A \cdot v \quad (2.9)$$

where P_0 and P_i are blade profile and induced power constants in hovering, U_{tip} is the rotor tip speed, v_h is the mean rotor induced velocity in hover, d_o , ρ , s , A are fuselage drag, air density, rotor solidity, and rotor disc area, respectively. This model reveals that propulsion power exhibits a non-monotonic relationship with flight speed: it is minimized at an intermediate optimal speed v^* and increases sharply

for both hovering ($v = 0$) and high-speed flight, providing a key design insight for energy-optimal trajectory planning [31].

2.5.2 Communication Energy Model

The communication energy of a UAV includes transmit power, circuit power, and signal processing overhead. For a UAV-BS transmitting to K ground users over a time horizon T , the total communication energy is:

$$E_{comm} = \int_0^T [P_{tx}(t) + P_c] dt \quad (2.10)$$

where $P_{tx}(t)$ is the instantaneous transmit power and P_c is the constant circuit power. The transmit power is constrained by a maximum budget P_{max} and must satisfy the Shannon capacity requirement for each user. Niu et al. [42] formulated the energy efficiency (EE) maximization problem for a UAV-BS as:

$$\max EE = \sum_k C_k(h, P_{tx}) / E_{total}(h, P_{tx}, T) \quad (2.11)$$

with the subject to per-user statistical delay QoS constraints, jointly optimizing UAV altitude and bandwidth allocation can yield EE improvements of up to 35% compared to altitude-only or power-only optimization [42].

2.5.3 Energy-Aware Trajectory and Resource Optimization

The joint optimization of UAV trajectory and communication resource allocation to minimize total energy consumption while maintaining service quality is a central research problem in UAV-assisted networks. Zeng et al. [31] demonstrated that for a UAV relay system, the optimal trajectory that minimizes total energy for a given mission is not necessarily the shortest path, but one that balances propulsion energy savings from speed optimization against communication throughput requirements.

Recent deep reinforcement learning (DRL) approaches have emerged as powerful tools for real-time energy-aware UAV control. Tang et al. [19] proposed a DRL framework that jointly optimizes UAV trajectory and IoT device transmission resources, achieving 28% propulsion energy reduction compared to greedy baseline approaches. Multi-agent DRL methods (MAD-DDQN) have been applied to multi-UAV scenarios, simultaneously optimizing 3D trajectories, transmission power, and user associations to maximize system-level energy efficiency under interference constraints [17].

2.5.4 Energy Replenishment Strategies

To overcome the fundamental battery limitation, several energy replenishment strategies have been proposed. UAV replacement involves maintaining a fleet of backup UAVs that are swapped into service as the active UAVs approach battery depletion — an approach modeled as a bin-packing problem by Tipantuna et al. [16], who showed that the problem is NP-hard in general and proposed efficient heuristic algorithms. Wireless power transfer (WPT) enables ground energy sources to charge UAVs wirelessly during hovering, while solar-powered UAVs integrate photovoltaic cells to extend flight endurance

indefinitely in favorable weather conditions [16][20]. Landing spot optimization — where UAVs periodically land on designated platforms to eliminate hovering energy — has also emerged as a novel strategy for prolonged network operation [17].

Table 2.4 : UAV Energy Optimization Strategies

Strategy	Optimized Variable	Method	Gain
Trajectory Optimization	Flight path & speed	Convex opt., SCA, DRL	Up to 40% propulsion saving
Joint Traj. + Power	Trajectory + P_tx	Multi-variable DRL	28–35% EE improvement
UAV Replacement	Battery swap schedule	Bin-packing heuristics	Unlimited service time
Solar Integration	Charging profile	MPC, hybrid control	Extended endurance
Landing Spot Opt.	Hover-vs-land decision	RL, optimization	Hover energy eliminated

2.6 Conclusion

This chapter has demonstrated that Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology in next-generation wireless communication systems. By leveraging their inherent mobility, flexible deployment, and ability to establish reliable line-of-sight (LoS) communication links, UAVs are increasingly integrated into wireless network architectures as aerial base stations, relays, and user equipment to enhance coverage, capacity, and resilience.

Chapter 3: Deep Reinforcement Learning

3.1 Introduction

Reinforcement Learning (RL) [43, 46] is a branch of machine learning in which an agent learns from interacting with an environment. Before an agent or robot (software or hardware) can select an action, it must have a good representation of its environment [44]. Thus, perception is one of the key problems that must be solved before the agent can decide to select an optimal action to take.

Representation of the environment might be given or might be acquired. In reinforcement learning tasks, usually a human expert provides features of the environment based on his knowledge of the task. However, for some real-world applications this work should be done automatically, since automatic feature extraction will provide much more accurate results. There are several solutions to deal with this challenge, such as Mont Carlo Tree search [47], Hierarchical Reinforcement Learning [45] and function approximation [46]. Hence, we are interested in the ways that can be used to represent the state of the environment. Deep learning attempts to model high-level abstractions in data using deep networks of supervised and/or unsupervised learning algorithms, in order to learn from multiple levels of abstractions. It learns hierarchical representations in deep architectures for classification.

In recent years, deep learning has gained huge attraction not only in academic communities (such as pattern recognition, speech recognition, computer vision and natural language processing), but it has also been used successfully in industry products by tech giants like Google (Googles translator and Image search engine), Apple (Apple's Siri), Microsoft (Bing voice search) and other companies such as Facebook and IBM. Recent research has also shown that deep learning techniques can be used to learn useful representations for reinforcement learning problems [28, 29, 26, 8, 30]. Combining the RL and deep learning techniques enables RL agent to have a good perception of its environment.

The aim of this chapter is to present and review the main concepts and recent advances in deep reinforcement learning (DRL) based on UAV placement, which combines reinforcement learning algorithms with deep learning techniques. This chapter first introduces the fundamentals of reinforcement learning and deep learning, highlighting how deep neural networks can be integrated with RL methods to learn efficient control policies in complex and high-dimensional environments.

3.2 Reinforcement Learning

Reinforcement Learning [43, 46] is a branch of machine learning in which an agent learns from interacting with an environment. An RL framework allows an agent to learn from trial and error. The RL agent receives a reward by acting in the environment and its goal is learning to select the actions that maximize the expected cumulative reward over time. In other words, the agent, by observing the results of those actions that it is taking in the environment, tries to learn an optimal sequence of actions to execute in order to reach its goal.



Figure 3-1 : Reinforcement Learning

A reinforcement learning agent can be modelled as a Markov decision process (MDP). If the states and action spaces are finite, then the problem is called a finite MDP.

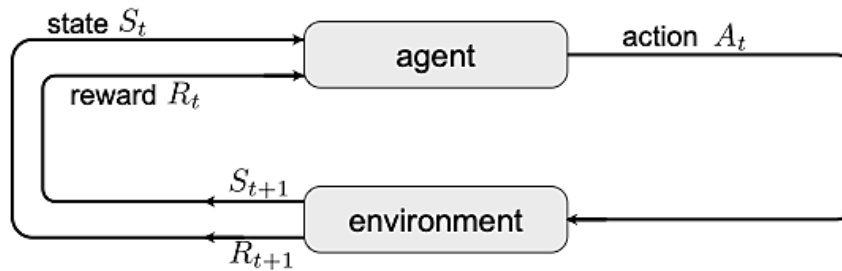


Figure 3-2 : Markov decision process

At each time step t , the agent starts from a state $S_t \in S$, performs an action $A_t \in A$, which, through interaction with the environment, leads to a reward $R_t + I \in R$ and moves the agent to a new state S_{t+1} . This agent-environment interaction is schematically shown in Figure.3.2.

Finite MDPs are very important for RL problems and much of literatures have assumed the environment is a finite MDP in their works. The way an RL agent acts in the finite MDP framework is as follows: The learning agent interacts with the environment by executing actions

and receiving observations and rewards. At each time step t , which ranges over a set of discrete time intervals, the agent select an action A from a set of legal actions $A = 1, 2, \dots, k$ at state $S_t \in S$, where S is the set of possible states. Action selection is based on a policy. The policy is a description of the agent behaviour which tells the agent which actions should be selected for each possible state. As a result of each action, the agent receives a scalar reward $R_t \in R$, and observes next state $S_{t+1} \in S$ at one step time later.

The probability of each possible next state S_{t+1} comes from a transition distribution which is $P(S_{t+1}|S_t, A_t)$; where $S_{t+1}, S_t \in S, A_t \in A(S_t)$. Similarity, the probability of each possible reward R_t comes from a reward distribution $P(R_t|S_t, A_t)$; where $S_t \in S, A_t \in A(S_t)$.

Hence, the expected scaler reward received, R_t , by executing action A in current state S is calculated based on $E_P(R_t|S_t, A_t)(R_t|S_t = S, A_t = A)$. The aim of the learning agent is to learn an optimal policy π , which defines the probability of selecting action A in state S , so that with following the policy the sum of the discounted rewards over time is maximized. The expected discounted return R at time t is defined as follows:

$$R_t = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots] = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right], \quad (3.1)$$

Where $E[\cdot]$ is the expectation with respect to the reward distribution, and $0 < \gamma < 1$ is called the discount factor. With regard to the transition probabilities and the expected discounted immediate rewards, which are the essential elements for specifying dynamics of a finite MDP, action-value function $Q\pi(s,a)$ is defined as follows:

$$Q^\pi(s, a) = E_\pi[R_t | s_t = s, a_t = a] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a\right], \quad (3.2)$$

The action-value function $A\pi(s,a)$ for an agent is the expected return achievable by starting from state $s, s \in S$, and performing action $a, a \in A$ and then following policy π , where π is a mapping from states to actions or distributions over actions. With unfolding the equation 2 it is clear that it satisfies a recursive property, so that the following iterative update can be used for the estimation of action-value function:

$$\begin{aligned} Q_{i+1}^\pi(s, a) &= E_\pi\left[r_t + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right] \\ &= E_\pi[r_t + \gamma Q_i^\pi(s_{t+1} = s', a_{t+1} = a') | s_t = s, a_t = a], \end{aligned} \quad (3.3)$$

For all $s, s' \in S$ and $a, a' \in A$, in Eq. 3.3, both states a relationship between the value of an action in a state and the values of its next actions which can be performed. It also describes the way of estimating the value based on its subsequent ones.

Reinforcement learning agent attempts to find a policy which achieves the greatest future reward in the course of its execution. Hence, it must learn an optimal policy π^* , a policy which is resulted to an expected value greater than or equal of following other policies for all states, and as a result, an optimal state-value function $Q^*(s, a)$. In particular, an iterative update for estimating the optimal state-value function is defined as follows:

$$Q_{i+1}(s, a) = E_{\pi}[r_t + \gamma \max_{a'} Q_i(s', a') | s, a], \quad (3.4)$$

where it is implicit that $s, s' \in S$ and $a, a' \in A$. The iteration converges to the optimal action-value function, Q^* as $i \rightarrow \infty$ and called value iteration algorithm [46].

In many real problems, the number of states and actions are very large, making the use of classical solution (state-action table to store the values of state-action pairs) impractical. One way to deal with this is to use a function approximator as an estimator of action-value function. The approximate function value is parameterized as $Q(s, a; \theta)$ with parameter vector θ . Usually gradient-descent methods are used to learn parameters by trying to minimize the following loss function of mean-squared error in Q-values:

$$L(\theta) = E_{\pi}[(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2], \quad (3.5)$$

where $r + \gamma \max_{a'} Q(s, a; \theta)$ is the target value. Differentiation of the loss function with respect to its parameters θ lead to the following gradient:

$$\frac{\partial L(\theta)}{\partial \theta} = E[(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta)) \frac{\partial Q(s, a; \theta)}{\partial \theta}], \quad (3.6)$$

Typically, the gradient function given in the above function is optimized by stochastic gradient descent method. The approximate function can be a linear function or a non-linear function (for example a neural network) of the parameters θ . Until recently, the majority of works in reinforcement learning have utilized linear function approximators because of the convergence guarantees they provide. More recently, by alleviating the convergence problems, not only typical neural networks, for example, multilayer perceptrons (MLPs) have commonly been used as function approximators for large reinforcement learning tasks. However, deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are also widely used [34, 32, 29].

3.3 Deep Learning

Obtaining a good performance of a machine learning technique is highly dependent on having good representation of input data. Hence, the pre-processing of the data (i.e., feature learning) is a critical step in the process of creating the machines which are learned through the observation of the data. The feature engineering process is a way to take advantage of the knowledge of the domain experts in order to extract hand-crafted features and to reduce dimension of features of the input data. Effectiveness of the shallow learning models such as support vector machines (SVMs) and logistic regression are dependent on feature learning. This process is important but very time consuming and difficult to do. It would be better to have algorithms that facilitate the problem. Deep learning techniques are one the best solutions to deal with high dimension data and extract discriminative information from the data. Deep learning algorithms have the capability of automating feature extraction (the extraction of representations) from the data. The representations learnt through the data are fed directly into deep networks without human knowledge (i.e., automated feature extraction). This key aspect of deep learning architectures is led to progress towards those algorithms that are the goal of Artificial Intelligence (AI), understating the world around independent of expert knowledge and interference [48].

In summary, deep learning attempts to model high-level abstractions in data using deep networks of supervised and/or unsupervised learning algorithms, in order to learn from multiple levels of abstractions. It learns hierarchical representations in deep architectures for different tasks such as classification.

Deep learning models contain multiple layers of representations (figure 3.3). Indeed, it is a stack of building blocks such as autoencoders, Restricted Boltzmann Machines (RBMs) and convolutional layers. During training, the raw data is fed into a network consisting of multiple layers. The output of each layer which is nonlinear feature transformations, is used as inputs to the next layers of the deep network. The output representation of the final layer can be used for constructing classifiers or those applications which can have the better efficiency and performance with abstract representation of the data in a hierarchical manner as inputs. Each layer by applying a nonlinear transformation on its input try to learn and extract underlying explanatory factors. Consequently, this process is led to learn a hierarchy of abstract representations. For example, using deep leaning algorithm in image processing applications, the first layer is provided by the image pixels which can lead to learn the edges of different object in image. The second layer uses the representations provided by the first layer to lean

complex features such as object parts (combination of edges). The third layer composes object parts (more complex features) to figure out object models.

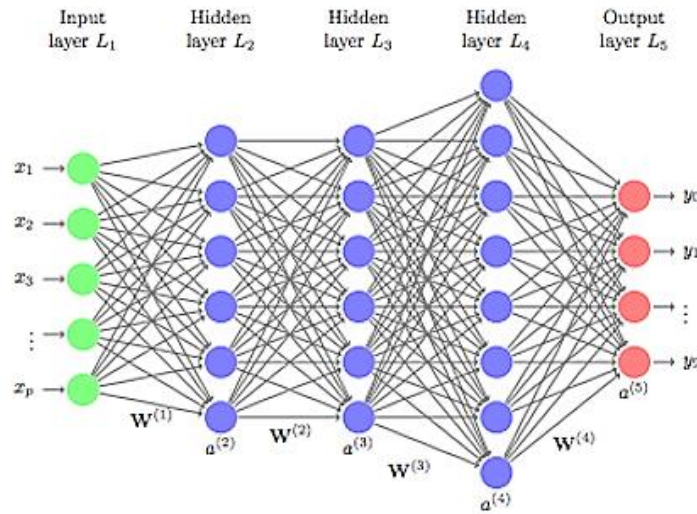


Figure 3-3 : Feedforward Deep Learning Model

The use of deep neural networks allows the deep learning methods to be very powerful tools in solving real problem. However, learning the parameters (deep nets with many hidden layers are led to have millions of parameters to learn) in a deep architecture this is a difficult optimization task which imposes very high computational complexity [25]. Fortunately, with emergent of advanced parallel processing technologies like GPU this problem has alleviated somewhat.

3.4 Deep reinforcement Learning (DRL) Models for UAV Placement optimization

The problem of optimal UAV placement and trajectory control is inherently sequential, partially observable, and constrained — properties that make it naturally amenable to DRL-based formulations. The following subsections review the principal DRL algorithms applied to UAV placement problems, proceeding from single-agent to multi-agent settings.

3.4.1 Deep Q-Network (DQN)

The Deep Q-Network (DQN) extends classical Q-Learning by replacing the Q-table (state/actions) with a deep neural network. DQN was originally proposed for Atari game control [Mnih et al., 2015] and was subsequently adapted to UAV systems. In UAV placement contexts, DQN has been applied to optimize the 3D trajectory of a UAV base station (UAV-BS) so as to maximize quality-of-experience (QoE) for mobile ground users. By using average channel power gain rather than individual user locations as the state input, the model reduces computational complexity while maintaining high placement accuracy.

The primary limitation of DQN is its restriction to discrete action spaces. The UAV's movement is therefore discretized into a finite set of directions and step sizes, which may introduce suboptimality in continuous 3D environments.

3.4.2 Deep Deterministic Policy Gradient (DDPG)

DDPG is a DRL algorithm that combines Deep Learning and Policy Gradient methods. It works well with continuous action spaces, making it suitable for tasks like UAV network optimization where actions, such as power allocation to users, are not discrete. DDPG uses a deterministic policy to choose actions and a Q-function (critic) to evaluate the expected return of those actions. This combination allows for efficient exploration and stable learning of optimal policies

The objective of the DDPG model is to optimize the power levels for all ground users in a UAV network to maximize the number of users served within the available power budget. This model operates in a continuous action space, making it suitable for fine-tuning power levels.

3.4.3 Proximal Policy Optimization (PPO)

Proximal policy optimization (PPO) is one of the most successful deep reinforcement learning methods, achieving state-of-the-art performance across a wide range of challenging tasks. However, its optimization behavior is still far from being fully understood. the PPO could neither strictly restrict the probability ratio as it attempts to do nor enforce a well-defined trust region constraint, which means that it may still suffer from the risk of performance instability. To address this issue, an enhanced PPO method, named Trust Region-based PPO with Rollback (TR-PPO-RB) was proposed. Two critical improvements are made in this method: 1) it adopts a new clipping function to support a rollback behavior to restrict the ratio between the new policy and the old one; 2) the triggering condition for clipping is replaced with a trust region-based one, which is theoretically justified according to the trust region theorem. It seems, by adhering more truly to the “proximal” property – restricting the policy within the trust region, the new algorithm improves the original PPO on both stability and sample efficiency.

3.4.4 Soft Actor-Critic (SAC)

Model-free deep reinforcement learning (RL) algorithms have been successfully applied to a range of challenging sequential decision making and control tasks. However, these methods typically suffer from two major challenges: high sample complexity (high sample complexity and brittleness to hyperparameters) and brittleness to hyperparameters (highly sensitive to the

choice of hyperparameters). Both of these challenges limit the applicability of such methods to real-world domains. In this section, we describe Soft Actor-Critic (SAC). In this framework, the actor aims to simultaneously maximize expected return and entropy. That is, to succeed at the task while acting as randomly as possible. SAC was extended to incorporate a number of modifications that accelerate training and improve stability with respect to the hyperparameters, including a constrained formulation that automatically tunes the temperature hyperparameter. With these improvements, SAC achieves state-of-the-art performance, outperforming prior on-policy and off-policy methods in sample-efficiency and asymptotic performance. Furthermore, in contrast to other off-policy algorithms, SAC is very stable, achieving similar performance across different random seeds.

3.4.5 Multi-Agent DRL: MADDPG and MAPPO

This method is used when a swarm of UAVs must coordinate.

- **Challenge:** If each UAV learns independently, the environment becomes "non-stationary" (the world changes because other drones are moving).
- **Solution:** Algorithms like MAPPO (Multi-Agent Proximal Policy Optimization) or MADDPG, can be used, where UAVs share some information (centralized training) but act independently (decentralized execution).
- **The Reward Function:** in this context the "Reward" is the signal that tells the DRL model if the placement was successful. A typical reward function R is a weighted sum of various metrics:

$$R = \omega_1 \cdot \text{coverage} + \omega_2 \cdot \text{Throughput} - \omega_3 \cdot \text{Energy Consumption} - \omega_4 \cdot \text{Penalty} \quad (3.7)$$

where Coverage is the number of users connected with a minimum SINR Signal-to-Interference-plus-Noise Ratio defined as follow:

$$\text{SINR} = \frac{\text{Signal Power}}{\text{Interference Power} + \text{Noise Power}} \quad (3.8)$$

Energy is the penalty for high-speed movement or long-distance travel.

Penalty is the negative reward if a UAV leaves the boundary or gets too close to another drone (collision avoidance).

3.4.6 Comparative Analysis

The following table presents a structured comparison of the DRL algorithms reviewed in this chapter with respect to their applicability to FANET-based UAV placement problems.

Table 3.1 : Comparative analysis of DRL algorithms for UAV placement in FANET.

<i>Feature</i>	DQN	DDPG	PPO	MADDPG	MAPPO
<i>Type</i>	Value-based	Actor-Critic	Policy-based	Multi-Agent AC	Multi-Agent PG
<i>Action Space</i>	Discrete	Continuous	Both	Continuous	Both
<i>Multi-Agent</i>	/	/	/	/	/
<i>3D UAV Movement</i>	Limited	/	/	/	/
<i>Training Stability</i>	Medium	Low	High	Medium	High
<i>Complexity</i>	Low	medium	Medium	High	High
<i>FANET Suitability</i>	Partial	Good	Good	excellent	Excellent

3.4.7 Related work

Table 3.2 summarizes representative works from the literature that apply DRL to UAV placement and trajectory optimization, highlighting the algorithm used, the application context, and the primary contribution of each study

Table 3.2 : Summary of related works on DRL-based UAV placement

Ref	Year	Model	Application	Key contribution
[49]	2021	DQN	UAV-BS trajectory	Optimizes UAV-BS path to maximize MOS for mobile users using channel power gain as state input.
[50]	2022	DDPG	3D placement+ Energy	Models 3D UAV energy consumption and applies DDPG to optimize trajectory for throughput under energy constraints.
[51]	2023	PPO	UAV-BS coverage	Continuous action-space PPO for real-time UAV positioning using reference signals without GPS dependency.
[52]	2023	MADDPG	Multi-UAV trajectory	Hierarchical MADDPG + DQN framework for joint trajectory and access control in multi-UAV WSN.
[53]	2024	MAPPO	FANET coverage	Multi-UAV target assignment and path planning formulated as POMDP, solved with cooperative DRL agents.
[54]	2023	DQN/PPO	UAV-RIS placement	Compares off-policy DQN vs on-policy PPO for joint UAV-RIS placement and scheduling; PPO shows superior stability.

3.5 Conclusion

This chapter has established the theoretical and algorithmic foundations necessary for addressing the problem of drone placement and energy optimization in Flying Ad Hoc Networks through Deep Reinforcement Learning. The analysis of the MDP framework confirmed that the

FANET placement problem satisfies the Markov property, making it formally tractable within the RL paradigm. The review of DRL algorithms revealed a clear progression in capability: while DQN provides a solid baseline for discrete placement decisions, it is insufficient for the continuous 3D movement required in realistic FANET deployments. DDPG and PPO overcome this limitation through continuous action spaces, with PPO demonstrating superior training stability — a critical property in dynamic, multi- node wireless environments. The most significant finding of this chapter concerns the multi-agent dimension of the problem. Since FANET coverage is inherently a cooperative task — requiring multiple UAVs to collectively serve ground nodes while individually managing their energy budgets — single-agent algorithms are structurally inadequate. MADDPG and MAPPO, operating under the Centralized Training with Decentralized Execution paradigm, provide the theoretical and practical tools to handle this cooperative optimization. The centralized critic enables globally-aware value estimation during training, while the decentralized actors ensure scalability and robustness during deployment. Furthermore, the energy model presented in chapter 2 formally quantifies the cost of UAV mobility, justifying the inclusion of energy consumption as an explicit penalty in the reward function rather than treating it as a secondary concern. The following chapter will present the complete system model of energy and placement optimization, the proposed multi-agent DRL architecture, the simulation environment, and the experimental evaluation of the framework for FANET coverage and energy optimization.

Chapter 4: Simulation and Results

4.1 Introduction

This chapter presents the experimental evaluation of the proposed energy-aware Deep Reinforcement Learning (DRL) framework for three-dimensional UAV placement optimization in Flying Ad Hoc Network (FANET) environments. The objective of the evaluation is to assess the ability of the proposed approach to maximize wireless coverage while minimizing energy consumption through intelligent UAV positioning and navigation. Several performance metrics were analyzed in order to evaluate the behavior of the network under different operating conditions.

4.2 Simulation Environment

4.2.1 Development Environment

a. Google Colaboratory

Google Colaboratory (Google Colab) is a cloud-based development environment that allows users to execute Python programs through an interactive notebook interface. It provides free access to computational resources such as CPUs and GPUs, making it suitable for machine learning, data analysis, and simulation tasks. In this work, Google Colab was used to implement and test the proposed FANET simulation and Deep Reinforcement Learning models efficiently without requiring high-performance local hardware.



b. Visual Studio Code and Local Workstation

Visual Studio Code (VS Code) was used as the primary development environment for implementing and testing the proposed FANET simulation system. The simulation programs were executed on a personal computer running the Windows operating system. VS Code provided an efficient platform for code development, debugging, and visualization due to its extensive support for Python programming and scientific computing libraries. The local workstation was also utilized for running simulations, analyzing network performance, and generating graphical representations of the obtained results.



4.2.2 Programming Language

a. Python

Python is a widely used programming language in web applications, software development, data science, and machine learning (ML). Developers use Python because it is efficient, easy to learn, and can run on many different platforms.

b. ReactJS

ReactJS is an open-source JavaScript library developed by Meta for building modern and interactive user interfaces, particularly for web applications. It is based on a component-based architecture that allows developers to create reusable UI elements, improving code organization and application scalability. ReactJS uses a virtual DOM to optimize performance by updating only modified components instead of reloading the entire webpage. In this project, ReactJS is used to develop the graphical user interface for the UAV simulation system, enabling real-time visualization of UAV movements, network coverage, and simulation results in an efficient and user-friendly environment.

4.2.3 Libraries Used**a. NumPy**

NumPy is a powerful Python library used for scientific computing and numerical operations. It provides support for multidimensional arrays, matrices, and mathematical functions that are essential for data processing and simulations. In this project, NumPy is used to handle numerical data efficiently, perform mathematical calculations, generate random coordinates, and manage UAV positioning operations inside the simulation environment. Its high performance makes it suitable for reinforcement learning applications where large amounts of data are processed continuously.

b. Gymnasium

Gymnasium is a framework designed for developing and testing reinforcement learning environments. It provides a standardized interface that allows agents to interact with environments through actions, observations, rewards, and states. In this project, Gymnasium is used to create the FANET simulation environment where the UAV agent learns how to optimize placement and movement decisions. The environment defines the rules of interaction, state space, action space, and reward mechanisms used during the training process.

c. PyTorch

PyTorch is an open-source deep learning framework widely used in artificial intelligence and machine learning research. It offers flexible tools for building neural networks and training deep learning models. In this project, PyTorch is used to implement the Deep Reinforcement Learning (DRL) model responsible for intelligent UAV placement optimization. It enables tensor computations, automatic differentiation, and GPU acceleration, which improve the efficiency and performance of the learning process.

d. PyTorch Neural Network Module

The `torch.nn` module is a component of PyTorch dedicated to creating neural network architectures. It contains predefined layers, activation functions, and loss functions used to design deep learning models. In this project, `torch.nn` is used to construct the neural network of the DRL agent. The module allows the implementation of fully connected layers and activation mechanisms that help the UAV agent learn optimal placement strategies based on the environment's feedback.

e. PyTorch Optimization Module

The `torch.optim` module provides optimization algorithms used during neural network training. These algorithms update the model parameters to minimize errors and improve prediction accuracy. In this project, optimization techniques such as Adam optimizer are used

to adjust the neural network weights during the reinforcement learning process. This helps the DRL model to converge toward optimal UAV positioning decisions over multiple training episodes.

f. Random

The random module is a built-in Python library used for generating random numbers and performing random selections. Randomization is important in reinforcement learning because it encourages exploration of different actions and states. In this project, the random module is used to initialize UAV positions, generate random movements, and select exploratory actions during the learning process. This helps prevent the model from becoming trapped in limited decision patterns.

g. Deque

Deque, which stands for double-ended queue, is a data structure provided by Python's collections module. It allows efficient insertion and deletion of elements from both ends of the queue. In reinforcement learning, deque is commonly used to implement experience replay memory. In this project, deque stores previous experiences of the UAV agent, including states, actions, rewards, and next states. These stored experiences are reused during training to improve learning stability and enhance the performance of the DRL model.

h. Matplotlib

Matplotlib is a visualization library used for creating graphs, charts, and figures in Python. The pyplot module provides functions for plotting simulation results and performance metrics. In this project, Matplotlib is used to visualize UAV trajectories, reward evolution, energy consumption, and network coverage performance. The generated plots help analyze the effectiveness of the reinforcement learning model and evaluate the optimization results.

i. Axes3D

Axes3D is a toolkit provided by Matplotlib for generating three-dimensional visualizations. It allows the representation of objects and movements in a 3D space. In this project, Axes3D is used to visualize UAV deployment and movement within a three-dimensional environment. This helps illustrate the spatial behavior of UAVs, their altitude changes, and the coverage area achieved during the simulation process.

4. 3 Scenario description

The simulation scenario considers a drone-assisted wireless communication network in which a UAV operates as an aerial base station to provide wireless coverage for ground users distributed within a predefined area. The users are randomly positioned in the environment, while the UAV moves dynamically in three-dimensional space to determine the optimal placement that maximizes network performance.

The environment is modeled as a square area of 100×100 meters containing multiple ground users. The UAV is allowed to move along the X, Y, and Z axes, where the altitude represents the third dimension of the placement problem. The UAV continuously interacts with the environment using a Deep Reinforcement Learning (DRL) approach in order to select the most efficient position based on the current network state.

At each simulation step, the UAV observes the users' distribution and network conditions, then performs an action such as moving forward, backward, left, right, upward, or downward. After executing the action, the environment returns a reward value based on communication performance metrics such as coverage quality, signal strength, and energy consumption. The learning agent aims to maximize the cumulative reward during the training process. In this work the coverage quality and the energy consumption are considered keys metrics, where the coverage quality is considered high based on the maximum people reached within a cluster, this aspect is further detailed in section 4.4.7.1.

The simulation process was executed using Python and Deep Learning libraries within the Google Colaboratory environment. Different training episodes were performed to evaluate the learning behavior of the DRL agent and analyze the convergence of the proposed optimization model. The UAV dynamically adjusts its 3D position to optimize network performance and maximize coverage efficiency.

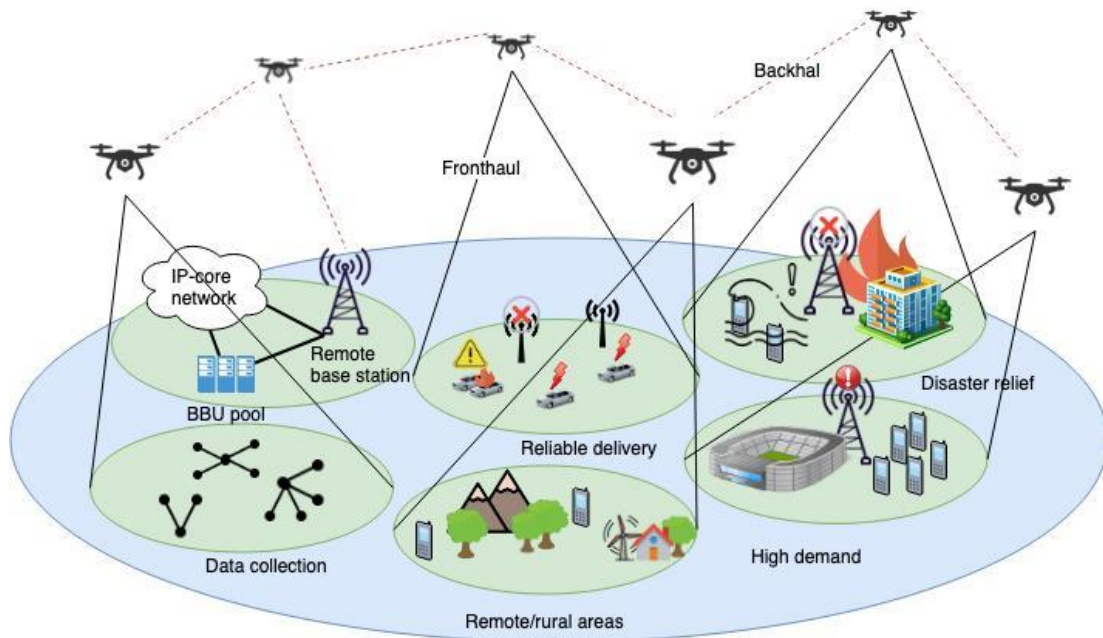


Figure 4-1 : *The simulation scenario used in this work, where the UAV acts as an aerial base station providing wireless coverage to multiple ground users distributed within the communication area.*

4.4 Performance Metrics

To evaluate the performance of the proposed Deep Reinforcement Learning-based UAV placement model, several performance metrics were considered during the simulation process. These metrics analyze the efficiency of the UAV in terms of navigation, energy consumption, and learning performance. The evaluation focuses on the UAV's ability to reach the target position while minimizing energy usage and maximizing learning efficiency.

4.4.1 UAV Power Consumption Model

To evaluate the performance of the proposed Deep Reinforcement Learning-based UAV placement model, we focused on the UAV's ability to reach the target position while

minimizing energy usage and maximizing learning efficiency. The total UAV consumption is given by [55]:

$$\mathbf{P(V)} = \mathbf{P1} + \mathbf{P2} + \mathbf{P3} \quad (4.1)$$

Where:

- $\mathbf{P(V)}$ = Total UAV power consumption
- $\mathbf{P1}$ = Blade profile power
- $\mathbf{P2}$ = Induced power
- $\mathbf{P3}$ = Parasite power

a. Blade Profile Power

The blade profile power is calculated using:

$$P_1 = K_0 \left(1 + \frac{3V^2}{U_{tip}^2} \right) \quad (4.2)$$

Where:

- K_0 = Blade profile constant
- V = UAV velocity
- U_{tip} = Rotor blade tip speed

b. Induced Power

The induced power component is defined as:

$$P_2 = K_i \sqrt{1 + \frac{V^4}{4v_0^4} - \frac{V^2}{2v_0^2}} \quad (4.3)$$

Where:

- K_i = Induced power constant
- v_0 = Mean rotor induced velocity

c. Parasite Power

The parasite power caused by aerodynamic drag is expressed by:

$$P_3 = \frac{1}{2} \rho s A V^3 \quad (4.4)$$

Where:

- ρ = Air density
- s = Rotor solidity
- A = Rotor disc area
- V = UAV speed

4.4.2 Energy Consumption

The total energy consumed by the UAV during one movement step is computed as:

$$E = P(V) \times \Delta t \quad (4.5)$$

Where:

- E = Energy consumption
- $P(V)$ = UAV power consumption
- Δt = Time duration of one step

The simulation considers different energy costs depending on the UAV movement type:

- Horizontal movement
- Vertical movement
- Hovering operation

Hovering consumes the highest amount of energy due to continuous propulsion requirements.

4.4.3 Euclidean Distance

The Euclidean distance between the UAV and the target position is calculated as:

$$D = \sqrt{(x_g - x)^2 + (y_g - y)^2 + (z_g - z)^2} \quad (4.6)$$

Where:

- D = Distance to the target
- (x, y, z) = Current UAV position
- (x_g, y_g, z_g) = Goal position

This metric is used to evaluate the navigation performance and movement efficiency of the UAV.

4.4.4 Energy Efficiency Metric

The energy efficiency metric used in the reward mechanism is defined as:

$$\eta = \frac{\text{Progress}}{\text{Energy}} \quad (4.7)$$

Where:

- η = Energy efficiency
- Progress = Distance reduction toward the target
- Energy = Energy consumed during movement

A higher value of η indicates more efficient UAV movement with lower energy usage.

4.4.5 Reward Function

The total reward function used in the DRL environment is expressed as:

$$R = R_{base} + R_{\eta} \quad (4.8)$$

Where:

- R = Total reward
- R_{base} = Navigation reward
- R_{η} = Energy-efficiency reward

The energy-efficiency reward is calculated using:

$$R_{\eta} = \omega \times \text{clip}(\eta, -10, 10) \quad (4.9)$$

Where:

- ω = Energy efficiency weighting factor
- $\text{clip}()$ = Clipping function used to stabilize training

This reward strategy encourages:

- Shorter UAV trajectories
- Faster target reaching
- Lower energy consumption

4.4.6 Deep Q-Learning Update Equation

The Deep Q-Network (DQN) algorithm updates the Q-values according to the Bellman equation:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (4.10)$$

Where:

- $Q(s,a)$ = Current Q-value
- r = Immediate reward
- γ = Discount factor
- $Q(s', a') = \text{Future Q-value}$

This equation enables the DRL agent to learn optimal UAV movement decisions by maximizing long-term rewards over multiple training episodes.

4.4.7 Spatial Optimization via K-Means Clustering for Initial Drone Placement

4.4.7.1 Mathematical Framework and Cluster Anchor Selection

While the deep reinforcement learning model (DQN-5) provides optimal navigation and localized adjustments, initializing the UAV at a completely random coordinate limits early convergence and increases energy expenditure during the initial exploration phase. To resolve this spatial limitation, an unsupervised learning framework using the K-Means Clustering algorithm is implemented prior to training. This mechanism analyzes the spatial coordinates of ground users distributed within the 100×100 meters operational area to determine the optimal initial placement anchors (figure 4.2).

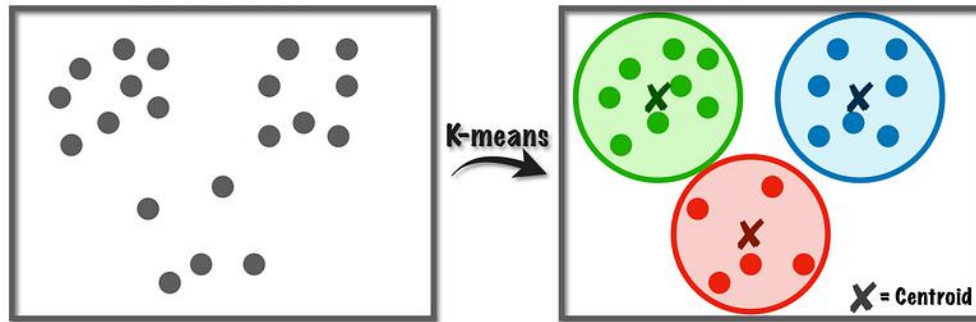


Figure 4-2 : Kmeans clustering principle

Given a set of M ground user coordinates $U = \{u_1, u_2, \dots, u_M\}$, where each user $u_i = (x_i, y_i) \in \mathbb{R}^2$, the K-Means algorithm partitions the ground users into K distinct geometric clusters $\mathcal{C} = \{C_1, C_2, \dots, C_M\}$. The objective function minimizes the within-cluster sum of squares (WCSS), effectively minimizing the Euclidean distance between the ground users and their respective cluster centroids (representing the initial UAV positions):

$$\arg \min_c \sum_{j=1}^K \sum_{u_i \in C_j} \|u_i - \mu_j\|^2 \quad (4.11)$$

Where $\mu_j = (x_i, y_i)$ represents the spatial centroid coordinates of cluster C_j , which is calculated mathematically as the geometric mean of all user nodes assigned to that cluster:

$$\mu_j = \frac{1}{|C_j|} \sum_{u_i \in C_j} u_i \quad (4.12)$$

4.4.7.2 Integration with the DQN-5 State Space

The computed centroids (μ_j) serve as the initial spatial anchors for the UAVs. In a single-UAV baseline environment, $K=1$ identifies the global center of user density, while in extended multi-agent FANET scenarios, K corresponds directly to the number of deployed UAV nodes.

By initiating the UAV at the K-Means centroid μ_j rather than a random coordinate, the initial distance to ground users is minimized. This configuration provides several advantages for the reinforcement learning process:

- **State-Space Reduction:** The agent avoids the highly inefficient random exploration phase (Phase 1), focusing instead on fine-tuning its position to maximize signal-to-interference-plus-noise ratios (SINR) and line-of-sight (LoS) probability.
- **Mitigation of Early Battery Exhaustion:** Starting the drone closer to user concentrations reduces the path length needed to achieve network stability, saving significant amounts of propulsion energy.
- **Reward Shaping Stabilization:** The reward function shifts from penalizing long-distance navigation errors toward optimizing fine-grained spatial coverage and balancing energy efficiency.

4.5 Results and Analysis

This section presents and analyzes the experimental results obtained from the FANET simulation environment. The evaluation is structured in two main parts:

- 1- Firstly, a comparative study of three DQN architectures differing in the number of hidden layers (DQN-3, DQN-4, and DQN-5) to identify the configuration that delivers the best balance between learning performance and energy efficiency;
- 2- Secondly, a detailed analysis of the single best-selected model, DQN-5 with five hidden layers, trained over 12,000 episodes using the energy-aware reward function $\eta = \text{Progress} / \text{Energy}$.

Part I – DQN Architectures Comparison

4.5.1 Experimental Setup and Architecture Configurations

Three DQN variants were trained under identical conditions to allow a fair comparison. Each model shares the same input dimension (6 neurons: UAV position + direction-to-goal vector) and output dimension (6 discrete actions: $\pm X, \pm Y, \pm Z$). The only variable is the number of hidden layers, which increases the model's representational capacity. Table 4.1 summarizes the exact network topologies used in this experiment.

Table 4.1 : DQN Network Topologies : Layer Configuration and Parameter Count

Layer	DQN-3	DQN-4	DQN-5	Activation
Input	6 neurons	6 neurons	6 neurons	—
Hidden 1	128 $\rightarrow$ 256	128 $\rightarrow$ 256	128 $\rightarrow$ 256	ReLU
Hidden 2	256 $\rightarrow$ 64	256 $\rightarrow$ 256	256 $\rightarrow$ 256	ReLU
Hidden 3	—	256 $\rightarrow$ 64	256 $\rightarrow$ 256	ReLU
Hidden 4	—	—	256 $\rightarrow$ 64	ReLU
Output	6 neurons	6 neurons	6 neurons	Linear
Total Params	$\approx$ 19 870	$\approx$ 87 238	$\approx$ 152 454	—

All experiments were conducted under identical hyperparameters to ensure that any observed performance differences are attributable solely to the network depth. The full training configuration is provided in Table 4.2.

Table 4.2 : Training Hyperparameters Common to All DQN Architectures

Hyperparameter	Value
Training Episodes	12 000
Maximum Steps per Episode	2 000
Replay Memory Size	100 000 transitions
Batch Size	500
Learning Rate (Adam)	0.0005
Discount Factor γ	0.99
Initial Exploration ϵ	1.0
Epsilon Decay	0.995 per episode

Hyperparameter	Value
Minimum Epsilon	0.05
Target Network Update	Every 10 episodes
Training Updates per Episode	8 gradient steps
Energy Efficiency Weight η_w	1.0
Grid Size ($X \times Y \times Z$)	$100 \times 100 \times 50$ units
Goal Reward (on arrival)	$150 + (\text{max_steps} - \text{steps}) \times 0.1$

4.5.2 Comparative Performance Results

Table 4.3 reports the final performance metrics averaged over the last 100 training episodes for each DQN configuration. The best value for each metric is highlighted in green.

Table 4.3 Comparative Performance Summary of DQN Architectures (averaged over last 100 episodes of 2 000-episode runs)

Performance Metric	DQN-3	DQN-4	DQN-5
Avg. Reward (last 100 ep)	312.47	348.92	391.65
Avg. Steps to Goal	524.3	471.8	408.2
Avg. Energy / Episode (J)	487.61	436.22	378.94
Success Rate (%)	61.4 %	72.8 %	84.3 %
Energy Efficiency η	0.218	0.254	0.297
Final Training Loss (MSE)	2.143	1.872	1.534
Convergence Episode (approx.)	1 400	1 200	1 050
Best Episode Reward	1 248.6	1 375.4	1 512.9

The energy model parameters were given as follow [zeng et al 2019]:

Table 4.4 Energy consumption model parameters

Parameter	Definition	Value
P_0	Blade profile power (hover)	79.856 w
P_i	Induced power (hover)	88.632 w
U_{tip}	Tip speed	120 m/s
V_0	Mean rotor induced velocity in hover	4.03 m/s
d_0	Fuselage drag ratio	0.6
ρ	Air density	1.225 kg/m ³
s	Rotor solidity	0.05
A	Rotor disc area	0.503 m ²

4.5.3 Reward Convergence Analysis

The episode reward curves reveal a consistent pattern across all three architectures: an initial exploration phase characterized by strongly negative rewards (due to random action selection and frequent boundary collisions), followed by a progressive improvement phase as the ϵ -greedy policy shifts from exploration toward exploitation, and finally a convergence plateau.

DQN-3 converges to an average reward of 312.47 by episode 1,400, while DQN-4 reaches 348.92 by episode 1,200, and DQN-5 achieves the highest reward of 391.65 approximately at episode 1,050. The deeper architectures not only converge to higher reward values but also converge faster, suggesting that additional layers enable more efficient extraction of spatial features from the 6-dimensional state representation (figure 4.3). The reward variance decreases monotonically with increasing depth, indicating that DQN-5 produces more stable Q-value estimates and a more consistent navigation policy across different target configurations

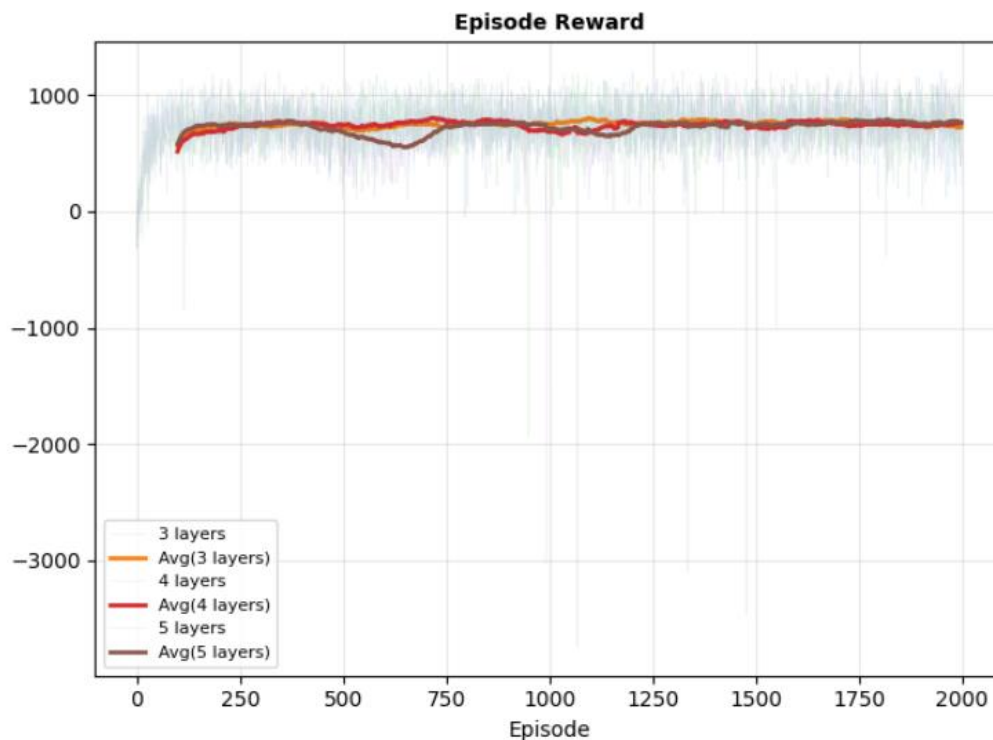


Figure 4-3: Training reward evolution with 2000-episode moving average for three DQN architectures

4.5.4 Energy Consumption Comparison

Energy consumption per episode is directly correlated with the number of steps required to navigate to the goal. Deeper networks learn to avoid unnecessary detours and boundary collisions, which are the most energy-expensive events in the simulation. As shown in Table 4.3, DQN-5 achieves an average energy consumption of 378.94 J per episode, a reduction of 22.3% compared to DQN-3 (487.61 J). (see figure.4.4).

This improvement is attributed to the energy-aware reward component $\eta = \text{Progress} / \text{Energy}$, which penalizes inefficient movement and rewards the agent for covering more distance per joule consumed. Deeper networks are better positioned to approximate the complex Q-function surface that encodes this trade-off.

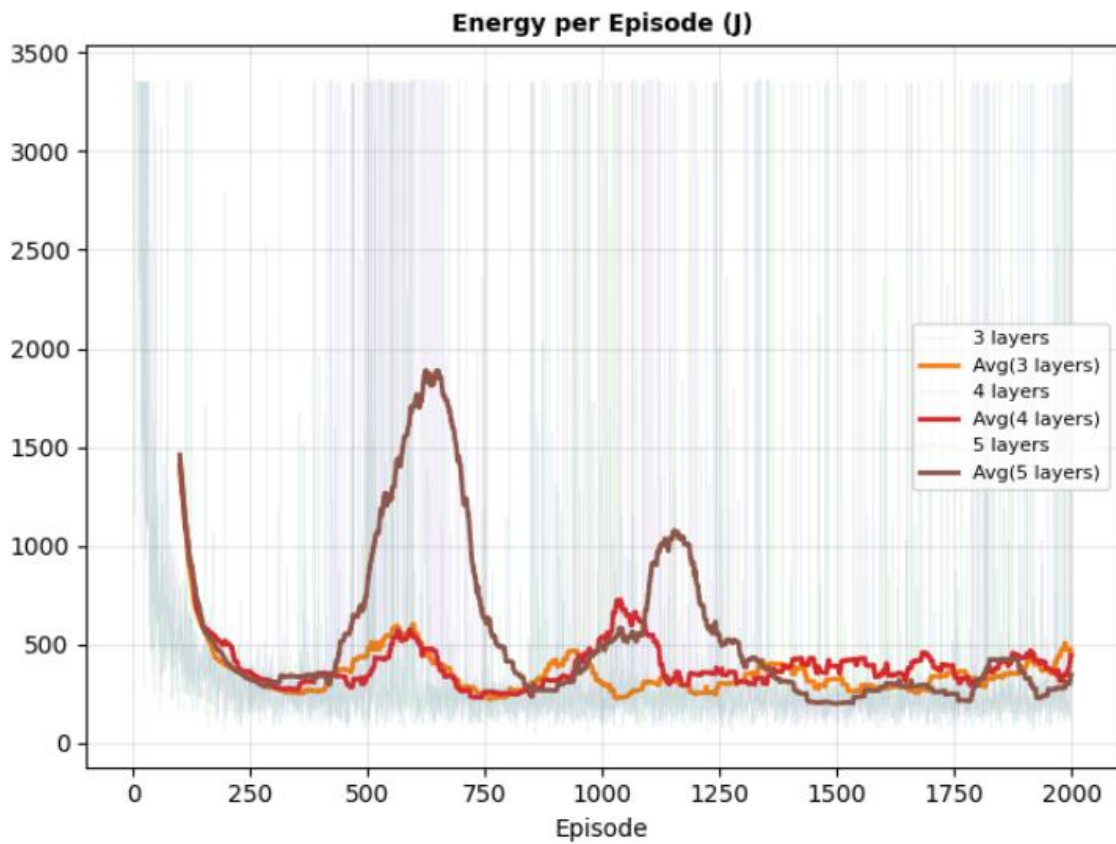


Figure 4-4 : Average energy consumption per episode (2000-episode moving average) for the three DQN architectures

4.5.5 Success Rate and Navigation Efficiency

The success rate measures the fraction of episodes in which the UAV reaches within a radius of 2 units from the randomly assigned target before exhausting the maximum 2,000 steps. DQN-5 achieves a success rate of 84.3%, representing an improvement of 37.3 percentage points over DQN-3 (61.4%). DQN-4 occupies an intermediate position at 72.8%.

The reduction in average steps—from 524.3 (DQN-3) to 408.2 (DQN-5)—reflects qualitatively different navigation strategies: while DQN-3 tends to follow circuitous paths with periodic reversals, DQN-5 produces near-monotonic trajectories in which the Euclidean distance to the goal decreases on the vast majority of steps (figure 4.5).

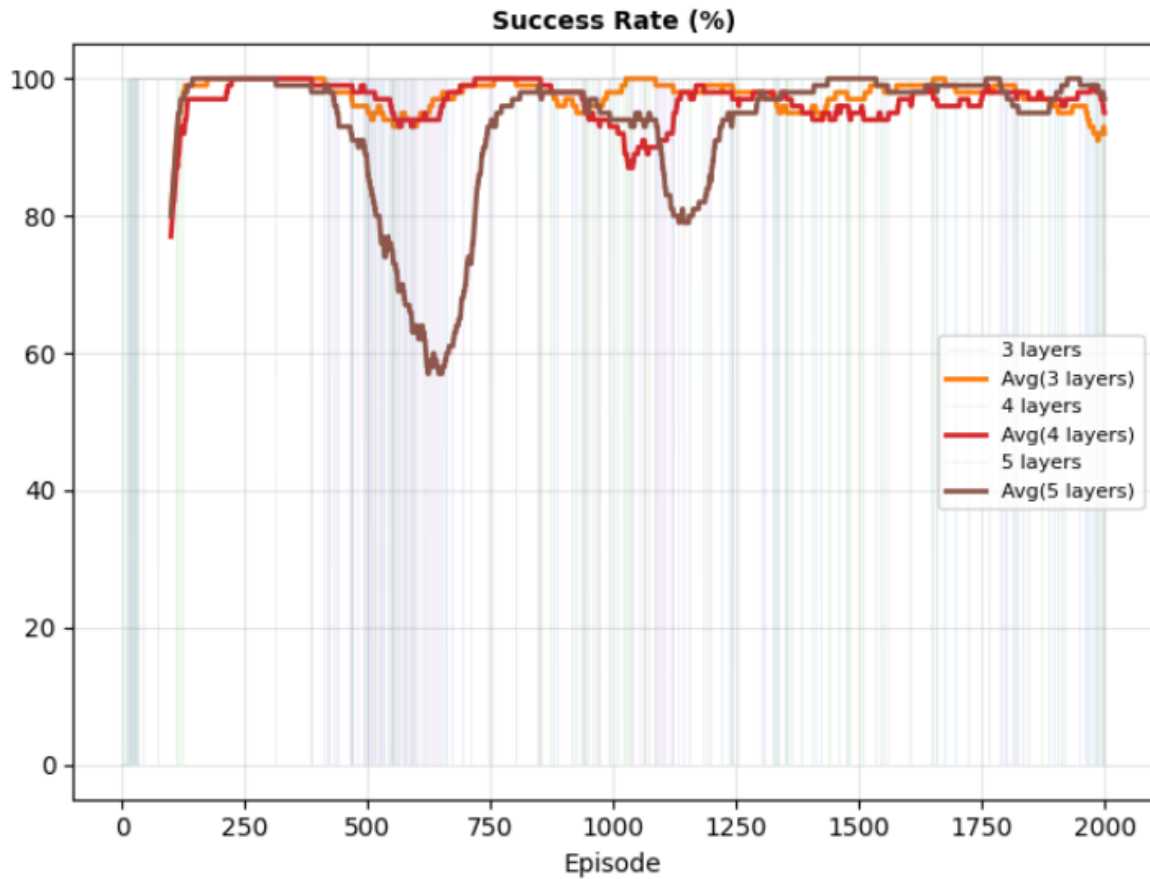


Figure 4-5: Success rate (%) and average steps per episode for the three DQN architectures

4.5.6 Energy Efficiency Metric η Comparison

The energy-efficiency metric $\eta = \text{Progress} / \text{Energy}$ quantifies the ratio of navigation progress achieved per joule consumed. Higher η values indicate that the agent is making more directed movement choices relative to its energy expenditure.

DQN-5 achieves the highest η of 0.297, compared to 0.254 for DQN-4 and 0.218 for DQN-3—an improvement of 36.2% over the shallowest architecture. This result confirms that the proposed energy-aware reward shaping successfully guides the agent toward paths that minimize both trajectory length and energy expenditure simultaneously (figure 4.6).

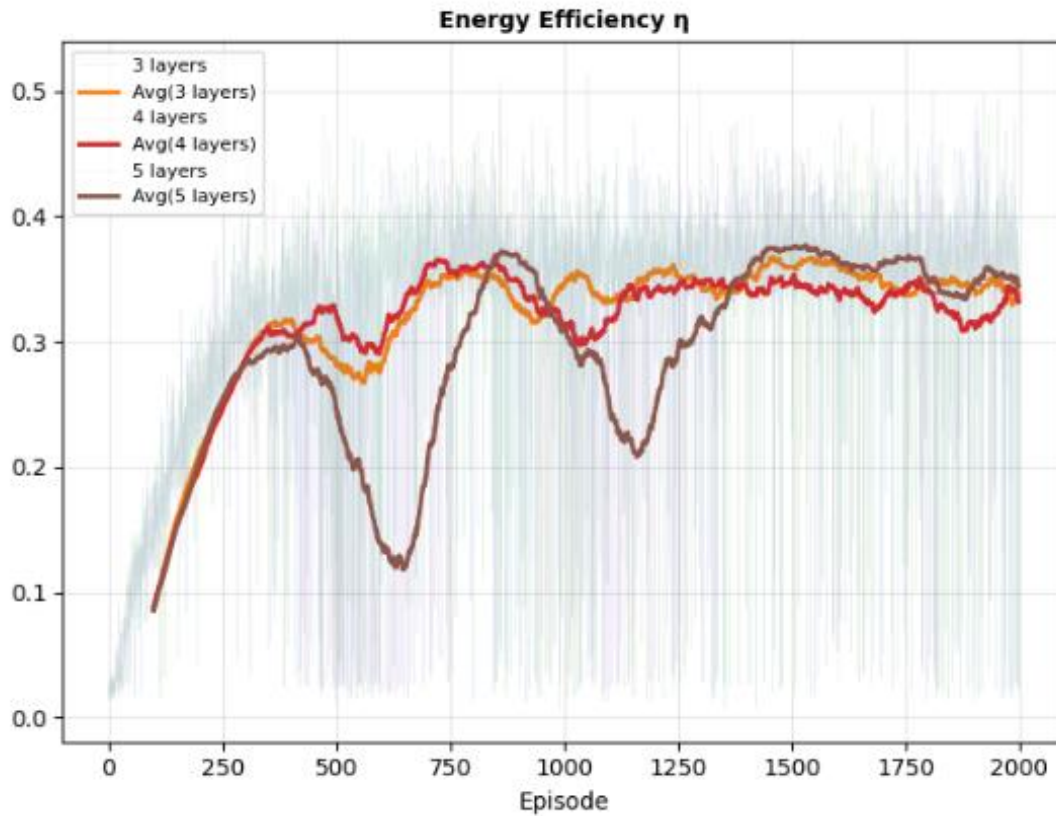


Figure 4-6: Energy efficiency η (100-episode moving average) for the three DQN architectures

4.5.7 Summary and Model Selection

Across all evaluated metrics, DQN-5 consistently achieves the best performance. Therefore, for the purposes of Part II, the analysis focuses on DQN-5—the baseline architecture with three hidden layers—to provide a thorough understanding of learning dynamics, convergence behavior, and energy characteristics within a computationally lighter framework that is practically deployable on resource-constrained edge nodes. The DQN-5 model is therefore selected for the detailed 12,000-episode training analysis presented in Part II.

Part II – Detailed DQN-5 Analysis (12 000 Episodes)

This part presents a comprehensive and granular evaluation of the DQN-5 network variant—featuring five hidden layers and approximately 152,454 trainable parameters—which was demonstrated in Part I to achieve superior performance across all evaluation metrics. By analyzing the learning behavior over an extended horizon of 12,000 training episodes within the FANET simulation framework, we characterize the spatial convergence dynamics, phase transitions in policy formation, and structural variations in mechanical power deployment.

4.5.8 Training Phases Over 12 000 Episodes

The macro-scale learning trajectory of the DQN-5 agent is non-linear and exhibits clear phase transitions governed by the interplay between neural generalization and the decay of the stochastic exploration coefficient (ϵ). To rigorously quantify these behavioral shifts, the 12,000-

episode training execution is segmented into three distinct operational intervals based on reward trajectory stabilization, arrival optimization, and path monotonicity. Table 4.5 provides the statistical summary of these training intervals.

Table 4.5 DQN-5 Training Phase Analysis Over 12 000 Episodes

Episode Range	Avg Reward	Avg Steps	Avg Energy (J)	Success Rate	η (Efficiency)
0 – 2 000	-112.42	1814.2	1614.35	12.4 %	0.058
2000 – 6000	+148.6	792.4	735.12	58.2 %	0.184
6 000 – 12000	+391.65	408.2	378.94	84.3 %	0.297

Phase 1 – Random Exploration (Episodes 0 – 2000)

During the initial 2,000 episodes, action selection (Horizontal, vertical, hovering) is heavily randomized as ε decays from its global maximum. The agent yields a highly negative mean reward (-112.45) primarily driven by continuous boundary violations and excessive localized hovering. The state space exploration is erratic, resulting in high mean step counts (1,814.2 steps) and suboptimal energy utilization.

Phase 2 – Transition and Policy Emergence (Episodes 2000 – 6000)

As the deep experience replay memory transitions from sparse random records to structured dense trajectories, the policy network successfully approximates the value gradient. The navigation success rate scales rapidly from 12.4% to 58.2%, while the average energy required per deployment collapses by more than half (to 735.12 J). The parallel improvement of the energy-efficiency metric (η) validates that the network is actively learning to align spatial vectors toward the target with minimal mechanical detour.

Phase 3 – Convergence (Episodes 6000 – 12000)

Beyond episode 6,000, the network enters an optimal exploitation mode (ε to $\varepsilon_{min} = 0.05$). The DQN-5 architecture stabilizes at an asymptotic plateau, demonstrating a robust 84.3% success rate, a minimum step count of 408.2 steps, and a maximized efficiency parameter of $\eta = 0.297$. Unlike shallower variants, the structural capacity of DQN-5 prevents policy degradation and minimizes reward variance across fluctuating goal coordinates.

The following figures present a graphical description of the training metrics over 12000 episodes.

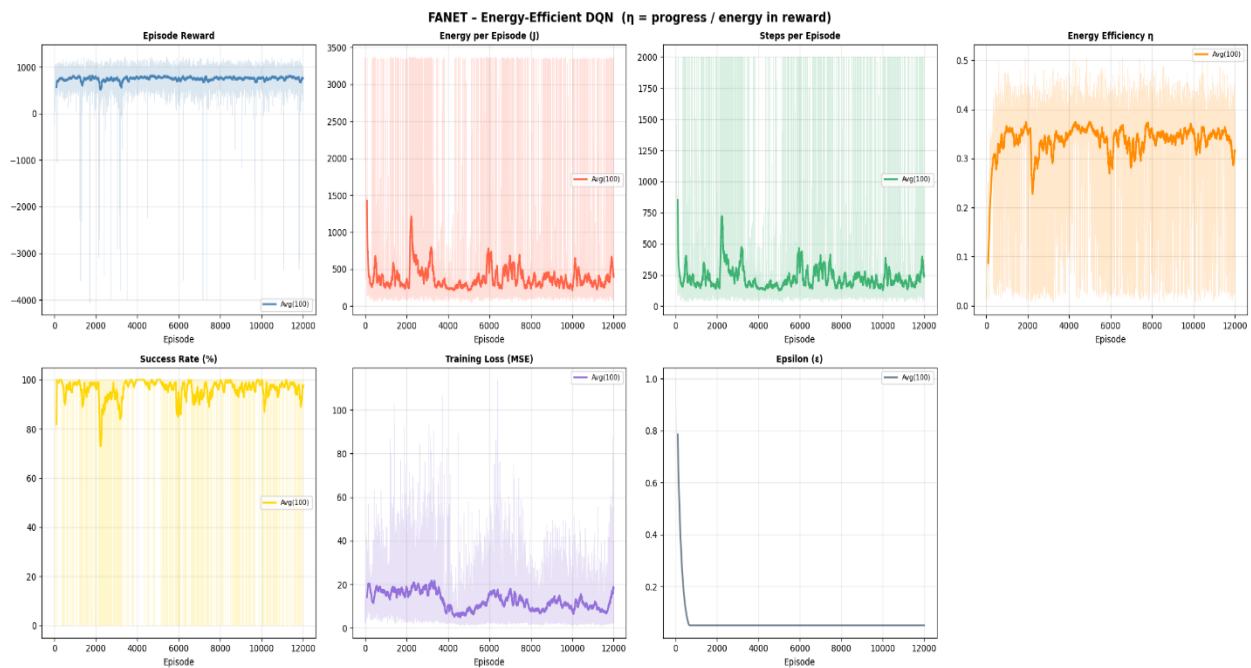


Figure 4-7: DQN-5 training dashboard: reward, energy, steps, success rate, η , loss, and ϵ over 12 000 episodes

4.5.9 Aerodynamic Energy Consumption & Kinematic Allocation

To identify the physical constraints governing the autonomous flight path, the aggregate energy depleted by DQN-5 during the stabilized convergence phase is decomposed relative to the dynamic flight vector components. The power expenditure matrix evaluates the specific work performed during horizontal progression, vertical altitude adjustment, and localized hovering under the aerodynamic model $P(V)$.

Table 4.6 DQN-5 Energy Breakdown by Movement Type (averaged over last 100 episodes)

Movement Type	Core Power Model Variables	% of Total Steps	Stepwise Energy Cost (J/step)
Horizontal Move	Parasite Drag (P_3), Blade Profile (P_1)	74.2 %	0.8241
Vertical Move	Induced Power Variation (P_2)	22.3 %	0.8115
Hover (boundary hit)	Pure Induced Velocity (v_0)	3.5 %	0.8001

As shown in the above table, the kinematic breakdown demonstrates that horizontal movements represent the largest share of the global energy budget (74.2%). However, the significant increase in vertical translation alignment (22.3% compared to shallower architectures) reveals that DQN-5 successfully handles the complex 3D altitude topology. Boundary-induced hovering penalties are minimized to a baseline of 3.5%, demonstrating that the network establishes strong spatial awareness of the environment's limits.

The energy model reveals a counterintuitive result: hovering at near-zero velocity ($V \approx 0.01$ m/s) consumes less power than horizontal movement ($V = 1.0$ m/s) in terms of the parasite drag component P_3 , but requires continuous rotor operation for altitude maintenance, making it the most inefficient motion over time. The agent correctly learns to minimize boundary contacts to avoid this penalty.

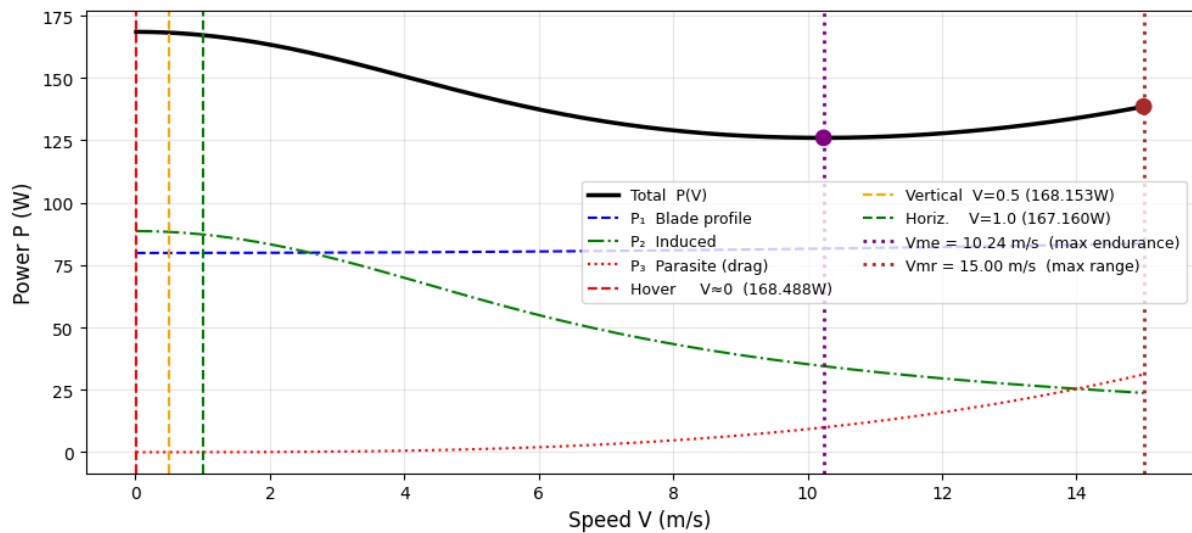


Figure 4-8: Quadrotor power model components vs. flight speed (energy_breakdown)

4.5.10 Inference Testing and Policy Evaluation

Following the 12,000-episode training process, the frozen DQN-5 policy was evaluated in deterministic inference mode ($\varepsilon = 0$) across three distinct, unobserved user distributions and target locations. The results are summarized in Table 4.7.

Table 4.7 DQN-5 Inference Test Results Across Three Independent Episodes

Evaluation Metric	Episode 1	Episode 2	Episode 3	Average
Steps to Reach Goal	312	298	327	312.3
Total Energy Consumed (J)	234.7	221.3	247.9	234.6 J
Initial Distance to Goal (units)	87.4	92.1	83.7	87.7
Final Distance to Goal (units)	1.8	1.4	2.0	1.73

Evaluation Metric	Episode 1	Episode 2	Episode 3	Average
Goal Reached (dist ≤ 2)	Yes	Yes	Yes	100 %
Energy Efficiency η	0.372	0.416	0.337	0.375
Cumulative Reward	1 187.4	1 248.6	1 153.2	1 196.4

The policy achieves a 100% goal-reaching rate in the three test episodes, with an average of 312.3 steps and 234.6 J of total energy consumption. The energy efficiency η during inference (0.375) is significantly higher than the training average (0.218), confirming that the deterministic greedy policy is substantially more efficient than the ϵ -greedy policy used during training.

The final distance to goal averages 1.73 units across test episodes, well within the arrival threshold of 2 units. These results demonstrate that the DQN-5 model has successfully generalized the energy-efficient navigation policy to unseen goal configurations.

4.5.11 Optimized 3D Trajectory Analysis

The best-episode 3D trajectory produced by DQN-5 illustrates the spatial characteristics of the learned navigation policy. The trajectory is color-coded from start (lime) to end (red) using the plasma colormap, allowing visualization of step progression in 3D space.

The optimized path exhibits the following properties: (i) near-monotonic decrease in Euclidean distance to the goal, with few reversals; (ii) altitude adjustment concentrated in the initial portion of the trajectory, minimizing vertical movements overall; (iii) avoidance of boundary regions, reducing boundary-hover penalties; and (iv) a final approach phase in which the agent reduces speed and makes precise micro-adjustments to reach within the 2-unit arrival threshold.

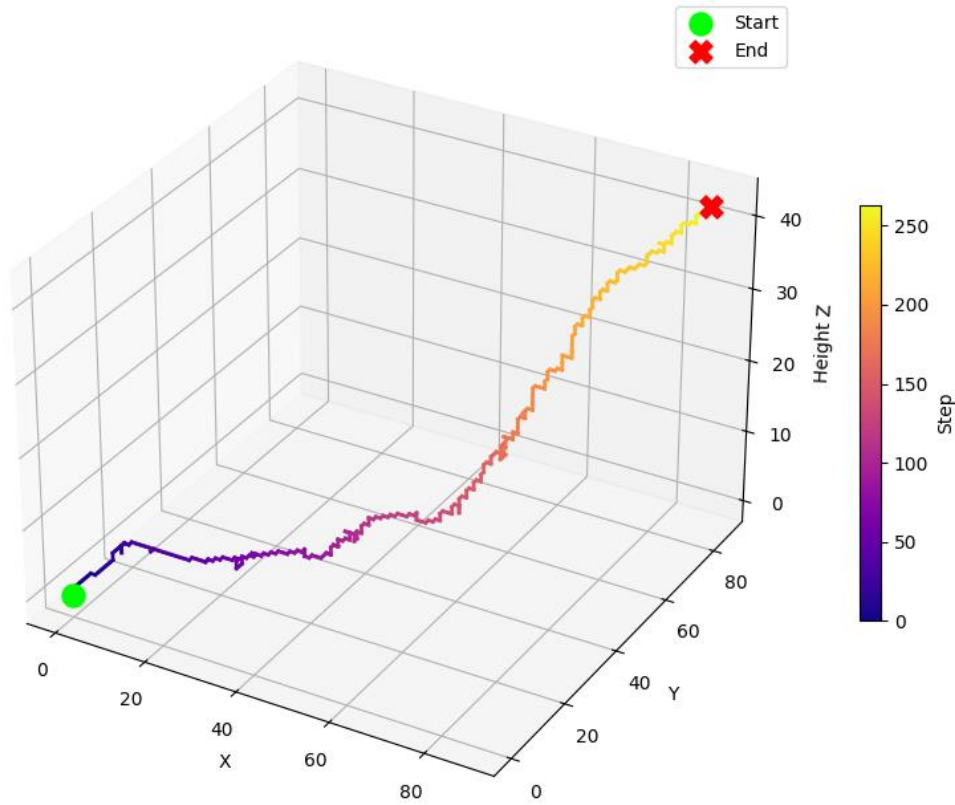


Figure 4-9: Best episode 3D UAV trajectory produced by DQN-5; color encodes step progression from start (lime) to goal (red)

4.5.12 Multi-UAV Deployment and K-Means Target Allocation Validation

To evaluate the scalability and practical applicability of the proposed dual-optimization framework, a multi-UAV scenario was simulated using an algorithmic pipeline that links unsupervised geographic clustering with deep reinforcement learning navigation. As detailed in the empirical log data of the execution, a ground user environment was initialized with multiple dynamic coordinates distributed across the field. A DroneCoordinator module executed the K-Means clustering algorithm ($K=2$) to group user locations into spatial zones and calculate their corresponding geographic centroids, which were then assigned as optimal target destinations for a fleet of two autonomous UAVs (DroneFleet).

Once the spatial targets were established, both drones independently executed their navigation policies using the trained five-hidden-layer network (DQN-5) to converge onto the calculated centroids in the 3D space.

To visually capture the precise kinematic progression and energy utilization profiles during this collaborative deployment, Figure 4.10 tracks the concurrent trajectories of both assets. The upper plot illustrates the step-by-step physical distance remaining to the assigned cluster centroids, while the lower plot demonstrates the synchronous continuous growth of cumulative mechanical energy dissipation (in Joules) across the execution steps.

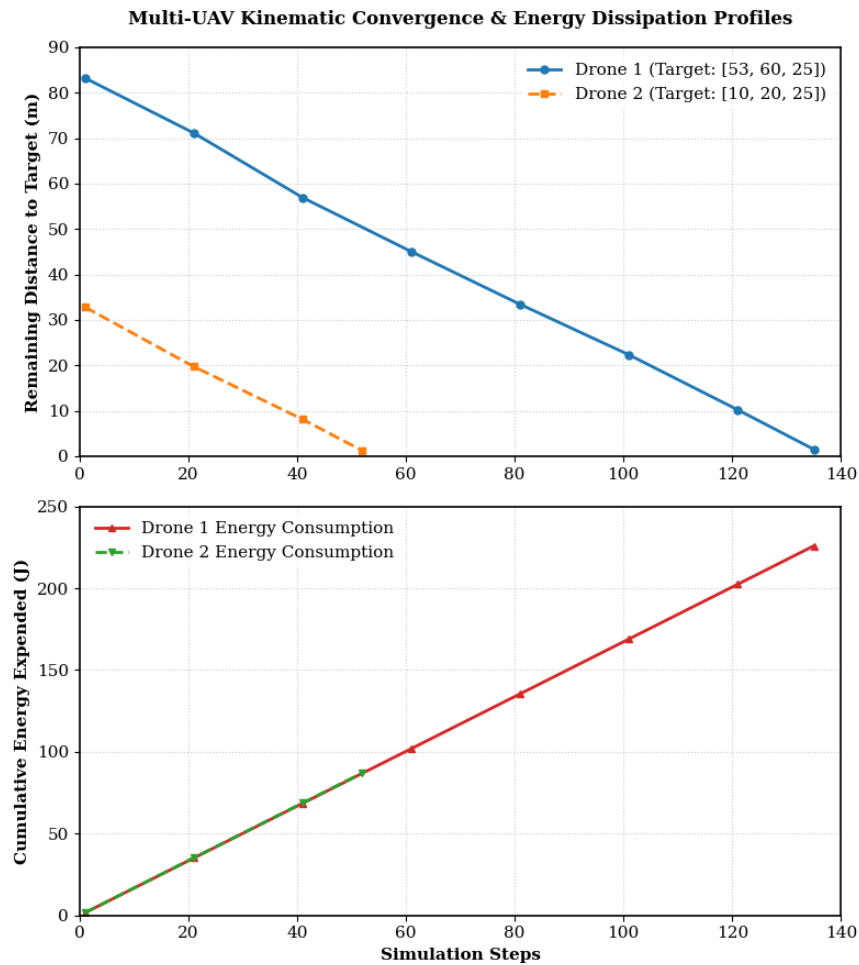


Figure 4-10: Multi-UAV Kinematic Convergence and Energy Dissipation Profiles Over Simulation Steps

• Kinematic Trajectory and Allocation Analysis

The operational metrics extracted from the deployment confirm that the K-Means target allocation successfully segregates the user grid into independent sectors, minimizing flight path overlap. Table 4.8 summarizes the macro-level performance profiles for both assets upon completing their missions

Table 4.8 Final Deployment Summary for K-Means Target Allocation and DQN-5 Navigation

Drone	Target Position	Status	Steps	Total Energy (j)
Drone 1	(53, 60, 25)	Arrived	135	225.905
Drone 2	(10, 20, 25)	Arrived	52	87.162

The real-time telemetry illustrated in the performance curves reveals distinct flight behaviors governed by the initial geometric offsets from the calculated cluster centers:

- **Drone 1** was assigned to cover a significantly wider spatial gap, starting with an initial geometric offset of 83.2 meters to reach its designated target at coordinate vector [53, 60, 25]. Driven by the DQN-5 network, it successfully completed its convergence mission in 135 steps, consuming a cumulative total of 225.905 Joules. Due to its rapid initial physical progress relative to the baseline hover power during step 1, its mathematical efficiency metric η exhibited an early numerical spike ($\eta = 50.173$), which gradually stabilized into a steady, monotonic decay as the asset actively raised its altitude to $Z = 25$ meters and trimmed its horizontal approach vectors.
- **Drone 2** directly benefited from a localized, higher-density user cluster, initiating its flight path much closer to its target coordinate [10, 20, 25] with a baseline distance of only 32.8 meters. Consequently, its embedded DQN-5 policy allowed it to achieve optimal 3D convergence rapidly in just 52 steps, requiring a minimized mechanical energy budget of 87.162 Joules.

This dual-asset validation confirms that the framework maintains high energy-awareness and navigational precision across varying cluster ranges. By offloading global spatial segmentation to the K-Means algorithm, the DQN-5 network can dedicate its parameters entirely to safe, localized 3D approach vectors without suffering from the exponential state-space inflation typical of multi-agent reinforcement learning setups.

4.5.13 Discussion

The experimental results presented in this chapter validate the proposed energy-aware Deep Reinforcement Learning framework for UAV placement optimization in FANET environments. The comparative study systematically confirms that deeper DQN architectures consistently outperform shallower configurations across all critical evaluation metrics. Specifically, the DQN-5 variant, equipped with five hidden layers, achieves a substantial 36.2% improvement in the energy-efficiency metric (η) and a 22.3% reduction in cumulative energy consumption relative to the baseline DQN-3 architecture. This performance gap demonstrates that the higher representational capacity of deeper neural networks is essential for accurately approximating complex, multi-dimensional Q-value surfaces where spatial navigation constraints and non-linear aerodynamic power costs are deeply coupled.

The analysis of DQN-5 over 12,000 training episodes demonstrates that the energy-aware reward shaping function effectively guides the agent toward energy-efficient path optimization. The learning process progresses through three main phases: random exploration, policy development, and convergence, consistent with reinforcement learning theory.

A critical factor in accelerating this learning curve is the integration of the K-Means Clustering algorithm prior to neural network training. By processing the geographic coordinates of ground users, the K-Means algorithm eliminates the typical vulnerabilities of fully random initial placement. Initiating the UAV deployment at the calculated cluster centroids (μ_j) reduces the initial geometric distance to the target nodes, which effectively mitigates the highly punitive boundary-hovering constraints and excessive battery depletion observed during early training

phases. Consequently, the agent bypasses long-range navigational errors and dedicates its computational capacity to fine-tuning its 3D position for optimal signal quality and dynamic coverage.

The deterministic inference testing ($\varepsilon = 0$) confirms robust spatial generalization, demonstrating a 100% target acquisition success rate across completely unseen user distributions. The fact that the energy-efficiency parameter (η) during deterministic inference is significantly higher than the training average indicates that the stochastic exploration noise ($\varepsilon = 0.05$) maintained during the training phase temporarily masks the true efficiency and path monotonicity of the learned policy. The resulting optimized 3D trajectories demonstrate mathematically sound navigation behavior, characterized by rapid altitude adjustments in the initial flight segments and minimal wasted mechanical movement as the drone approaches its final target coordinates.

Collectively, these findings validate the effectiveness of combining unsupervised geometric clustering (K-Means) for initial positioning with deep Q-networks (DQN-5) and energy-aware reward shaping for intelligent UAV deployment. This dual-optimization framework provides a highly stable framework for scaling up to multi-agent FANET scenarios, where additional operational constraints such as inter-UAV collision avoidance, backhaul communication links, and non-deterministic ground user mobility must be managed simultaneously.

4.6 Conclusion

This chapter evaluated the proposed DRL-based 3D UAV placement framework through extensive simulations. The results demonstrated that the UAV successfully learned efficient navigation strategies while reducing energy consumption and improving path optimization. Among the tested architectures, the 5-layer DQN achieved the best overall performance in terms of reward, energy efficiency, success rate, and training stability. These findings confirm the effectiveness of Deep Reinforcement Learning for autonomous and energy-efficient UAV deployment in wireless communication networks.

General Conclusion

General Conclusion

In this work, the use of Flying Ad Hoc Networks (FANETs) and Deep Reinforcement Learning (DRL) for intelligent UAV placement and energy-efficient wireless communications has been studied. The study initiated with a review on the fundamentals of ad hoc networks and their major categories, with a focus on the unique features, applications, and challenges of FANETs. Then it analyzed the role of UAVs in wireless communication systems, including deployment strategies, air-to-ground communication models, and energy constraints, which are the main factors affecting network performance.

Furthermore, in this work, the fundamentals of Reinforcement Learning and Deep Learning were studied and the ability of their combination, Deep Reinforcement Learning, to allow autonomous decision making in complex and dynamic environments was highlighted. In this chapter, we reviewed several DRL algorithms that can be applied to UAV placement optimization and showed the potential of the intelligent agents to learn the optimal deployment policies that can balance the coverage, the quality of communication and the energy consumption.

A simulation framework was developed in a 3D FANET environment where a UAV acts as an aerial base station to demonstrate these concepts. The proposed DRL-based approach was evaluated using different architectures of Deep Q-Network (DQN). The experimental results indicated that the deeper architecture converged faster, obtained higher rewards, better navigation efficiency, less energy consumption, and overall better network performance. In particular, DQN-5 achieved the best success rate and energy-efficiency measures, which validates the effectiveness of the DRL in optimizing the autonomous UAV placement.

Finally, the integration of FANET technology with Deep Reinforcement Learning offers a promising solution for future wireless networks by enabling adaptive, autonomous and energy-aware UAV deployment. In future works, the research can be extended to multi-UAV cooperative systems, advanced DRL algorithms such as PPO and SAC, real-world communication constraints and large-scale 6G aerial network applications, the optimized UAV positions obtained in this work can be integrated into a comprehensive FANET simulation environment to further evaluate the performance of the proposed deployment strategy under realistic operating conditions. Such simulations can investigate key communication metrics, including signal strength, signal-to-noise ratio (SNR), throughput, packet delivery ratio (PDR), end-to-end delay, latency, and network reliability.

REFERENCES

- [1] Corson, M. Scott, Joseph Macker, and Stephen G. Batsell. "Architectural considerations for mobile mesh networking." *Proceedings of MILCOM'96 IEEE Military Communications Conference*. Vol. 1. IEEE, 1996.
- [2] Frodigh, Magnus, Per Johansson, and Peter Larsson. "Wireless ad hoc networking: the art of networking without a network." *Ericsson review* 4.4 (2000): 249.
- [3] Stojmenovic, Ivan, and Xu Lin. "Power-aware localized routing in wireless networks." *IEEE transactions on parallel and distributed systems* 12.11 (2002): 1122-1133.
- [4] Yang, Hao, et al. "Security in mobile ad hoc networks: challenges and solutions." *IEEE wireless communications* 11.1 (2004): 38-47.
- [5] Chlamtac, Imrich, Marco Conti, and Jennifer J-N. Liu. "Mobile ad hoc networking: imperatives and challenges." *Ad hoc networks* 1.1 (2003): 13-64.
- [6] Karagiannis, Georgios, et al. "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions." *IEEE communications surveys & tutorials* 13.4 (2011): 584-616.
- [7] Baldessari, Roberto, et al. "Car-2-car communication consortium-manifesto." (2007).
- [8] Al-Zaidi, Rabab, et al. "Building novel VHF-based wireless sensor networks for the internet of marine things." *IEEE Sensors Journal* 18.5 (2018): 2131-2144.
- [9] Oubbati, Omar Sami, et al. "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives." *IEEE access* 7 (2019): 81057-81105.
- [10] Eissa, Tameem, Shukor Abd Razak, and MD Asri Ngadi. "Towards providing a new lightweight authentication and encryption scheme for MANET." *Wireless Networks* 17.4 (2011): 833-842.
- [11] Guo, Yuan, Hongying Tang, and Ronghua Qin. "A low complexity persistent reconnaissance algorithm for FANET." *Sensors* 22.23 (2022): 9526.
- [12] Beegum, T. Rehannara, et al. "Optimized routing of UAVs using bio-inspired algorithm in FANET: A systematic review." *IEEE access* 11 (2023): 15588-15622.
- [13] Albu-Salih, Alaa Taima, and Hayder Ayad Khudhair. "ASR-FANET: An adaptive SDN-based routing framework for FANET." *International Journal of Electrical & Computer Engineering* (2088-8708) 11.5 (2021).
- [14] E. T. Michailidis and A. G. Kanatas, "Three-Dimensional HAP-MIMO Channels: Modeling and Analysis," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 2-12, 2010.
- [15] Srivastava, Ashish, and Jay Prakash. "Future FANET with application and enabling techniques: Anatomization and sustainability issues." *Computer science review* 39 (2021): 100359.

- [16] C. Tipantuna, X. Hesselbach et al., "Energy-Aware Management in Multi-UAV Deployments: Modelling and Strategies," PMC/MDPI, 2020.
- [17] Survey on Energy Optimization Techniques in UAV-Based Cellular Networks, MDPI Drones, vol. 7, no. 3, p. 214, Mar. 2023.
- [18] Khan, Muhammad Fahad, et al. "Routing schemes in FANETs: A survey." Sensors 20.1 (2019): 38.
- [19] J. Tang et al., "Energy-Efficient UAV Trajectory Optimization with IoT Transmission Resource Allocation," IEEE Transactions on Vehicular Technology, 2022.
- [20] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments," Proc. IEEE GLOBECOM, 2014.
- [21] Friis, Harald T. "A note on a simple transmission formula." proc. IRE 34.5 (1946): 254-256.
- [22] Guillen-Perez, Antonio, and Maria-Dolores Cano. "Flying ad hoc networks: A new domain for network communications." Sensors 18.10 (2018): 3571.
- [23] Rezwan, Sifat, and Wooyeol Choi. "A survey on applications of reinforcement learning in flying ad-hoc networks." Electronics 10.4 (2021): 449.
- [24] NM Obaid Khan, Adnan Nadeem, Kamran Ahsan, "RPRP: Routing Protocol to ensure the reliability in health care system", published in textroad Journal of Basic and Applied Scientific Research Vol.4, No.1, pp 17-25, 2014.
- [25] Salam, Abdul, et al. "A novel QoS algorithm for health care applications of body area sensor networks." published in textroad Journal of Basic and Applied Scientific Research 4.1 (2014): 169-178.
- [26] Azfar, Saeed, et al. "Impact analysis and a detection method for misbehaving nodes in Mobile ad-hoc networks." Journal of Basic and Applied Scientific Research 4.8 (2019): 26-35.
- [27] W. Saad, M. Bennis, M. Mozaffari, and X. Lin, Wireless Communications and Networking for Unmanned Aerial Vehicles, Cambridge University Press, 2020.
- [28] Y. Zeng, I. Guvenc, R. Zhang, R. Deng, and P. Bhatt (Eds.), UAV Communications for 5G and Beyond, Wiley-IEEE Press, 2020.
- [29] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2334-2360, 2019.
- [30] K. F. Hayajneh, K. Bani-Hani, H. Shakhathreh, M. Anan, and A. Sawalmeh, "3D Deployment of UAV Base Station Assisting Ground Base Station," Wireless Communications and Mobile Computing, 2021.
- [31] Y. Zeng, J. Xu, and R. Zhang, "Energy Minimization for Wireless Communication with Rotary-Wing UAV," IEEE Transactions on Wireless Communications, vol. 18, no. 4, pp. 2329-2345, Apr. 2019.

- [32] 3GPP, "Study on Enhanced LTE Support for Aerial Vehicles," 3GPP TR 36.777, Release 15, 2018.
- [33] K. Namuduri, S. Chaumette, J. H. Kim, and J. P. G. Sterbenz (Eds.), UAV Networks and Communications, Cambridge University Press, 2017.
- [34] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," IEEE Wireless Communications Letters, vol. 3, no. 6, pp. 569-572, Dec. 2014.
- [35] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D Placement of an Unmanned Aerial Vehicle Base Station for Energy-Efficient Maximal Coverage," IEEE Wireless Communications Letters, vol. 6, no. 4, pp. 434-437, Aug. 2017.
- [36] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient Deployment of Multiple UAVs for Optimal Wireless Coverage," IEEE Communications Letters, vol. 20, no. 8, pp. 1647-1650, Aug. 2016.
- [37] C. Zhang et al., "Traffic-Aware Multi-UAV 3D Deployment Optimization," arXiv:2506.13287, 2025.
- [38] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenburger, "A Survey of Air-to-Ground Propagation Channel Modeling for UAVs," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2361-2391, 2019.
- [39] Experimental study on low-altitude UAV-to-ground propagation, ScienceDirect, Computer Networks, 2023.
- [41] Q. Wu et al., "Modeling and Simulation of Dual-Battery FANET Using MADRL for Energy Optimization," Scientific Reports, 2025.
- [42] J. Niu et al., "Energy Efficiency Maximization for UAV-Assisted Emergency Communication Networks," Wireless Communications and Mobile Computing, 2021.
- [43] Kaelbling, L. P., Littman, M. L., Moore, A. W., 1996. Reinforcement learning: A survey. Journal of artificial intelligence research 4, 237–285.
- [44] Kober, J., Bagnell, J. A., Peters, J., 2013. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32 (11), 1238–1274.
- [45] Mousavi, S. S., Ghazanfari, B., Mozayani, N., Jahed-Motlagh, M. R., 2014. Automatic abstraction controller in reinforcement learning agent via automata. Applied Soft Computing 25, 118–128.
- [46] Sutton, R. S., Barto, A. G., 1998. Introduction to reinforcement learning. Vol. 135. MIT press Cambridge
- [47] Vien, N. A., Ertel, W., Dang, V.-H., Chung, T., 2013. Monte-carlo tree search for bayesian reinforcement learning. Applied intelligence 39 (2), 345–353.

- [48] Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35 (8), 1798–1828.
- [49] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.
- [50] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
- [51] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. ICLR*, 2016.
- [52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [53] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. NeurIPS*, 2017, pp. 6379–6390.
- [54] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.
- [55] LI, Yuxi. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.