الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي والبحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة قاصدي مرباح - ورقلة
KASDI MERBAH UNIVERSITY - OUARGLA




كلية التكنولوجيات الحديثة للإعلام والاتصال
FACULTY OF NEW INFORMATION AND COMMUNICATION TECHNOLOGIES
قسم الإلكترونيات والاتصالات
DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS

**Specialization:** Telecommunications Systems

**Submitted By:** Mehaya Abdeldjalil
Djendouci Abdelghani

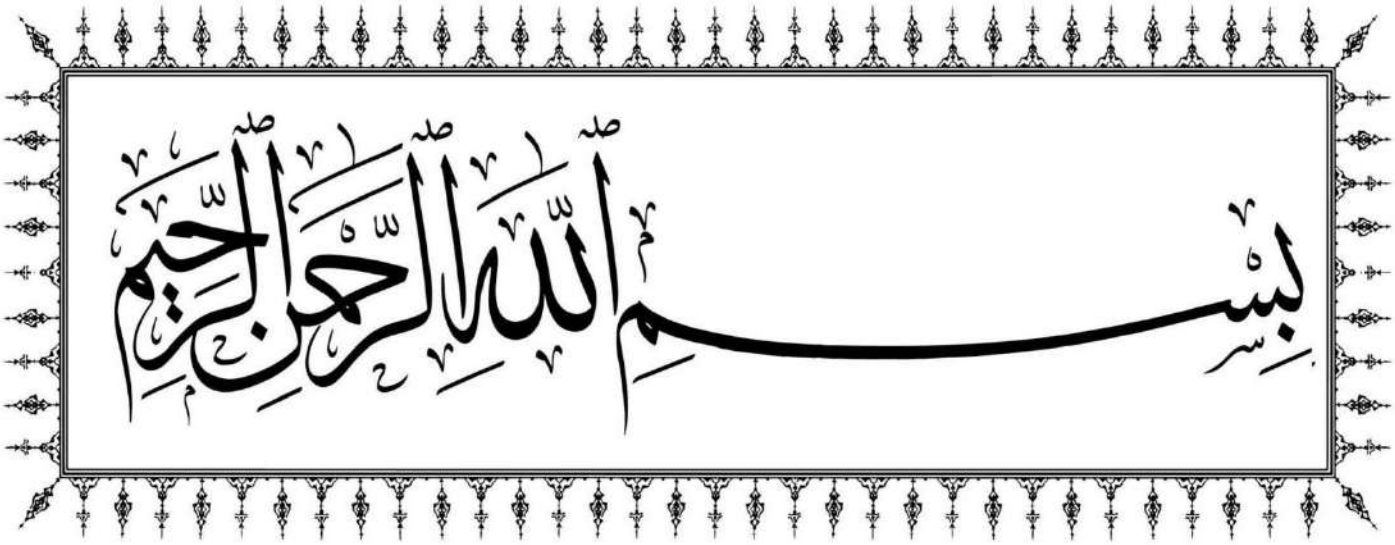
**Theme**

# Intelligent Routing for Walker Delta Satellite Network: A DRL Framework for Latency QoS Optimization

Jury members:

| | | |
|---|---|---|
| Pr.AUONALLAH Naceur | University of Ouargla | President |
| Dr.HAMMOUCHI Fatah | University of Ouargla | Examiner |
| Dr.CHENINA Hachemi | University of Ouargla | Advisor |

**Academic Year:** 2025/2026


بسم الله الرحمن الرحيم

# DEDICATION

**I dedicate this work:**

To my beloved parents, whose unconditional love, sacrifices, encouragement, and support have been the foundation of all my achievements. Their patience and belief in me have inspired me throughout every stage of my academic journey.

To my family, for their constant motivation, understanding, and support during the preparation of this thesis.

To my friends and colleagues, whose encouragement and companionship made this journey more enjoyable and rewarding.

Finally, I dedicate this work to everyone who contributed, directly or indirectly, to my education and personal development.

Thank you all for being part of this achievement.

***The autors***

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to Allah for granting me the strength, patience, and perseverance necessary to complete this work.

I would like to extend my deepest appreciation to my supervisor, Dr. CHENINA Hachemi , for his/her valuable guidance, continuous support, constructive suggestions, and insightful comments throughout the development of this thesis. His expertise and encouragement greatly contributed to the successful completion of this work.

I would also like to thank the members of the jury for accepting to evaluate this thesis and for the time and effort they dedicate to reviewing this work.

My sincere thanks are extended to all the professors and staff members of the Department of DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS for the knowledge and assistance they provided during my academic studies.

I am especially grateful to my parents and family for their unwavering support, patience, and encouragement. Their confidence in my abilities has always motivated me to pursue my goals.

Finally, I would like to thank my friends and colleagues who have supported me throughout this journey and contributed to creating a positive and inspiring academic environment.

To all those who contributed, directly or indirectly, to the realization of this work, I express my heartfelt appreciation.



**الملخص:**

تهدف هذه الدراسة إلى اقتراح إطار ذكي للتوجيه داخل كوكبات الأقمار الصناعية من نوع Walker Delta بالاعتماد على التعلم المعزز العميق (Deep Reinforcement Learning - DRL) من أجل تحسين مؤشرات جودة الخدمة المرتبطة بزمن التأخير. تم في البداية تقديم المفاهيم الأساسية المتعلقة بشبكاتLEO ، والروابط بين الأقمار الصناعية (ISLs)، وبنيةWalker Delta ، إضافة إلى استعراض مشكلات التوجيه ومتطلبات جودة الخدمة في البيئات الفضائية الديناميكية.

تمت صياغة مشكلة التوجيه على شكل عملية قرار ماركوفية(MDP) ، كما تم تصميم وكيل تعلم معزز عميق من نوع (DDQN) -Q Double Deep Network لاتخاذ قرارات التوجيه بصورة تكيفية اعتماداً على الحالة الآنية للشبكة. وقد أُنشئت بيئة محاكاة تمثل كوكبة مكونة من 100 قمر صناعي موزعة وفق نموذجWalker Delta ، مع تضمين مؤشرات الأداء الأساسية مثل زمن التأخير، وعدد القفزات، ونسبة تسليم الحزم، ومعدل أخطاء البت.

أظهرت نتائج المحاكاة أن النهج المقترح القائم على DDQN يحقق أداءً أفضل مقارنةً بطرق التوجيه التقليدية، حيث ساهم في تقليل زمن التأخير وتحسين موثوقية الاتصالات وزيادة نسبة تسليم الحزم، مع قدرة عالية على التكيف مع التغيرات المستمرة في طوبولوجيا الشبكة. وتؤكد النتائج أن تقنيات التعلم المعزز العميق تمثل حلًا واعداً لتطوير أنظمة توجيه ذكية وفعالة في شبكات الأقمار الصناعية المستقبلية.

**الكلمات المفتاحية:** الأقمار الصناعية ذات المدار الأرضي المنخفض (LEO)، كوكبة Walker Delta، التوجيه الذكي، التعلم المعزز العميق (DRL)، الشبكات العصبية العميقة.

**Abstract :**

This thesis proposes an intelligent routing framework for Walker Delta satellite constellations based on Deep Reinforcement Learning (DRL) to optimize latency-related QoS metrics. First, the fundamental concepts of LEO satellite networks, inter-satellite links (ISLs), Walker Delta architecture, routing challenges, and QoS requirements are presented. The routing problem is then formulated as a Markov Decision Process (MDP), enabling adaptive decision-making in a dynamic network environment.

A Double Deep Q-Network (DDQN) agent is designed to learn optimal routing policies through interaction with the satellite network. A simulation environment representing a 100-satellite Walker Delta constellation is developed, incorporating key performance metrics such as end-to-end latency, hop count, packet delivery ratio, and bit error rate.

Simulation results demonstrate that the proposed DDQN-based routing framework outperforms conventional routing approaches by reducing communication latency, improving routing reliability, and increasing packet delivery performance. Furthermore, the learning agent successfully adapts to dynamic topology variations, confirming the effectiveness of Deep Reinforcement Learning for intelligent routing in large-scale LEO satellite constellations.

**Keywords:** Low Earth Orbit (LEO), Walker Delta Constellation, Intelligent Routing, Deep Reinforcement Learning (DRL), Deep Neural Networks (DNN).

# LIST OF ABBREVIATIONS

**AI:** Artificial Intelligence

**BER:** Bit Error Rate

**DDQN:** Double Deep Q-Network

**DNN:** Deep Neural Network

**DQN:** Deep Q-Network

**DRL:** Deep Reinforcement Learning

**GEO:** Geostationary Earth Orbit

**GNN:** Graph Neural Network

**GPS:** Global Positioning System

**IoT:** Internet of Things

**ISL:** Inter-Satellite Link

**LEO:** Low Earth Orbit

**MADRL**: Multi-Agent Deep Reinforcement Learning

**MARL**: Multi-Agent Reinforcement Learning

**MDP:** Markov Decision Process

**MEO:** Medium Earth Orbit

**ML:** Machine Learning

**PDR:** Packet Delivery Ratio

**PPO:** Proximal Policy Optimization

**QoS:** Quality of Service

**Q-value:** Action-Value Function

**RAAN:** Right Ascension of the Ascending Node

**RF:** Radio Frequency

**RL:** Reinforcement Learning

**RTMDP:** Real-Time Markov Decision Process

**SDN:** Software-Defined Networking

**SDSN:** Software-Defined Satellite Network

**SNR:** Signal-to-Noise Ratio

**TTL:** Time-To-Live

**UAV:** Unmanned Aerial Vehicle

**VLEO:** Very Low Earth Orbit

# TABLE OF CONTENTS

DEDICATION ..... I

ACKNOWLEDGEMENTS ..... III

ABSTRACT ..... IV

LIST OF ABBREVIATIONS ..... V

TABLE OF CONTENTS ..... VII

LIST OF FIGURES ..... X

LIST OF TABLES ..... XI

GENERAL INTRODUCTION ..... 02

**Chapter I : INTRODUCTION TO LEO SATELLITE NETWORKS AND WALKER DELTA ARCHITECTURE**

Introduction ..... 06

I.1. Definition of Satellites ..... 06

I.2. Satellite Orbits ..... 07

I.3. Satellite Constellations ..... 08

I.4. Types of Satellite Constellations According to Orbit ..... 09

I.4.2. Medium Earth Orbit (MEO) Constellations ..... 10

I.4.3. Geostationary Earth Orbit (GEO) Constellations ..... 10

I.5. Walker delta constellation architecture ..... 10

I.6. Inter-satellite links (ISLs) ..... 12

I.7 Satellite Constellation Routing ..... 14

I.7.1 Routing in LEO Satellite Networks ..... 15

I.7.2 Routing in MEO Satellite Networks ..... 15

I.7.3 Routing in GEO Satellite Networks ..... 16

I.7.4 Routing in Multi-Layer Satellite Networks (LEO/MEO/GEO) ..... 16

I.7.5 Routing Problems in LEO Satellite Networks ..... 17

I.8. Satellite constellation routing ..... 17

I.8.1. Multi-Layer satellite networks ..... 18

I.8.2. Routing problem in LEO satellite networks ..... 18

I.9. Quality of service (QOS) requirements ..... 20

**Chapter II: FUNDAMENTALS OF DEEP REINFORCEMENT LEARNING (DRL) IN NETWORKING**

Introduction ..... 24

II.1. Reinforcement learning (RL) Concepts ..... 24

II.1.1. What is Deep Reinforcement Learning (DRL) ..... 25

II.1.2. How DRL Works (End-to-End Loop)? ..... 29

II.1.3. Why "Deep"? ..... 30

II.1.4. How DRL Maps to QoS-Aware Routing (Example)? ..... 31

II.1.5. When to Use Which Algorithm? ..... 32

II.1.6. From RL to Deep Reinforcement Learning ..... 32

II.2. Deep Neural Networks in Routing ..... 33

II.3. Modeling routing as a markov decision process (MDP) ..... 35

II.4. Reward function design ..... 37

**Chapter III: SIMULATION, IMPLEMENTATION, AND RESULTS DISCUSSION**

Introduction ..... 41

III.1 Orbital Design and Analysis of a 100-Satellite Constellation Using the Walker Delta Method ..... 42

III.1.1 Mathematical Formulation of the Constellation ..... 42

III.2 Numerical Application: Configuration of a 100-Satellite Constellation ..... 43

III.2.1 System Parameters and Assumptions ..... 44

III.2.2 Step-by-Step Computations and Results ..... 44

III.3. Kinematic-Driven DRL Environment for Multi-Objective Satellite Routing ..... 45

III.3.1 Dynamic Network Graph Topology ..... 46

III.3.2 Mathematical Formulation of Routing Optimization Metrics ..... 46

III.3.3 MDP Mapping for Deep Reinforcement Learning ..... 48

III.4. Detailed DDQN Agent Architecture and Numerical Parameterization ..... 49

III.4.1 State Space Dimensionality and Input Layer Design ..... 49

III.4.2 Action Space and Action Masking Mechanism ..... 50

III.4.3 Hidden Layers and Network Hyperparameters ..... 51

III.4.4 Step-by-Step Optimization and Value Calculations ..... 52

III.5. Convergence Rate and Learning Stability Analysis ..... 53

III.6. Average Hop Count Optimization Analysis ..... 56

III.7. Packet Delivery Ratio (PDR) Evaluation ..... 59

III.8. Geographical Simulation Setup and Environmental Modeling ..... 61

III.9. Integrated Performance Analysis for Algiers-to-Tokyo Scenario ..... 62

Conclusion ..... 64

General conclusion ..... 66

Références ..... 69

# LIST OF FIGURES

**Figure I.1:** Satellite Orbits [17] **07**

**Figure I.2:** GEO vs. MEO vs. LEO Satellite Constellations [21] **09**

**Figure I.3:** An example of Walker-delta constellation consists of O = 5 orbits, each having N o = 8 satellites, orchestrated by H = 2 HAPs. Gray cones depict LEO satellites' coverage [26]. **11**

**Figure I.4:** ISL structure [33]. **13**

**Figure I.5:** Illustration of different types of ISL: intraplane (blue) and interplane (green) [36] **13**

**Figure I.6:** Simplified schematic diagram of Low-Earth Orbit (LEO) satellite network [46]. **19**

**Figure II.1:** Overview model of Reinforcement Learning [64]. **25**

**Figure II.2:** Deep reinforcement learning [67] **33**

**Figure II.3:** A basic structure of the deep neural network [69] **34**

**Figure II.4:** Structure of a prototypical reinforcement learning project emphasizing the design of the reward function by the engineer based on the specified task [91]. **37**

**Figure III.1:** Three-Dimensional Spatial Distribution of the $53^{\circ}$: $100/10/2$ Walker Delta Satellite Constellation **45**

**Figure III.2:** Convergence performance and cumulative reward of the proposed DDQN agent compared to Standard Q-Learning and Random Walk routing over 1000 training episodes. **53**

**Figure III.3:** Evolution of the end-to-end propagation latency over 1000 training episodes, highlighting the learning progression of the DDQN agent compared to static and baseline routing algorithms. **55**

**Figure III.4:** Theoretical convergence trajectory of the average hop count over 1000 training episodes, demonstrating the direct learning progression of the DDQN agent towards optimal satellite node utilization **56**

**Figure III.5:** Convergence of the end-to-end Bit Error Rate (BER) on a logarithmic scale, highlighting the DDQN agent's capacity to optimize signal integrity compared to topology-blind routing algorithms **57**

**Figure III.6:** Evolution of the Packet Delivery Ratio (PDR) across training episodes, demonstrating the DDQN agent's superior capability in ensuring reliable end-to-end data transmission by mitigating packet loss. **58**

**Figure III.7:** Comparison of the average route computation time (inference execution time) per packet. The DDQN agent demonstrates rapid decision-making capabilities highly suitable for real-time routing in dynamic LEO satellite networks. **60**

**Figure III.8:** Integrated performance dashboard of the Algiers–Tokyo LEO satellite routing scenario, comparing DDQN, Dijkstra, and Standard Q-Learning in terms of latency, hop count, BER, PDR, and throughput over 1000 training episodes. **61**

# LIST OF TABLES


| | |
|---|---|
| **Table III.1:** proposed internal architecture and numerical hyperparameters. | **51** |

# GENERAL INTRODUCTION

## GENERAL INTRODUCTION :

The rapid development of communication technologies has significantly increased the demand for global, reliable, and low-latency connectivity. In this context, satellite communication systems have become an essential component of modern telecommunication infrastructures, enabling data transmission across large geographical areas where terrestrial networks may be limited or unavailable. With the continuous evolution of satellite technologies, communication networks are becoming more intelligent, flexible, and capable of supporting emerging applications such as broadband internet, Internet of Things (IoT), remote sensing, and real-time services [1].

Among the different satellite systems, Low Earth Orbit (LEO) satellite constellations have attracted considerable attention due to their reduced propagation delay and improved communication performance compared to traditional Geostationary Earth Orbit (GEO) systems [2]. Modern LEO constellations consist of a large number of interconnected satellites distributed across multiple orbital planes to ensure continuous and near-global coverage. Architectures such as the Walker Delta constellation provide an organized and scalable deployment model that facilitates network management and connectivity [3].

The integration of Inter-Satellite Links (ISLs) has transformed satellite constellations into dynamic space-based networks capable of routing data directly between satellites. This evolution improves communication efficiency and reduces dependency on ground infrastructure. However, the highly dynamic topology of LEO networks introduces significant challenges, particularly in routing and Quality of Service (QoS) management. Frequent topology changes, variable traffic distribution, and strict latency requirements make traditional routing approaches insufficient for achieving optimal performance [4].

To address these limitations, intelligent routing techniques based on Artificial Intelligence (AI) and Machine Learning (ML) have recently emerged as promising solutions. In particular, Deep Reinforcement Learning (DRL) combines reinforcement learning with deep neural networks to enable adaptive and autonomous decision-making in complex environments [5]. DRL-based routing approaches are capable of learning optimal routing strategies through interaction with the network environment while continuously adapting to topology and traffic variations [6].

The objective of this work is to propose an intelligent routing framework for Walker Delta satellite networks using Deep Reinforcement Learning in order to optimize latency-related QoS metrics. The proposed approach aims to improve routing efficiency and adaptability in highly dynamic LEO environments.

Several studies have investigated intelligent routing and optimization techniques in Low Earth Orbit (LEO) satellite networks. Guoyi Zhang, Kai Wang, Changqing Lai, Chong Wang, Hongyan Xu, and Hao Qi proposed a Deep Reinforcement Learning (DRL)-based routing optimization framework for Software-Defined Satellite Networks (SDSN) [7]. Their approach combines Software-Defined Networking (SDN) with a Deep Q-Network (DQN) algorithm in order to dynamically optimize routing decisions according to real-time network conditions such as congestion, latency, and packet loss. The proposed architecture improves routing efficiency and network reliability, particularly during sudden link outages.

Zhe Niu, Hui Yang, Qiuyan Yao, Bingda Wu, Sentian Yin, and Shikui Shen introduced a reliable low-latency routing framework for Very Low Earth Orbit (VLEO) optical satellite networks using a multiagent reinforcement learning approach [8]. Their work focuses on reducing latency and improving reliability in optical inter-satellite communication systems under dynamic traffic conditions and varying network scales.

Hailong Liao, Xian Zhang, Jiaen Zhou, and Xuehua Li proposed an enhanced Multi-Agent Deep Reinforcement Learning (MADRL) method for real-time routing in LEO satellite networks [9]. Their approach models the routing problem using a Real-Time Markov Decision Process (RTMDP) and incorporates a self-attention mechanism to improve the understanding of network dynamics. The obtained results demonstrate improvements in throughput and end-to-end latency compared with conventional routing methods.

Shuyang Li, Qiang Wu, Ran Wang, Long Chen, and Hongke Zhang developed a DRL-based multipath differential routing and traffic scheduling framework for ultra-dense LEO satellite networks [10]. Their method combines Deep Reinforcement Learning with a Stackelberg game model to jointly optimize traffic scheduling, energy consumption, and transmission delay. Simulation results show better network throughput and reduced transmission time compared to baseline approaches.

Fei Zheng, Juzhong Wei, Shichao Li, Peng Yu, Shaoyong Guo, and Jilong Zhao proposed a dual-layer DRL routing approach for integrated UAV and satellite Internet of Things (IoT) networks [11]. Their framework divides the routing process into UAV-layer routing and LEO satellite-layer routing while applying DRL techniques to optimize transmission efficiency and reduce communication delay. The proposed solution achieves better convergence performance and lower overall delay.

Senbai Zhang, Aijun Liu, Chen Han, Xin Xu, Xiaohu Liang, and Kang an introduced the GRLR routing algorithm for mega LEO satellite constellations [12]. Their approach integrates Graph Neural Networks (GNNs) with Reinforcement Learning to improve distributed routing decisions in Walker-Delta constellations. The proposed method demonstrates rapid convergence and strong adaptability to dynamic network conditions while minimizing transmission delay.

Although these studies demonstrated the effectiveness of DRL techniques in satellite routing optimization, several limitations remain. Some approaches mainly focus on SDN architectures, optical communication systems, or integrated UAV-satellite environments, while others prioritize throughput and energy efficiency rather than latency-oriented Quality of Service (QoS) optimization in Walker Delta constellations. Therefore, this work proposes an intelligent DRL-based routing framework dedicated to Walker Delta satellite networks with particular emphasis on latency QoS optimization under highly dynamic network conditions.

This thesis is organized into three chapters. Chapter I presents the fundamental concepts related to LEO satellite networks, Walker Delta constellation architecture, inter-satellite links, routing challenges, and QoS requirements. Chapter II introduces the fundamentals of Deep Reinforcement Learning and its application in networking environments,describes the proposed DRL-based routing framework and its main components. Finally, Chapter III presents the simulation environment, obtained results, and performance analysis.

# CHAPTER I:

## INTRODUCTION TO LEO SATELLITE NETWORKS AND WALKER DELTA ARCHITECTURE

# CHAPTER I: INTRODUCTION TO LEO SATELLITE NETWORKS AND WALKER DELTA ARCHITECTURE

## Introduction:

The rapid evolution of global communication systems has led to increasing interest in Low Earth Orbit (LEO) satellite networks as a viable solution for low-latency and wide-area connectivity. Their deployment in large-scale constellations enables continuous coverage and supports a variety of modern applications.

Designing such networks requires a clear understanding of their architectural structure and operational constraints. In particular, constellation design, inter-satellite connectivity, and routing mechanisms play a central role in determining overall network performance.

This chapter introduces the fundamental concepts underlying LEO satellite networks. It presents the Walker Delta constellation architecture, explains the role of inter-satellite links, discusses the routing problem in dynamic environments, and outlines the main Quality of Service (QoS) requirements.

## I.1. Definition of Satellites:

A satellite is an object that revolves around another larger body in space under the influence of gravity. Satellites can be classified into two main categories: natural satellites, such as the Moon orbiting the Earth, and artificial satellites, which are human-made objects launched into space for specific purposes [13].

Artificial satellites are designed to perform various functions depending on their mission objectives. These functions include communication, navigation, weather monitoring, Earth observation, scientific research, and military operations [14]. Communication satellites, in particular, play a fundamental role in modern telecommunication systems by enabling the transmission of voice, video, and data signals across large geographical regions.

A communication satellite operates as a relay station in space. It receives signals transmitted from a ground station, processes or amplifies them, and retransmits them toward another ground station or user terminal. This capability allows communication over long distances without relying entirely on terrestrial infrastructure [15].

## I.2. Satellite Orbits:

A satellite orbit is the curved path followed by a satellite as it revolves around the Earth under the influence of gravitational force. The characteristics of an orbit, such as altitude, inclination, and orbital period, determine the satellite's coverage area, communication delay, and operational performance [16].

Satellite orbits are generally classified according to their altitude above the Earth's surface. The main types are Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO).

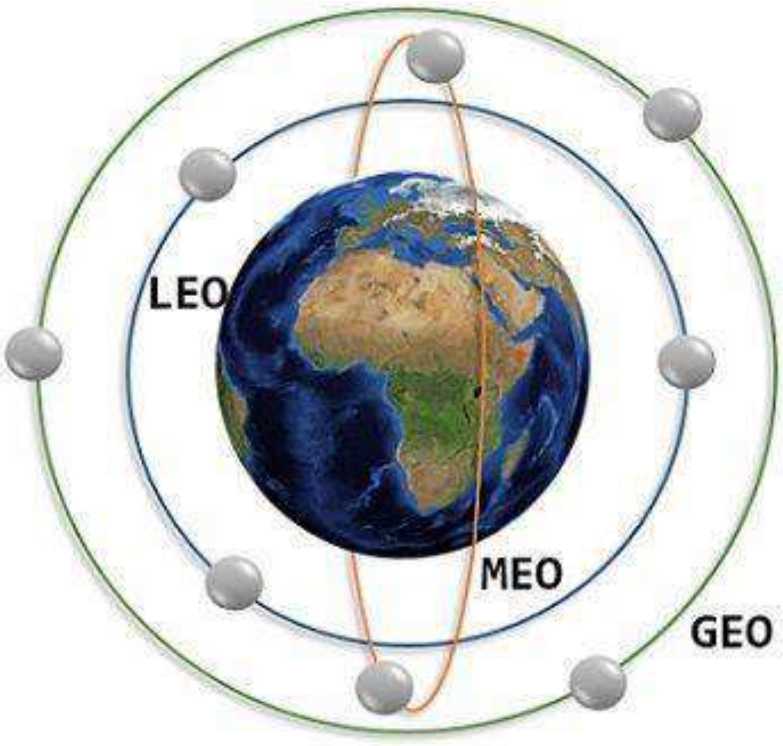



**Figure I.1:** Satellite Orbits [17].

- **Low Earth Orbit (LEO):**

LEO satellites operate at altitudes ranging approximately from 160 km to 2000 km above the Earth. Due to their proximity to the Earth, they provide low propagation delay, high data rates, and reduced latency [18]. However, their coverage area is limited, requiring the deployment of large satellite constellations to ensure continuous global coverage.

- **Medium Earth Orbit (MEO):**

MEO satellites are typically positioned between 2000 km and 35,786 km. They provide larger coverage areas than LEO satellites while maintaining lower latency than GEO systems. MEO orbits are commonly used in navigation systems such as GPS [18].

- **Geostationary Earth Orbit (GEO):**

GEO satellites are located at an altitude of approximately 35,786 km above the equator. At this altitude, the satellite rotates at the same angular velocity as the Earth, appearing stationary relative to a ground observer. GEO satellites provide wide coverage areas and are widely used for television broadcasting and long-distance communications. However, they experience higher latency due to the large propagation distance [15].

The selection of an orbital configuration depends on the application requirements, including coverage, latency, bandwidth, and system cost. In recent years, LEO constellations have gained significant attention because of their suitability for low-latency communication services and next-generation satellite networks [19].

Most satellites consist of several essential components, including power systems, antennas, transponders, onboard processors, propulsion systems, and control units. These components work together to ensure stable operation, communication capability, and orbital control throughout the satellite mission lifetime.

## I.3. Satellite Constellations:

A satellite constellation is a group of artificial satellites operating together in a coordinated manner to provide continuous communication or coverage over a specific region or across the entire Earth. Unlike single-satellite systems, constellations consist of multiple satellites distributed across different orbital planes in order to ensure uninterrupted service, improve reliability, and enhance network performance [20].

Satellite constellations are designed to overcome the limitations of individual satellites, particularly in terms of coverage area, latency, and communication continuity. By coordinating multiple satellites, the system can maintain permanent connectivity even when satellites move along their orbital paths.

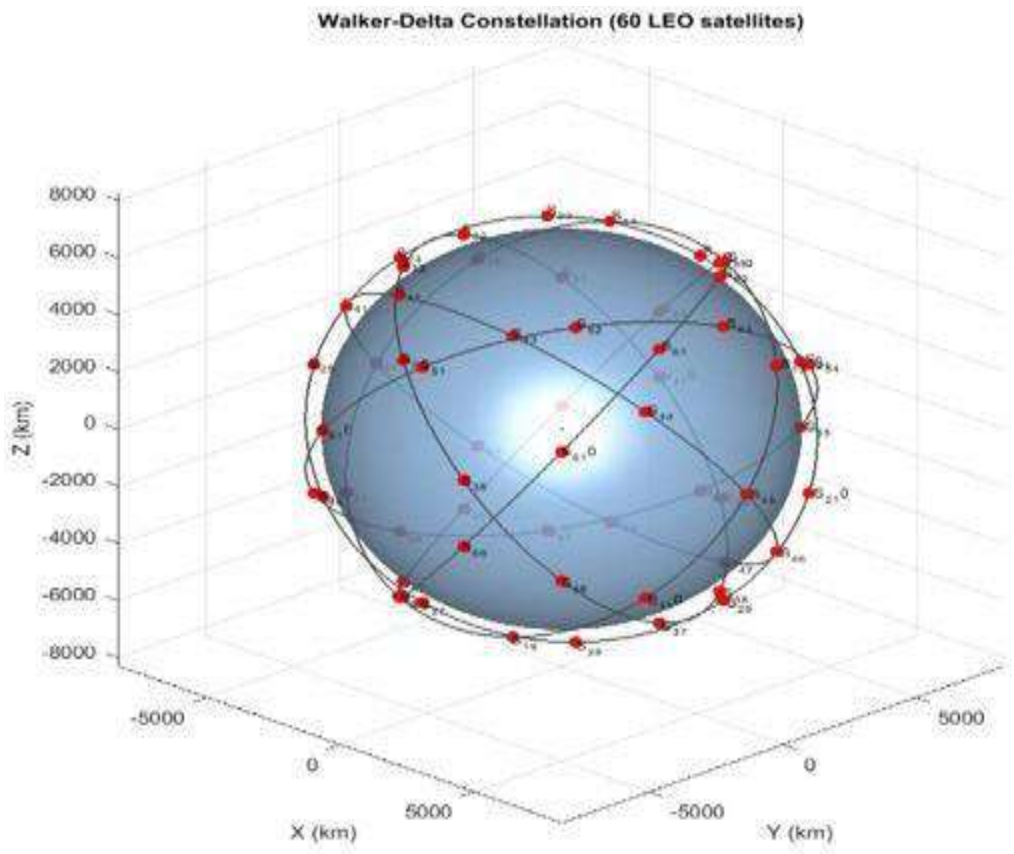



**Figure I.2:** GEO vs. MEO vs. LEO Satellite Constellations [21].

Modern satellite constellations are widely used in communication systems, navigation, Earth observation, and broadband internet services. Recent advances in satellite technology have led to the development of large-scale constellations composed of hundreds or thousands of satellites operating cooperatively in space [22].

## I.4. Types of Satellite Constellations According to Orbit:

Satellite constellations can be classified according to the orbital altitude of the satellites. The main categories are Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO) constellations [23]:

### I.4.1. Low Earth Orbit (LEO) Constellations:

LEO constellations operate at altitudes ranging from approximately 160 km to 2000 km above the Earth's surface. These constellations provide low propagation delay and high communication performance due to their proximity to the Earth. However, because each satellite

covers a relatively small area, a large number of satellites is required to ensure continuous global coverage.

LEO constellations are widely used for broadband internet services and low-latency communications. Examples include Starlink and OneWeb.

### I.4.2. Medium Earth Orbit (MEO) Constellations:

MEO constellations operate at intermediate altitudes between LEO and GEO systems. They provide larger coverage areas than LEO satellites while maintaining lower latency compared to GEO systems. MEO constellations generally require fewer satellites to achieve global coverage.

These constellations are commonly used in navigation and positioning systems such as the Global Positioning System (GPS).

### I.4.3. Geostationary Earth Orbit (GEO) Constellations

GEO constellations are positioned at an altitude of approximately 35,786 km above the equator. Satellites in this orbit rotate at the same angular velocity as the Earth, allowing them to appear stationary relative to ground observers.

GEO systems provide very large coverage areas, making them suitable for television broadcasting, weather monitoring, and long-distance communication services. However, they experience higher latency due to the long propagation distance between the Earth and the satellite.

## I.5. Walker delta constellation architecture:

The Walker Delta constellation is one of the most widely used architectures for designing Low Earth Orbit (LEO) satellite networks, particularly in large-scale communication systems. It was originally introduced as a structured method to distribute satellites uniformly around the Earth in order to achieve continuous and predictable coverage [24].

This architecture is defined by a set of orbital parameters that determine how satellites are arranged in space. A typical Walker Delta constellation is described using three main parameters [24, 25]:

- T: total number of satellites
- P: number of orbital planes

- F: phasing factor between satellites in adjacent planes

This configuration is commonly denoted as Walker Delta (T/P/F), offering a compact representation of the constellation geometry.

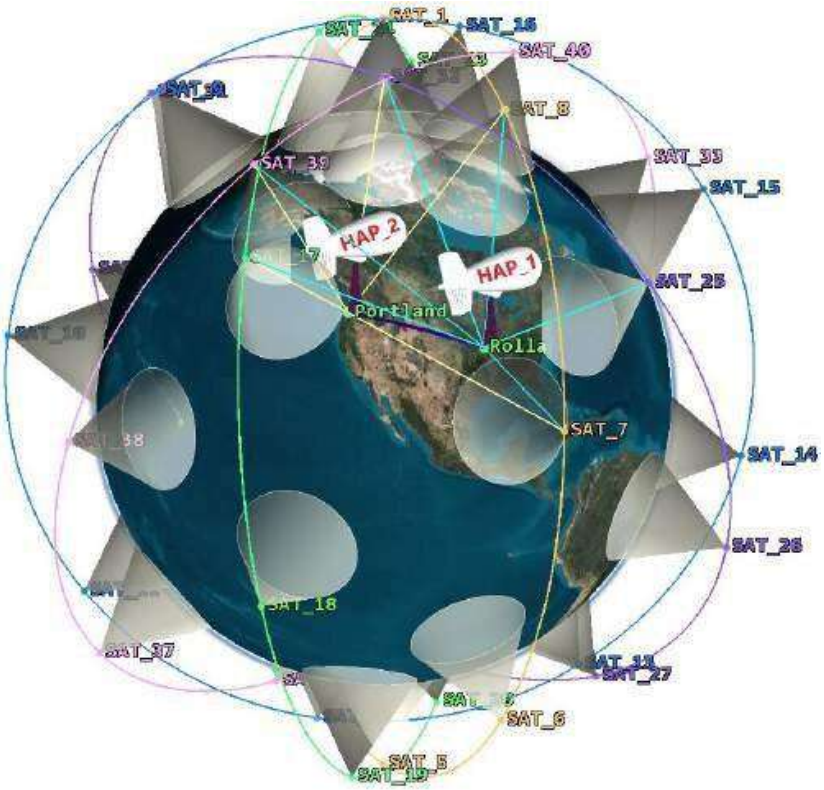



**Figure I.3:** An example of Walker-delta constellation consists of O = 5 orbits, each having N o = 8 satellites, orchestrated by H = 2 HAPs. Gray cones depict LEO satellites' coverage [26].

In this architecture, satellites are evenly distributed across multiple orbital planes, which are themselves uniformly spaced in terms of the right ascension of the ascending node (RAAN). Within each plane, satellites are equally spaced along the orbit [15, 27]. The phasing parameter (F) introduces a controlled angular offset between satellites in neighboring planes, ensuring that satellites do not align vertically and instead provide more uniform spatial coverage.

One of the main advantages of the Walker Delta architecture is its geometric regularity, which simplifies constellation design, analysis, and implementation. The predictable structure

allows efficient modeling of satellite positions, coverage areas, and network topology, making it particularly suitable for large constellations composed of hundreds or thousands of satellites [24].

From a coverage perspective, Walker Delta constellations are especially effective at providing uniform service in low and mid-latitude regions. However, due to the lack of synchronization between satellites in different orbital planes, coverage performance may degrade at higher latitudes, where gaps or inefficiencies can occur [28].

In terms of networking, the structured arrangement of satellites facilitates the establishment of inter-satellite links (ISLs). These links enable communication between neighboring satellites within the same orbital plane and across adjacent planes, forming a dynamic space-based network. This mesh-like topology supports efficient data routing and reduces dependency on ground stations [29].

Despite its advantages, the Walker Delta architecture is not universally optimal. Recent studies suggest that alternative constellation designs may offer improved performance in specific scenarios, particularly in terms of coverage optimization and routing efficiency. Nevertheless, the Walker Delta model remains a fundamental reference in the design and analysis of LEO satellite constellations [30, 31].

### I.6. Inter-satellite links (ISLs):

Inter-Satellite Links (ISLs) are a core enabler of modern Low Earth Orbit (LEO) satellite networks, allowing satellites to communicate directly with one another without relying exclusively on ground stations. This represents a shift from traditional “bent-pipe” architectures to fully interconnected space-based networks capable of in-orbit data routing [29].

Through ISLs, data can be forwarded across multiple satellites before reaching its final ground destination, effectively transforming the constellation into a distributed mesh network. In this architecture, each satellite functions not only as a relay node but also as a routing entity, enabling flexible and efficient end-to-end communication [32].

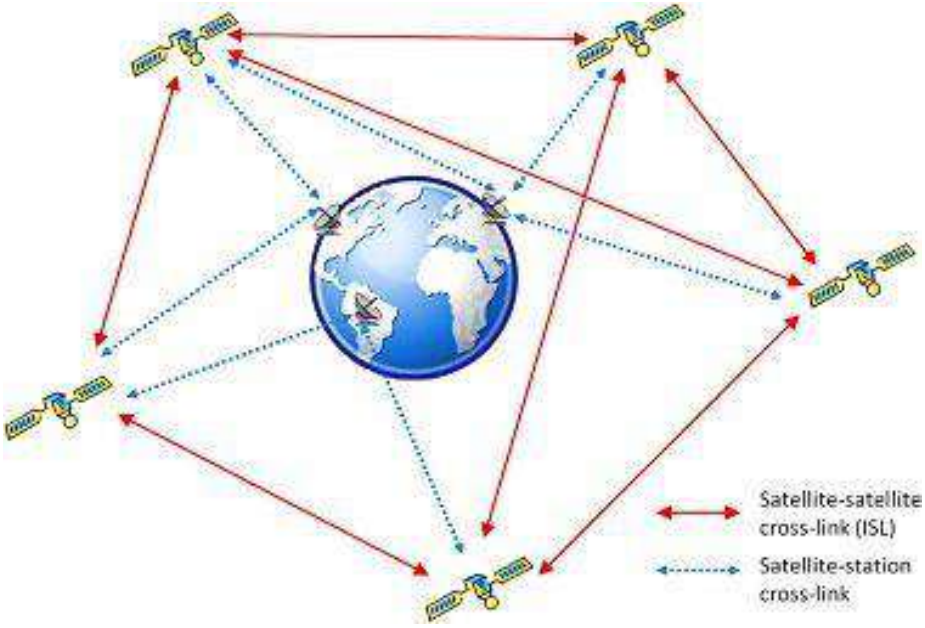



**Figure I.4:** ISL structure [33].

ISLs are generally categorized into two main types [34, 35]:

- **Intra-plane ISLs:** links between satellites within the same orbital plane. These links are relatively stable due to fixed relative positions along the orbit.
- **Inter-plane ISLs:** links between satellites in adjacent orbital planes. These are more dynamic due to continuous changes in relative geometry caused by orbital motion.

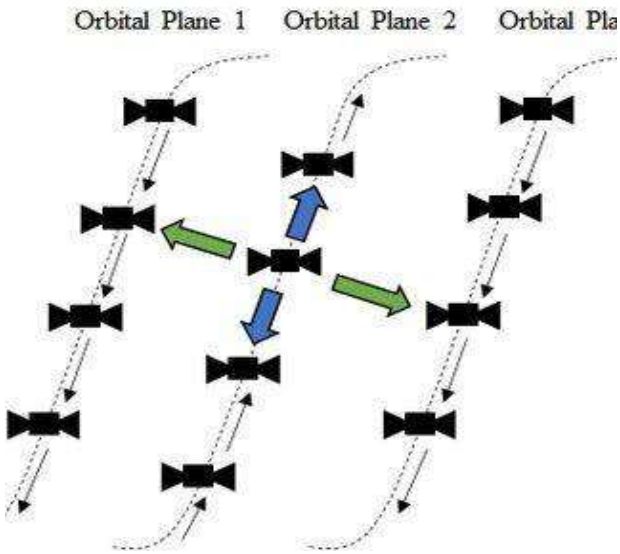



**Figure I.3:** Illustration of different types of ISL: intraplane (blue) and interplane (green) [36].

From a technological standpoint, ISLs can be implemented using radio frequency (RF) or optical (laser) communication systems. Optical ISLs are increasingly favored due to their high data rates, low latency, and immunity to electromagnetic interference. However, they require highly precise alignment and tracking mechanisms due to the high speed and motion of LEO satellites [37].

One of the most significant advantages of ISLs is their ability to reduce end-to-end latency [38]. Instead of routing traffic through terrestrial infrastructure, data can travel along near-geodesic paths in space, often resulting in faster communication over long distances. Additionally, ISLs enhance network robustness by providing multiple alternative paths, which improves fault tolerance and load distribution [39].

Despite these advantages, ISLs introduce several challenges. The dynamic nature of LEO constellations leads to frequent changes in link availability, especially for inter-plane connections. This requires routing mechanisms that can continuously adapt to topology variations [40]. Moreover, constraints such as limited onboard power, antenna steering requirements, and hardware complexity must be carefully managed [41].

From a networking perspective, ISLs are essential for enabling advanced routing strategies. They create a highly dynamic and multi-path environment, which is particularly suitable for intelligent routing approaches such as Deep Reinforcement Learning (DRL) [42]. These approaches can leverage real-time network conditions to optimize routing decisions and improve Quality of Service (QoS), especially in terms of latency and reliability.

In summary, Inter-Satellite Links are a foundational element in LEO satellite networking. They enable direct satellite-to-satellite communication, support flexible routing, reduce latency, and provide the structural basis for next-generation intelligent networking solutions.

### I.7 Satellite Constellation Routing:

Routing in satellite constellations refers to the process of transferring data packets between satellites and ground stations through optimized communication paths. In modern satellite networks, especially in Low Earth Orbit (LEO) constellations, satellites are interconnected through inter-satellite links, allowing them to exchange information directly without constant dependence on terrestrial infrastructure [49].

Unlike conventional terrestrial routing, satellite routing operates in a highly dynamic environment where satellites continuously move along orbital trajectories. As a result, communication links may frequently change, requiring routing mechanisms capable of adapting to topology variations and traffic fluctuations in real time.

The primary objective of routing is to ensure reliable and efficient communication while minimizing transmission delay, packet loss, and network congestion. Consequently, routing protocols used in satellite constellations must consider factors such as latency, link quality, bandwidth availability, and orbital movement.

### I.7.1 Routing in LEO Satellite Networks:

Routing in Low Earth Orbit satellite networks is considered one of the most challenging tasks in satellite communications due to the rapid motion of satellites and the continuously changing network topology. LEO satellites orbit close to the Earth, which significantly reduces propagation delay and improves communication speed. However, because each satellite covers only a limited geographical area, communication paths must frequently change as satellites move [13].

In this type of network, data packets are usually transmitted through multiple satellites before reaching their final destination. Therefore, routing algorithms must continuously update routing tables and select appropriate paths according to current network conditions.

One of the main goals of LEO routing is to minimize end-to-end latency while maintaining stable communication links. Efficient routing strategies also aim to distribute network traffic evenly in order to avoid congestion and improve overall network performance.

### I.7.2 Routing in MEO Satellite Networks:

Medium Earth Orbit satellite networks operate at altitudes situated between LEO and GEO systems. Compared to LEO constellations, MEO satellites move more slowly relative to the Earth and provide wider coverage areas. As a result, routing in MEO systems is generally more stable and requires fewer handovers.

Although communication latency in MEO networks is higher than in LEO systems, it remains lower than in GEO networks. Routing protocols in MEO constellations focus on

maintaining efficient communication paths while balancing transmission delay and network coverage.

MEO routing is commonly used in navigation and positioning systems, where stable connectivity and wide-area coverage are essential.

### I.7.3 Routing in GEO Satellite Networks:

Geostationary Earth Orbit satellites remain fixed relative to the Earth's surface due to their orbital altitude and rotational speed. This characteristic simplifies network management because the network topology remains almost unchanged over time [22].

Routing in GEO satellite networks is therefore more predictable compared to LEO and MEO systems. Communication links remain stable for long periods, reducing the complexity of routing decisions and topology management.

Despite these advantages, GEO systems suffer from relatively high communication delay because of the long distance between the Earth and the satellite. Consequently, GEO routing is generally more suitable for applications that prioritize coverage and stability over low latency [24].

### I.7.4 Routing in Multi-Layer Satellite Networks (LEO/MEO/GEO):

Multi-layer satellite networks combine satellites operating at different orbital layers, including LEO, MEO, and GEO systems, within a unified communication architecture. This integration allows the network to benefit from the advantages of each orbital layer.

In such architectures, routing becomes more complex because data packets may traverse multiple orbital layers before reaching the destination. Routing decisions must therefore consider differences in latency, coverage, bandwidth, and link availability between the different layers.

The objective of multi-layer routing is to optimize overall network performance by selecting the most efficient communication path according to Quality of Service requirements. This approach improves network flexibility, scalability, and communication reliability, especially in large-scale satellite systems [48].

Due to the complexity of multi-layer environments, intelligent routing approaches based on machine learning and Deep Reinforcement Learning are increasingly investigated to improve routing efficiency and adaptability.

### I.7.5 Routing Problems in LEO Satellite Networks:

Although LEO satellite networks provide important advantages such as low latency and global connectivity, they also introduce several routing challenges. The continuous movement of satellites causes frequent topology changes, making communication links highly dynamic and sometimes unstable. One major challenge is maintaining reliable routes while minimizing transmission delay. As satellites move, routing paths may rapidly become invalid, requiring continuous route updates and frequent handovers. These operations can increase packet loss and negatively affect network performance [44].

Another issue is traffic imbalance across the constellation. Some satellites may experience heavy traffic loads while others remain underutilized, leading to congestion and inefficient resource usage. In addition, routing decisions must satisfy strict Quality of Service requirements related to latency, throughput, reliability, and bandwidth efficiency.

Traditional routing methods designed for terrestrial networks are often unable to handle these highly dynamic conditions effectively. For this reason, recent research increasingly focuses on intelligent routing techniques based on Artificial Intelligence and Deep Reinforcement Learning, which are capable of adapting routing decisions according to real-time network conditions and continuously evolving topologies.

## I.8. Satellite constellation routing:

Satellite constellation routing refers to the process of selecting and managing communication paths between satellites and ground stations within a satellite constellation network. In modern satellite systems, particularly Low Earth Orbit (LEO) constellations, satellites communicate through inter-satellite links (ISLs), forming a dynamic space-based network in which each satellite can act as a routing node [43].

The main objective of routing is to determine the optimal path for transmitting data packets while satisfying network performance requirements such as low latency, high reliability, load balancing, and efficient bandwidth utilization. Unlike terrestrial networks, satellite constellation

routing must deal with continuously changing topologies caused by the orbital movement of satellites, making routing decisions more complex and time-dependent.

Routing approaches in satellite constellations may be static, dynamic, or intelligent. Recently, intelligent routing techniques based on Artificial Intelligence (AI) and Deep Reinforcement Learning (DRL) have gained significant attention due to their ability to adapt routing decisions according to real-time network conditions.

### I.8.1. Multi-Layer satellite networks:

A multi-layer satellite network is a communication architecture composed of multiple satellite layers operating at different orbital altitudes, such as Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO). Each orbital layer provides specific advantages in terms of coverage, latency, bandwidth, and communication capacity [44].

The integration of multiple orbital layers aims to combine the strengths of different satellite systems in order to improve overall network performance. LEO satellites provide low latency and high-speed communication, MEO satellites offer broader regional coverage, while GEO satellites ensure wide-area and continuous coverage.

In multi-layer architectures, satellites belonging to different orbital layers communicate through inter-satellite links, creating a heterogeneous and interconnected network infrastructure. This architecture enhances network flexibility, reliability, scalability, and traffic management capabilities.

However, multi-layer satellite networks also introduce several challenges, including complex routing, resource allocation, interoperability between layers, and dynamic topology management. Consequently, intelligent optimization techniques such as Deep Reinforcement Learning (DRL) are increasingly explored to improve routing efficiency and Quality of Service (QoS) in these environments.

### I.8.2. Routing problem in LEO satellite networks:

Routing in Low Earth Orbit (LEO) satellite networks is a fundamental and challenging problem due to the unique characteristics of the space environment [43]. Unlike terrestrial networks, where topology is relatively stable, LEO satellite networks exhibit a highly dynamic

topology caused by the continuous movement of satellites along their orbital paths [45]. This results in frequent changes in connectivity, making routing decisions significantly more complex.

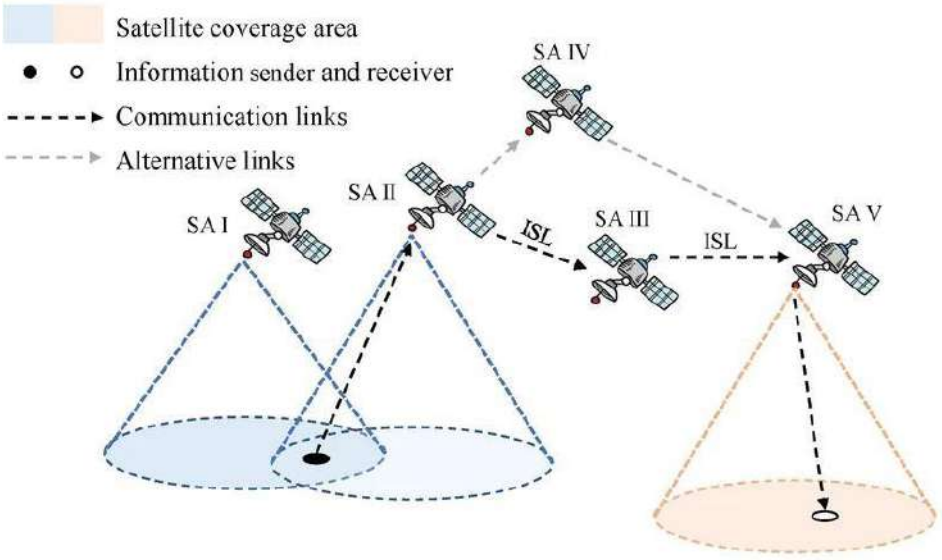



**Figure I.5:** Simplified schematic diagram of Low-Earth Orbit (LEO) satellite network [46].

One of the primary challenges in LEO routing is the time-varying network graph. As satellites orbit the Earth, links between nodes are continuously established and broken, particularly for inter-plane connections [44]. Consequently, routes that are optimal at a given moment may quickly become suboptimal or even unavailable. This dynamic behavior requires routing strategies that can adapt in real time or anticipate future topology changes.

Routing approaches in LEO networks are generally classified into two main categories [47]:

- **Snapshot-based routing:** the network topology is assumed to be static over short time intervals (snapshots). Routing tables are precomputed for each snapshot and updated periodically as the topology evolves.
- **Dynamic routing:** routing decisions are made in real time based on the current network state, allowing greater flexibility but requiring higher computational and communication overhead.

Another critical challenge is frequent handovers. Due to the high speed of satellites relative to ground users, connections must be transferred from one satellite to another within short time

periods. These handovers can introduce packet loss, increased latency, and additional signaling overhead, all of which negatively impact Quality of Service (QoS) [48].

In addition, LEO networks experience non-uniform traffic distribution. Traffic demand is typically concentrated over densely populated regions, while satellites over oceans or remote areas handle significantly less traffic. This imbalance can lead to congestion in certain parts of the network while other resources remain underutilized [48].

Routing decisions must therefore consider multiple, often conflicting objectives, including [49]:

- Minimizing end-to-end latency;
- Avoiding congestion and balancing load;
- Maintaining link stability;
- Ensuring high packet delivery ratio.

Traditional routing algorithms, such as shortest-path methods (e.g., Dijkstra), are not well suited for LEO environments, as they assume relatively stable topologies and do not account for rapid changes or future network states. Even enhanced versions of these algorithms may struggle to perform efficiently under highly dynamic conditions [49].

To address these limitations, recent research has explored more advanced approaches, including predictive routing, Software-Defined Networking (SDN), and machine learning-based techniques [50-53]. In particular, Deep Reinforcement Learning (DRL) has emerged as a promising solution, as it enables agents to learn optimal routing policies through interaction with the environment, taking into account both current and future network conditions [54].

Routing in LEO satellite networks can therefore be viewed as a complex optimization problem under uncertainty, where decisions must be continuously adapted to a rapidly evolving network state while satisfying strict QoS requirements.

### I.9. Quality of service (QOS) requirements:

Quality of Service (QoS) is a critical aspect of Low Earth Orbit (LEO) satellite networks, as these systems are expected to support a wide range of applications with strict performance constraints [55]. Due to the dynamic nature of LEO constellations and the diversity of traffic types, ensuring consistent QoS remains a significant challenge.

One of the most important QoS metrics in LEO networks is end-to-end latency. Many modern applications, such as real-time communication, online gaming, and remote-control systems, require low and stable latency [56]. LEO satellites offer a clear advantage over higher-orbit systems due to their reduced propagation delay; however, routing inefficiencies and dynamic topology changes can still introduce additional delays.

Another key parameter is the packet delivery ratio (PDR), which reflects the reliability of the network. Frequent topology changes, handovers, and link disruptions can lead to packet loss, especially if routing decisions are not updated Fast enough. Maintaining a high PDR is essential for ensuring reliable communication, particularly for mission-critical applications [57].

Throughput is also a fundamental QoS metric, representing the amount of data successfully delivered over the network within a given time. In LEO satellite systems, throughput can be affected by factors such as link capacity, congestion, and uneven traffic distribution across satellites. Efficient routing and load balancing are therefore necessary to maximize network utilization [58].

In addition, jitter, or delay variation, plays an important role in applications that require consistent data flow, such as video streaming and voice over IP (VoIP). High jitter can degrade user experience even if average latency remains low. Managing jitter in a highly dynamic environment like LEO networks requires stable routing paths and adaptive control mechanisms [59].

QoS in LEO networks are further complicated by the heterogeneity of traffic. Different applications impose different requirements [60]; for example, latency-sensitive traffic must be prioritized over delay-tolerant data transfers. This necessitates intelligent routing strategies capable of differentiating between traffic types and allocating resources accordingly.

Traditional QoS mechanisms developed for terrestrial networks are often insufficient in LEO environments due to rapid topology changes and limited onboard resources. As a result, there is increasing interest in adaptive and intelligent approaches that can dynamically optimize QoS metrics in real time [61].

In this context, latency-aware routing becomes particularly important, especially for applications requiring near real-time communication. Intelligent techniques such as Deep Reinforcement Learning (DRL) offer the ability to learn optimal routing policies that balance

multiple QoS objectives, making them well suited for the complex and dynamic nature of LEO satellite networks.

## conclusion

In conclusion, this chapter has provided a comprehensive overview of the fundamental principles governing Low Earth Orbit (LEO) satellite networks, with a specific focus on the Walker Delta constellation architecture. The analysis highlighted how this structured geometric arrangement facilitates continuous global coverage and enables the formation of interconnected, space-based mesh networks through the integration of Inter-Satellite Links (ISLs).

Furthermore, the discussion identified the significant routing challenges inherent in these highly dynamic environments, such as time-varying network topologies, frequent handovers, and non-uniform traffic distribution, all of which complicate the maintenance of stable and efficient communication paths. Meeting rigorous Quality of Service (QoS) requirements, particularly regarding end-to-end latency and packet delivery ratio (PDR), remains essential for supporting modern, real-time applications.

These operational complexities and the limitations of traditional routing algorithms in the face of rapid topological shifts underscore the critical need for more adaptive and intelligent pathfinding solutions. Consequently, the next chapter will transition into a detailed examination of Deep Reinforcement Learning (DRL) as a powerful framework for addressing these dynamic optimization challenges in satellite networking.

# CHAPTER II:

## FUNDAMENTALS OF DEEP REINFORCEMENT LEARNING (DRL) IN NETWORKING

# CHAPTER II: FUNDAMENTALS OF DEEP REINFORCEMENT LEARNING (DRL) IN NETWORKING

## Introduction:

The increasing complexity and dynamic nature of modern communication networks have made traditional optimization and rule-based approaches insufficient for efficient decision-making. In environments such as Low Earth Orbit (LEO) satellite networks, where topology changes rapidly and multiple objectives must be balanced, adaptive and data-driven solutions are required.

Reinforcement Learning (RL) provides a powerful framework for sequential decision-making, where an agent learns optimal actions through interaction with its environment. By extending RL with deep neural networks, Deep Reinforcement Learning (DRL) enables the handling of large and complex state spaces, making it suitable for real-world networking problems.

In the context of networking, DRL has been increasingly applied to tasks such as routing, congestion control, and resource allocation. Its ability to learn from experience and adapt to changing conditions makes it particularly relevant for highly dynamic systems like LEO satellite networks.

This chapter presents the fundamental concepts of Reinforcement Learning and its extension to Deep Reinforcement Learning. It also explains how routing problems can be modeled as Markov Decision Processes (MDPs) and discusses key components such as neural network design and reward function formulation.

## II.1. Reinforcement learning (RL) Concepts:

Reinforcement Learning (RL) is a paradigm of machine learning in which an agent learns optimal behavior through interaction with an environment, aiming to maximize cumulative rewards over time [62]. Unlike supervised learning, RL does not rely on labeled datasets but instead learns from feedback signals obtained through trial-and-error interactions.

An RL system is typically defined by the following core components [63]:

- ✓ **Agent:** the decision-making entity that interacts with the environment.
- ✓ **Environment:** the system with which the agent interacts. It provides feedback in response to the agent's actions.

- ✓ **State (s):** a representation of the current situation of the environment.
- ✓ **Action (a):** a decision taken by the agent based on the current state.
- ✓ **Reward (r):** a scalar feedback signal that evaluates the quality of an action.

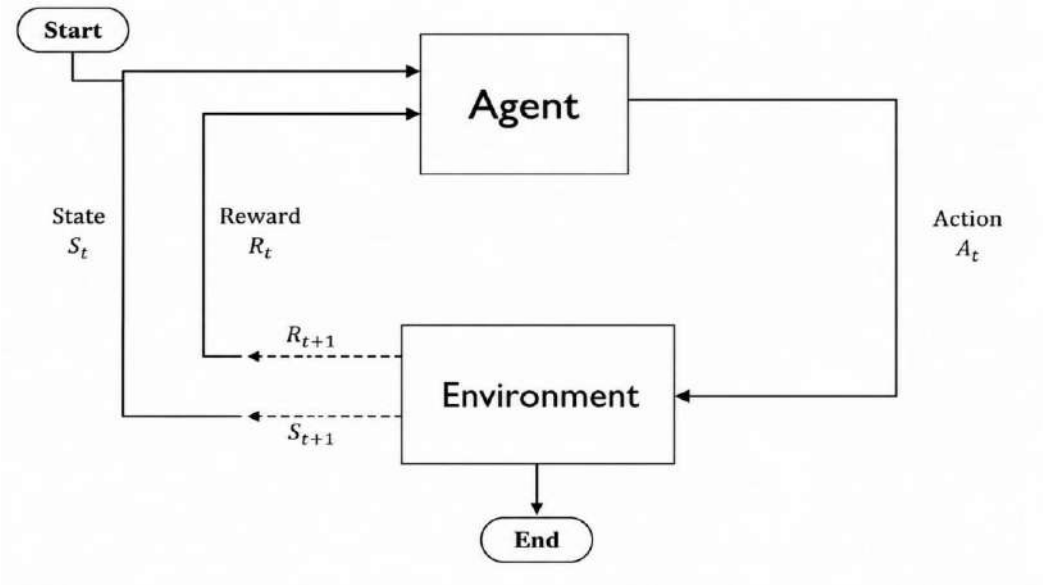



**Figure II.1:** Overview model of Reinforcement Learning [64].

At each time step $t$, the agent observes the current state $S_t$, selects an action $A_t$according to a policy $\pi(a \mid s)$, and receives a reward $R_t$, while transitioning to a new state $S_{t+1}$. This sequential interaction forms a trajectory:

$$\tau = \{(S_t, A_t, R_t)\}_{t=0}^{T} \qquad \text{Eq.(II.1)}$$

The fundamental objective of RL is to maximize the expected return, defined as the discounted cumulative reward [65]:

$$G = \sum_{t=0}^{T} \gamma^t R_t \qquad \text{Eq.(II.2)}$$

where $\gamma \in [0,1]$is the discount factor that balances immediate and future rewards.

### II.1.1. What is Deep Reinforcement Learning (DRL)?

#### a) Definition:

Deep Reinforcement Learning (DRL) is a learning paradigm in which an agent interacts with an environment by taking actions, receiving rewards, and updating a policy so as to maximize

the expected cumulative reward. DRL combines reinforcement learning (RL), which formalizes

decision making as a sequential control problem, with deep neural networks (DNNs) as function approximators for policies, value functions, and/or action-value ("Q") functions [65].

**b) Mathematical Model (MDP):**

DRL typically models the problem as a Markov Decision Process (MDP) :

$$M = \{S, A, P, r, \gamma\} \qquad \text{Eq.(II.3)}$$

where:

- $S$: state space (environment observations).
- $A$: action space (choices the agent can make).
- $P(s' \mid s, a)$: transition dynamics (probability of next state).
- $r(s, a)$: reward function (scalar feedback).
- $\gamma \in (0,1)$: discount factor for future returns.

The agent seeks a policy $\pi(a \mid s)$that maximizes the expected return:

$$J(\pi) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^{\pi} \gamma_t] \qquad \text{Eq.(II.4)}$$

Where:

- $J(\pi)$: Expected cumulative reward under policy $\pi$.
- $\mathbb{E}_{\pi}$: Expectation under policy $\pi$.
- $t$: Time step index.
- $\gamma$: Discount factor ( $0 < \gamma < 1$ ).
- $r_t$: Reward received at time step $t$.
- $\pi$: Policy followed by the agent.

**c) Core DRL quantities and equations:**

**Value Functions**

- **State value:**

$$V^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{k=0}^{\infty} \gamma^k \, r_{t+k} \mid s_t = s] \qquad \text{Eq.(II.5)}$$

Where:

- $V^{\pi}(s)$: Expected return starting from state $s$under policy $\pi$.
- $s$: Current state.
- $\mathbb{E}_{\pi}$: Expectation under policy $\pi$.
- $k$: Future time-step offset.
- $\gamma$: Discount factor.
- $r_{t+k}$: Reward received at future time step $t+k$.
- $s_t = s$: Condition indicating that the agent starts from state $s$at time $t$.

**Action value:**

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_t = s, a_t = a] \qquad \text{Eq.(II.6)}$$

Where:

- $Q^{\pi}(s,a)$: Expected return obtained by taking action $a$in state $s$under policy $\pi$.
- $s$: Current state.
- $a$: Action selected by the agent.
- $a_t = a$: Condition indicating that action $a$is taken at time $t$.
- $\gamma$: Discount factor.
- $r_t$: Reward at time step $t$.

**d) Bellman optimality equation:**

$$Q^{\star}(s,a) = \mathbb{E}\,[r_t + \gamma \max_{a'} Q^{\star}(s_{t+1}, a') \mid s, a] \qquad \text{Eq.(II.7)}$$

Where:

- $Q^{\star}(s,a)$: Optimal action-value function.
- $r_t$: Immediate reward received at time step $t$.
- $\gamma$: Discount factor.
- $\max_{a'}$: Maximum value over all possible actions $a'$.
- $Q^{\star}(s_{t+1}, a')$: Optimal future Q-value at next state $s_{t+1}$.
- $s_{t+1}$: Next state after taking action $a$.
- $a'$: Possible future action.

- $\mathbb{E}$: Expectation operator.

**e) Policy Gradient Methods**

The policy gradient theorem provides:

$$\nabla_\theta J(\pi_\theta) \approx \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\hat{A}_t] \quad \text{Eq.(II.8)}$$

Where:

- $\nabla_\theta$: Gradient with respect to parameters $\theta$.
- $J(\pi_\theta)$: Objective function of policy $\pi_\theta$.
- $\pi_\theta(a_t \mid s_t)$: Probability of selecting action $a_t$in state $s_t$.
- log: Natural logarithm.
- $\hat{A}_t$: Advantage estimate at time step $t$.
- $a_t$: Action selected at time $t$.
- $s_t$: State observed at time $t$.
- $\theta$: Trainable parameters of the policy network.

**f) Popular Objectives**

- **DQN (value-based, discrete actions)**

$$L(\theta) = (Q_\theta(s,a) - [r + \gamma \max_{a'} Q_{\theta^-}(s',a')])^2 \quad \text{Eq.(II.9)}$$

Where:

- $L(\theta)$: Loss function.
- $Q_\theta(s,a)$: Predicted Q-value using parameters $\theta$.
- $r$: Immediate reward.
- $\gamma$: Discount factor.
- $\max_{a'} Q_{\theta^-}(s',a')$: Maximum target Q-value at next state $s'$.
- $\bar{\theta}$ : Parameters of the target network.
- $s'$: Next state.
- $a'$: Possible next action.

- **PPO (actor-critic)**

$$L^{clip}(\theta) = \mathbb{E}[\min(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad \text{Eq.(II.10)}$$

Where:

- $L^{clip}(\theta)$: PPO clipped objective function.
- $\mathbb{E}$: Expectation operator.
- $\rho_t$: Probability ratio between new and old policies.
- $\hat{A}_t$: Advantage estimate.
- clip(·): Clipping function limiting policy updates.
- $\epsilon$: Clipping threshold hyperparameter.

with

$$\rho_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad \text{Eq.(II.11)}$$

Where:

- $\rho_t$: Probability ratio at time step $t$.
- $\pi_\theta(a_t \mid s_t)$: Probability under the updated policy.
- $\pi_{\theta_{old}}(a_t \mid s_t)$: Probability under the old policy.
- $a_t$: Action selected at time $t$.
- $s_t$: State observed at time $t$.
- $\theta$: Current policy parameters.
- $\theta_{old}$: Previous policy parameters.

### II.1.2. How DRL Works (End-to-End Loop)

**a) Perception (State):**

At time $t$, the agent receives a state $s_t$(e.g., local neighborhood features on a time-varying network).

**b) Action Selection:**

A policy $\pi_\theta(a \mid s_t)$produces a distribution over feasible actions; the agent samples or chooses greedily.

**c) Environment Step:**

The environment returns a reward $(r_t)$and a next state $(s_{t+1})$.

**d) Storage:**

The agent stores $(s_t, a_t, r_t, s_{t+1})$and masks/valid action sets, if any.

**e) Learning Update:**

After a batch of steps, the agent updates:

- **Value-based (e.g., DQN):** fit $Q_\theta$to TD targets.
- **Actor-critic (e.g., PPO):** update the policy with clipped objectives using advantages $(\hat{A}_t)$, and update the value network $(V_\phi)$.

**f) Repeat:**

The process continues until convergence or budget exhaustion; periodically evaluate the trained policy.

**II.1.3. Why "Deep"?**

Classical Reinforcement Learning struggles when the state space $(S)$is large, continuous, or when the mapping from states to optimal actions is highly nonlinear. DRL addresses this limitation by using deep neural networks to approximate:

- $\pi_\theta(a \mid s)$(policy networks),
- $V_\phi(s)$or $Q_\theta(s, a)$(value networks).

This enables generalization across states, scalability to large action spaces, and learning directly from structured features such as vectors, images, or graphs.

**a) Design Choices and Good Practices:**

**On-policy vs Off-policy:**

- **On-policy (e.g., PPO):** learns from the current policy's data; stable and robust.

- **Off-policy (e.g., DQN/TD3/SAC):** reuses replay buffers; sample efficient and powerful with careful tuning.

**b) Exploration vs Exploitation:**

Entropy bonuses, $\epsilon$-greedy strategies, or parameter noise prevent premature convergence and encourage efficient exploration.

**c) Normalization and Scaling**

Normalize inputs (state features) and key reward components (e.g., distance, latency) to stabilize training.

**d) Action Masking (Constrained Decisions)**

If some actions are invalid (e.g., blocked links), mask them before the softmax or max operator. This prevents learning from impossible actions and drastically improves stability.

**e) Stability Tricks**

Target networks, gradient clipping, advantage normalization, learning-rate schedules, and early stopping are standard tools for robust training.

### II.1.4. How DRL Maps to QoS-Aware Routing (Example)

- **State** $(s_t)$**:** local view; per-neighbor features (distance, latency surrogate, line-of-sight flags), progress toward the destination, and time index.
- **Action** $(a_t)$**:** choose the next hop from feasible neighbors (after masks for visibility/range); include a "terminate" action at the destination.
- **Reward** $(r_t)$**:** scalarized QoS penalty (smaller is better):

$$r_t = -\alpha \cdot \text{distance} - \beta \cdot \text{hop} - \gamma \cdot \text{latency} + \delta \cdot \mathbf{1}\{\text{destination}\} \quad \text{Eq.(II.12)}$$

Where:

- $r_t$: Reward at time step $t$.
- $\alpha$: Weight associated with distance penalty.
- $\beta$: Weight associated with hop-count penalty.

- $\gamma$: Weight associated with latency penalty.
- $\delta$: Positive reward coefficient for reaching destination.
- distance: Distance metric between satellites.
- hop: Number of hops in the routing path.
- latency: End-to-end communication delay.
- $\mathbf{1}\{\text{destination}\}$: Indicator function equal to 1 if the destination is reached, otherwise 0.

- **Objective:** learn a policy that minimizes total path length, hop count, and end-to-end latency on a time-varying graph while achieving near-Pareto routing decisions.

### II.1.5. When to Use Which Algorithm?

- **Discrete, masked actions with moderate branching:** DQN (with invalid-action masking) or PPO with masked softmax.
- **Large, structured neighborhoods (graphs):** consider GNN-enhanced actor-critic encoders.
- **Non-stationary environments (e.g., moving satellites):** prefer PPO or robust actor-critic variants; their on-policy updates tolerate distribution shifts more gracefully.

### II.1.6. From RL to Deep Reinforcement Learning:

Traditional RL struggles with high-dimensional state and action spaces. To address this limitation, Deep Reinforcement Learning (DRL) integrates neural networks as function approximators [66].

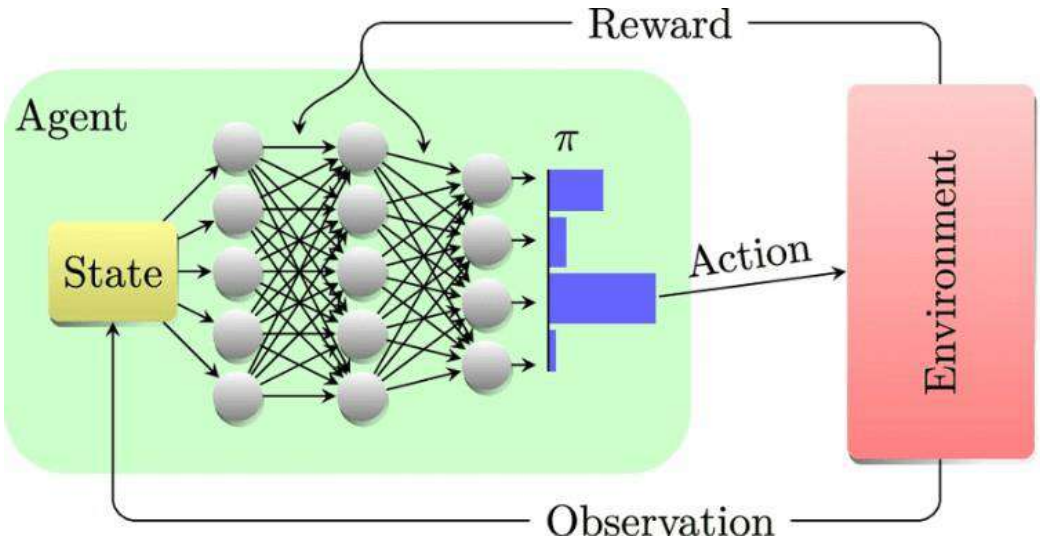



**Figure II.2:** Deep reinforcement learning [67].

For example:

Q-values can be approximated using deep neural networks (Deep Q-Networks)

Policies can be parameterized and optimized using gradient-based methods

This combination allows RL to scale to complex problems such as robotics, games, and network optimization, as emphasized in recent surveys and studies (e.g.,).

## II.2. Deep Neural Networks in Routing:

Deep Neural Networks (DNNs) have emerged as an effective tool for addressing complex optimization problems in network routing, particularly in the context of Traffic Engineering (TE). Traditional routing approaches often rely on solving optimization problems such as Mixed Integer Linear Programming (MILP), which can provide optimal solutions but are computationally expensive and impractical for real-time deployment. Consequently, heuristic-based methods are typically used, although their design is often complex and does not guarantee optimal performance [68].

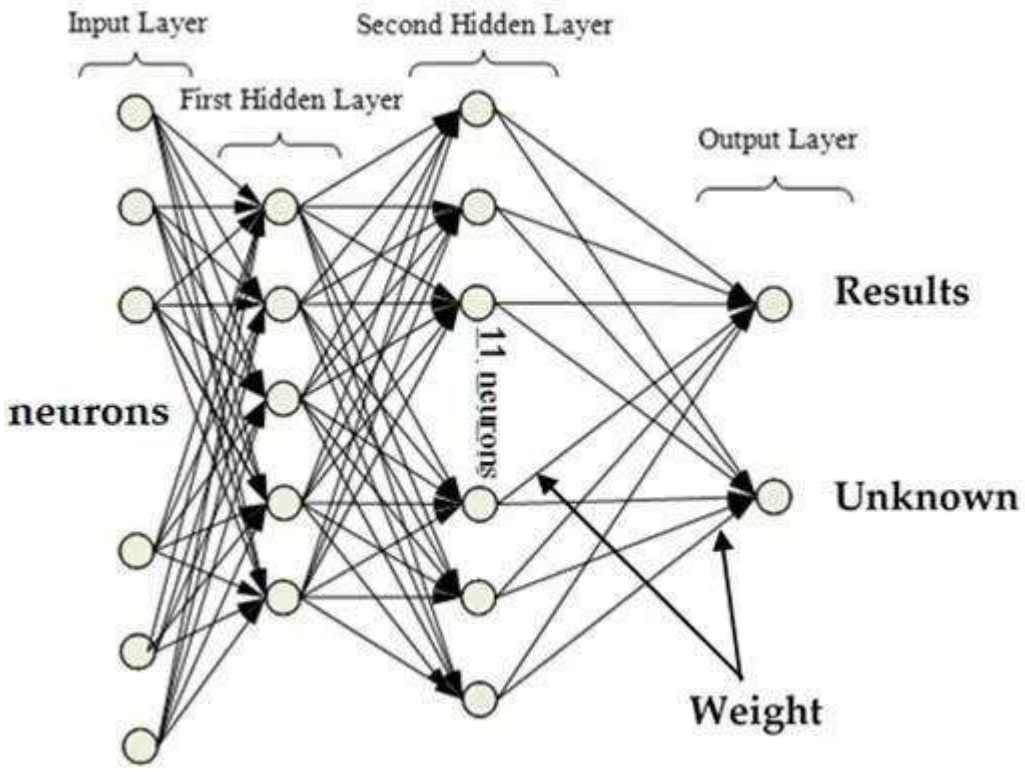



**Figure II.3:** A basic structure of the deep neural network [69].

DNNs offer an alternative by learning routing decisions directly from data. Instead of solving optimization problems online, a neural network can be trained offline using optimal or near-optimal routing solutions. Once trained, the model can generate routing decisions efficiently, making it suitable for dynamic and time-sensitive environments [70].

In routing applications, a DNN learns a mapping between network states and routing actions. The input to the model generally includes features that describe both the traffic demand and the current network condition, such as source and destination nodes, bandwidth requirements, and link utilization across the network. The output represents routing decisions, which can be expressed as path selections, next-hop choices, or link activation probabilities [71].

This approach enables the model to approximate complex optimization processes. By learning from a dataset generated using optimization techniques, the neural network captures underlying patterns in traffic distribution and network behavior [72]. As a result, it can produce routing decisions that achieve near-optimal performance while significantly reducing computational overhead.

One of the main advantages of DNN-based routing is its ability to generalize. After training, the model can adapt to unseen network conditions and traffic patterns, allowing it to operate effectively in dynamic environments. This is particularly relevant in scenarios where network states change frequently and rapid decision-making is required [73].

However, several challenges must be addressed. The performance of the model depends heavily on the quality and diversity of the training data [74]. In addition, neural network outputs do not always correspond to valid routing paths, which requires the use of post-processing techniques to ensure feasibility [75]. For example, graph-based algorithms can be applied to transform the model outputs into valid paths.

Despite these challenges, experimental results show that DNN-based approaches can achieve performance close to optimal routing solutions, particularly in terms of congestion minimization and efficient resource utilization [73, 76, 77].

### II.3. Modeling routing as a markov decision process (MDP):

Markov Decision Processes (MDPs) provide a rigorous mathematical framework for modeling sequential decision-making under uncertainty, making them particularly suitable for dynamic routing problems [78]. An MDP models the interaction between an agent and its environment through a set of states, actions, transition dynamics, and rewards, where decisions are made to maximize the expected cumulative reward over time [79].

Formally, an MDP is defined by a tuple consisting of a state space, an action space, a transition function, and a reward function. At each decision step, the agent observes the current state, selects an action according to a policy, and transitions to a new state based on probabilistic dynamics while receiving a reward. The objective is to determine an optimal policy that maximizes the expected sum of rewards over a given horizon [80].

In routing contexts, MDPs are used to model the sequential nature of path selection, where decisions are not limited to choosing a complete path in advance but instead consist of making step-by-step routing decisions. This perspective allows the system to incorporate uncertainty in traffic demand, travel time, and network conditions [81]. MDPs explicitly capture the interaction between state, action, reward, and transition dynamics, enabling the modeling of agent behavior over time.

A key advantage of MDP-based routing is its ability to account for future consequences of current decisions. Instead of optimizing only immediate costs, the agent evaluates both immediate rewards and expected future rewards through a recursive value function [82]. This recursive structure allows the modeling of trade-offs between short-term gains and long-term performance, which is essential in dynamic environments such as transportation networks.

In network routing, states are often defined as nodes or links, and actions correspond to selecting the next node or link. This leads to a formulation where routing decisions are made sequentially, effectively transforming the routing problem into a stochastic control problem [83]. The diagram on page 8 illustrates this idea by showing that routing is not a single decision but a sequence of local decisions, each influenced by expected future outcomes.

Recent research extends this framework to more complex routing scenarios. For example, MDP routing games model situations where multiple agents simultaneously make routing decisions while interacting with each other. In such settings, each agent selects a transition policy rather than a fixed path, and the reward depends not only on individual decisions but also on the collective behavior of the population [84-86].

Moreover, in stochastic dynamic vehicle routing problems (SDVRPs), MDPs are used to capture the evolution of information over time, such as the arrival of new customer requests or changes in travel conditions. These problems are inherently sequential and stochastic, making MDPs a natural modeling choice [87]. However, classical MDP formulations may struggle to represent route planning explicitly, leading to the development of route-based MDPs, which extend the state and action spaces to include planned routes and better align with practical solution methods [88].

Despite their strong modeling capabilities, MDP-based approaches face several challenges in routing applications. The state and action spaces can become extremely large, especially in real-world networks, making exact solutions computationally intractable [82]. Additionally, accurately modeling transition probabilities and reward functions requires high-quality data and domain knowledge.

To address these limitations, recent works combine MDPs with machine learning techniques, such as reinforcement learning and deep neural networks, to approximate optimal

policies. These approaches enable scalable solutions while preserving the ability to model uncertainty and sequential decision-making [89].

In summary, MDPs provide a powerful and flexible framework for modeling routing problems under uncertainty. They shift the perspective from static path optimization to dynamic decision-making, enabling more realistic and adaptive routing strategies in complex and stochastic environments.

### Ii.4. Reward function design:

The reward function is a fundamental component of reinforcement learning (RL), as it defines the objective that the agent seeks to optimize. It provides numerical feedback based on the agent's actions and directly shapes the learned policy [90]. The quality of the reward function has a critical impact on the performance of the learning process, as it determines whether the agent converges toward desirable or unintended behaviors.

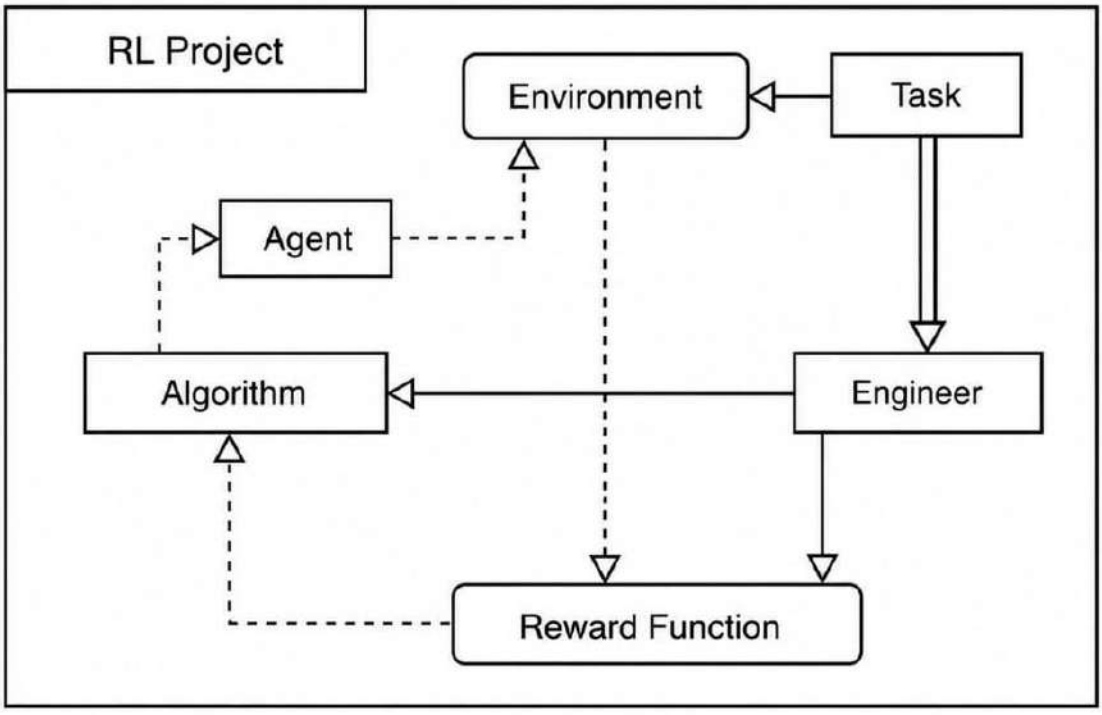



**Figure II.4:** Structure of a prototypical reinforcement learning project emphasizing the design of the reward function by the engineer based on the specified task [91].

In its simplest form, a reward function assigns positive values for desirable outcomes and negative values for undesirable ones [92]. While this approach is sufficient for simple tasks, it

becomes inadequate in complex environments such as network routing, where multiple objectives must be considered simultaneously. In such cases, the reward function must capture not only the final goal but also the quality of intermediate decisions [93].

A key distinction in reward design is between sparse (goal-oriented) and dense (process-oriented) rewards. Sparse rewards provide feedback only when a final objective is achieved, such as successfully delivering a packet or reaching a destination. Although simple, this approach often leads to slow convergence due to the lack of intermediate feedback [94]. In contrast, dense rewards evaluate intermediate states and actions, providing continuous guidance that accelerates learning and improves stability [95].

In practical applications, especially in dynamic systems, it is essential to consider multiple performance criteria simultaneously. Therefore, reward functions are typically designed as a combination of several components, each reflecting a specific objective [96]. In routing problems, these components may include latency, packet loss, congestion levels, and energy efficiency. The overall reward is then computed as an aggregation of these factors, allowing the agent to balance trade-offs between competing objectives [97].

More advanced reward formulations adopt a structured approach in which each objective is evaluated independently through a dedicated function. These evaluation functions are then combined to produce the final reward. This modular design offers greater flexibility, as new criteria can be added without modifying the entire reward structure, and priorities can be adjusted by tuning the contribution of each component [98].

Despite its importance, reward function design remains a challenging task. One major issue is reward misalignment, where the agent learns to exploit the reward function in unintended ways without achieving the true objective [99]. Another challenge is balancing multiple objectives that may conflict with each other, requiring careful calibration to avoid bias toward a single criterion. Additionally, overly complex reward functions can make the learning process unstable or difficult to interpret [100].

To address these challenges, several strategies are commonly used. Reward shaping introduces intermediate feedback to guide the agent more effectively. Multi-objective optimization

techniques allow explicit handling of competing goals. In some cases, reward functions can be partially learned from expert behavior, reducing the need for manual design [95].

In the context of routing, an effective reward function must reflect the key performance indicators of the network while ensuring stability and adaptability. It should encourage low latency, efficient resource utilization, and robustness under dynamic conditions. A well-designed reward function is therefore essential for enabling intelligent and reliable routing decisions in DRL-based systems.

# CHAPTER III:

## SIMULATION, IMPLEMENTATION, AND RESULTS DISCUSSION

# CHAPTER III: SIMULATION, IMPLEMENTATION, AND RESULTS DISCUSSION

## Introduction:

The rapid evolution of satellite communication systems, particularly in Low Earth Orbit (LEO), has led to the emergence of large-scale constellations composed of hundreds of satellites. Among the most widely adopted architectures, the Walker-Delta constellation offers a structured and efficient framework for achieving global coverage with predictable satellite distribution. However, as the number of satellites increases—such as in a 100-satellite constellation—the complexity of routing decisions becomes significantly more challenging, especially under dynamic network conditions.

Traditional routing approaches in satellite networks often rely on static or heuristic-based algorithms, which are not well-suited to handle the highly dynamic topology, frequent link disruptions, and varying traffic demands inherent in LEO environments. These limitations directly impact critical performance metrics such as end-to-end latency, packet delivery ratio, and overall Quality of Service (QoS).

To address these challenges, this chapter proposes a practical implementation of an intelligent routing framework based on Deep Reinforcement Learning (DRL). Specifically, a Deep Q-Network (DQN) model is employed to enable adaptive and autonomous decision-making in the routing process. The proposed framework allows satellites to learn optimal routing policies through interaction with the network environment, thereby improving latency performance and ensuring QoS requirements are met.

This chapter focuses on the MATLAB-based implementation of the proposed DRL routing framework within a Walker-Delta constellation consisting of 100 satellites. The simulation environment is carefully designed to model orbital dynamics, inter-satellite links (ISLs), and traffic flows. The integration of the DRL agent with the satellite network enables real-time route optimization based on network state observations.

The main contributions of this chapter can be summarized as follows:

- Development of a realistic MATLAB simulation model for a 100-satellite Walker-Delta constellation.

- Design and implementation of a DRL-based routing strategy using the DQN algorithm.
- Definition of a reward function that captures latency and QoS optimization objectives.
- Performance evaluation through simulations, including comparisons with baseline routing methods.

The remainder of this chapter is organized as follows: Section 2 presents the system model and network architecture. Section 3 details the mathematical formulation of the routing problem. Section 4 introduces the proposed DRL-based routing approach. Section 5 describes the MATLAB simulation setup. Section 6 discusses the obtained results, and Section 7 concludes the chapter.

## III.1 Orbital Design and Analysis of a 100-Satellite Constellation Using the Walker Delta Method:

The spatial architecture of the proposed satellite constellation is developed utilizing the Walker Delta geometric pattern. This methodology, formally denoted by the standard nomenclature : $i : T/P/F$, ensures a highly symmetrical and homogenous distribution of satellites across circular orbits, which is essential for optimizing continuous global coverage. The fundamental parameters governing this constellation model are defined as follows:

- $\boldsymbol{T}$: The total number of satellites within the constellation network.
- $\boldsymbol{P}$: The total number of distinct orbital planes.
- $\boldsymbol{S}$: The number of equally spaced satellites allocated per orbital plane.
- $\boldsymbol{F}$: The relative phasing parameter, constrained to an integer value ranging from 0 to $P - 1$.
- $\boldsymbol{i}$: The designated orbital inclination angle.

### III.1.1 Mathematical Formulation of the Constellation:

The mathematical framework detailing the spatial and angular distribution of the constellation is derived through the following fundamental equations:

### A. Satellite Distribution per Plane (*S*)

The number of satellites allocated to each orbital plane is determined by the ratio of the total constellation size to the number of symmetric planes. This requires $\boldsymbol{T}$ to be evenly divisible by $\boldsymbol{P}$ :

$$S = \frac{T}{P} \qquad \text{Eq.(III.1)}$$

### B. Right Ascension of the Ascending Node (RAAN) Spacing ()

To ensure uniform global coverage, the orbital planes are distributed symmetrically around the Earth. The inter-plane angular separation along the equator is calculated as:

$$\Delta\Omega = \frac{360^o}{P} \qquad \text{Eq.(III.2)}$$

### C. True Anomaly Spacing $\Delta\theta$

The intra-plane angular spacing defines the physical distance between adjacent satellites orbiting within the same plane. This uniform separation is expressed as:

$$\Delta\,\theta = \frac{360^o}{S} \qquad \text{Eq.(III.3)}$$

### D. Inter-Plane Phase Shift (ΔΦ)

To mitigate collision risks at orbital intersection nodes and to enhance the overlapping geographic footprint, a kinetic phase shift is applied. This angle dictates the offset of a satellite relative to its counterpart in the adjacent plane:

$$\Delta\,\Phi = F \cdot \frac{360^0}{T} \qquad \text{Eq.(III.5)}$$

## III.2 Numerical Application: Configuration of a 100-Satellite Constellation:

In order to validate the theoretical model and establish a baseline for the system architecture, a numerical application is conducted for a proposed constellation comprising 100 satellites.

### III.2.1 System Parameters and Assumptions:

The structural parameters are selected to establish a robust coverage network, utilizing standard Low Earth Orbit (LEO) telecommunication characteristics:

- Total satellite count: $\boldsymbol{T = 100}$
- Selected number of orbital planes: **P = 10** * Selected phasing parameter: **F = 2**
- Orbital inclination angle: $\boldsymbol{i = 53}^{\circ}$ Based on the established Walker Delta notation, the comprehensive constellation configuration for this study is formally expressed as $\boldsymbol{53}^{\circ}$: **100 / 10 / 2** .

### III.2.2 Step-by-Step Computations and Results:

Substituting the proposed system parameters into the governing theoretical equations yields the following architectural framework:

**1. Satellites per Orbital Plane:**

$$S = 100/10 = 10$$

The system will be structured with 10 active satellites operating symmetrically within each of the 10 defined orbital planes.

**2. RAAN Separation:**

*Analysis:* The ascending nodes of the 10 orbital planes are distributed evenly, separated by an angular distance of 36° along the equatorial plane.

$$\Delta\Omega = \frac{360^{\circ}}{10} = 36^{\circ} \qquad \text{Eq.(III.6)}$$

**3. Intra-Plane Spacing:**

$$\Delta\theta = \frac{360^{\circ}}{10} = 36^{\circ} \qquad \text{Eq.(III.7)}$$

Within each distinct orbital plane, the 10 respective satellites maintain a strict, uniform angular separation of 36° in true anomaly to ensure continuous line-of-sight sequencing.

**4. Phase Difference:**

$$\Delta\Phi = 2 \cdot \frac{360^{\circ}}{100} = 7.2^{\circ} \qquad \text{Eq.(III.8)}$$

*Analysis:* Satellites positioned in any given plane are deliberately phase-shifted by 7.2° relative to their nominal counterparts in the adjacent plane. This spatial offset is a critical design feature intended to actively prevent inter-satellite collisions at higher latitudes and to optimize the overall geometric mesh of the terrestrial coverage area.

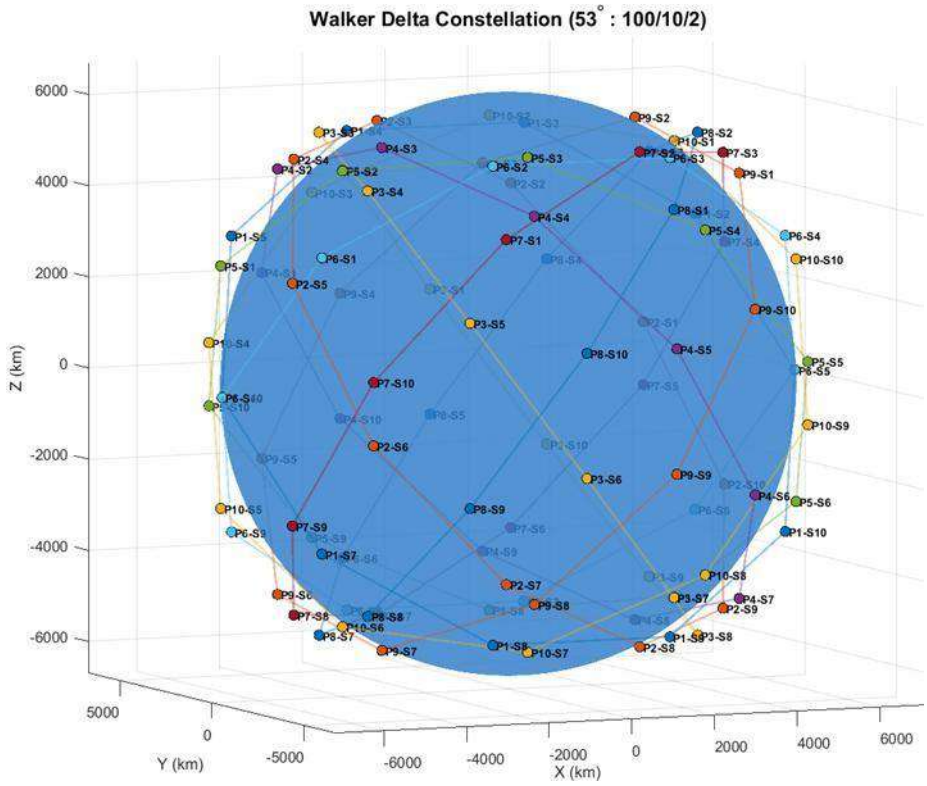



**Figure III.1:** Three-Dimensional Spatial Distribution of the 53°: 100/10/2 Walker Delta Satellite Constellation

This figure illustrates the complete geometric and architectural configuration of the 100-satellite constellation modeled in a Low Earth Orbit (LEO) environment.

**III.3. Kinematic-Driven DRL Environment for Multi-Objective Satellite Routing:**

To transition the deterministic kinematic orbit propagation model into an autonomous routing mechanism, the time-series state vectors of the 100-satellite constellation must be

formalized into a dynamic environment. The satellite routing challenge is mathematically formulated as a Multi-Objective Markov Decision Process (MOMDP) over a time-varying graph.

The primary objective of the Deep Reinforcement Learning (DRL) agent is to discover an end-to-end routing path from a source node to a destination node that simultaneously optimizes four conflicting criteria: minimizing geographic path distance, minimizing propagation latency, minimizing the cumulative Bit Error Rate (BER), and minimizing the total number of intermediate satellite hops (energy and node conservation).

### III.3.1 Dynamic Network Graph Topology:

At any discrete temporal epoch $t$, the satellite network is modeled as a time-varying directed graph:

$$G(t) = (V, E(t)) \qquad \text{Eq.(III.9)}$$

Where:

- $V = \{v_1, v_2, \dots, v_T\}$ is the static set of vertices representing the $T = 100$ satellites within the Walker Delta constellation.
- $E(t)$ is the dynamic set of edges representing active Inter-Satellite Links (ISLs). An edge

$e_{ij}(t) \in E(t)$ exists between satellite $i$ and satellite $j$ if and only if they satisfy the line-of-sight (LoS) visibility matrix and their instantaneous physical distance is bounded by the maximum communication transceiver range $D_{max}$:

$$E(t) = \{e_{ij}(t) \mid d_{ij}(t) \leq D_{max} \wedge \text{LoS}_{ij}(t) = 1\} \qquad \text{Eq.(III.10)}$$

### III.3.2 Mathematical Formulation of Routing Optimization Metrics:

To quantify path optimality for the DRL agent, the environment continuously computes the following physical-layer and topological metrics at every one-second time step:

#### A. Inter-Satellite Euclidean Distance ($d_{ij}$)

The instantaneous geometric distance between any two interconnected satellites $i$ and $j$ is derived directly from the Cartesian state vectors generated by the kinematic propagator:

$$d_{ij}(t) = \sqrt{(X_i(t) - X_j(t))^2 + (Y_i(t) - Y_j(t))^2 + (Z_i(t) - Z_j(t))^2} \quad \text{Eq.(III.11)}$$

**B. End-to-End Propagation Latency ($\tau_{path}$)**

The single-hop propagation delay is a direct linear function of the Euclidean distance, governed by the speed of light in a vacuum ($c \approx 3 \times 10^8$ m/s):

$$\tau_{ij}(t) = \frac{d_{ij}(t)}{c} \quad \text{Eq.(III.12)}$$

For a complete selected routing path $P$ composed of a sequence of $M$ satellites, the total accumulated propagation latency is the summation of individual hop delays:

$$\tau_{path}(t) = \sum_{k=1}^{M-1} \tau_{k,k+1}(t) \quad \text{Eq.(III.13)}$$

**C. Link and Cumulative Path Bit Error Rate ($BER_{path}$)**

The signal quality degrades dynamically due to Free-Space Path Loss ($FSPL_{ij}$), which expands over the time-varying distance:

$$FSPL_{ij}(t) = \left(\frac{4\pi f \cdot d_{ij}(t)}{c}\right)^2 \quad \text{Eq.(III.14)}$$

Where $f$ denotes the carrier frequency of the cross-link RF/optical transceiver. Consequently, the instantaneous Signal-to-Noise Ratio ($SNR$) received at node $j$ from node $i$ is formulated via the link budget equation:

$$SNR_{ij}(t) = \frac{P_{tx} \cdot G_{tx} \cdot G_{rx}}{N_0 \cdot B \cdot FSPL_{ij}(t)} \quad \text{Eq.(III.15)}$$

Here, $P_{tx}$ is the transmission power, $G_{tx}$ and $G_{rx}$ represent antenna gains, $N_0$ is the noise power spectral density, and $B$ is the channel bandwidth. Assuming a Binary Phase Shift Keying (BPSK) modulation scheme for the LEO constellation, the Bit Error Rate for a single hop is given by the complementary error function:

$$BER_{ij}(t) = \frac{1}{2}\operatorname{erfc}(\sqrt{SNR_{ij}(t)}) \quad \text{Eq.(III.16)}$$

Assuming independent memoryless channels across distinct hops, the cumulative end-to-end path BER ($BER_{path}$) across the selected satellite sequence is non-linearly accumulated as:

$$BER_{path}(t) = 1 - \prod_{k=1}^{M-1} (1 - BER_{k,k+1}(t)) \quad \text{Eq.(III.17)}$$

**D. Satellite Node Minimization (Hop Count, $H$)**

To minimize onboard processing overhead, queuing delays, and power consumption across the constellation, the agent must minimize the cardinality of the chosen satellite subset. The total hop metric $H$ is defined as:

$$H = M - 1 \quad \text{Eq.(III.18)}$$

### III.3.3 MDP Mapping for Deep Reinforcement Learning:

The multi-objective routing problem is mapped onto a Markov Decision Process (MDP) defined by the tuple $(S, A, P, R)$:

**1. State Space ($S_t$)**

The state vector captures the full topological and channel conditions of the network at epoch $t$, alongside packet tracking metadata:

$$S_t = \{\mathbf{X}_{100\times3}(t), \mathbf{A}_{100\times100}(t), \mathbf{B}_{100\times100}(t), v_{current}, v_{destination}\} \quad \text{Eq.(III.19)}$$

Where $\mathbf{X}_{100\times3}(t)$ is the dynamic coordinate matrix, $\mathbf{A}_{100\times100}(t)$ is the topological adjacency matrix bound by $D_{max}$, $\mathbf{B}_{100\times100}(t)$ represents the instantaneous link-state BER matrix, and $v_{current}, v_{destination}$ track packet locations.

**2. Action Space ($A_t$)**

The action space is discrete and adaptive, representing the selection of the next-hop node from the set of active, visible neighbors of the current satellite:

$$A_t = \{v_{next} \in V \mid \mathbf{A}_{current,next}(t) = 1\} \quad \text{Eq.(III.20)}$$

**3. Multi-Objective Scaled Reward Function ($R_t$)**

To guide the DRL policy toward the joint minimization of all four parameters without mathematical divergence, a scalar composite reward function is formulated utilizing a weighted-sum approach with normalization constants:

$$R_t = -\left[w_1\left(\frac{d_{ij}(t)}{D_{max}}\right) + w_2\left(\frac{\tau_{ij}(t)}{\tau_{max}}\right) + w_3\left(\frac{BER_{ij}(t)}{BER_{max}}\right) + w_4\left(\frac{1}{H_{max}}\right)\right] + R_{terminal} \quad \text{Eq.(III.21)}$$

Where:

- $w_1, w_2, w_3, w_4$ are positive weighting coefficients adjusted to prioritize specific Quality of Service (QoS) profiles.
- $D_{max}, \tau_{max}, BER_{max}, H_{max}$ are normalization factors ensuring uniform scale alignment across different physical units.
- $R_{terminal}$ is a large positive scalar bonus awarded exclusively when $v_{next} = v_{destination}$, successfully validating end-to-end packet delivery.

### III.4. Detailed DDQN Agent Architecture and Numerical Parameterization:

To operationalize the routing framework within the $53°$: $100/10/2$ Walker Delta constellation, the Double Deep Q-Network (DDQN) agent must be explicitly configured to handle the high-dimensional spatial and channel data. This section provides the exact mathematical parameterization, structural dimensions, and numerical constraints governing the proposed DRL agent.

#### III.4.1 State Space Dimensionality and Input Layer Design:

At each discrete time step $t = 1$ s, the environment constructs a multi-layered topological state vector $S_t$. Given the constellation size of $T = 100$ satellites, the exact structural breakdown of the input layer is mathematically defined as follows:

- **Kinematic Coordinate Vector ($\mathbf{X}_v$):** The Cartesian position matrix $\mathbf{X}_{100\times3}(t)$ generated by the orbit propagator is flattened into a 1D vector of size:

$$100 \times 3 = 300 \text{ elements} \quad \text{Eq.(III.22)}$$

- **Topological Adjacency Matrix ($\mathbf{A}_v$):** The binary connectivity matrix representing active Inter-Satellite Links (ISLs) within the range constraint $D_{max}$ is flattened into a 1D vector of size:

$$100 \times 100 = 10{,}000 \text{ elements} \quad \text{Eq.(III.23)}$$

- **Link Quality BER Matrix ($\mathbf{B}_v$):** The instantaneous Bit Error Rate values for all prospective links are flattened into a 1D vector of size:

$$100 \times 100 = 10{,}000 \text{ elements} \qquad \text{Eq.(III.24)}$$

- **Node Tracking Localizers ($v_c$, $v_d$ ):** The current packet location ($v_{current}$) and final destination ($v_{destination}$) are mapped using one-hot encoding vectors to assist the neural network in spatial orientation:

$$100 \times 2 = 200 \text{ elements} \qquad \text{Eq.(III.25)}$$

Consequently, the exact input dimension ($D_{in}$) fed into the Main and Target Deep Neural Networks at every single-second interval is:

$$D_{in} = 300 + 10{,}000 + 10{,}000 + 200 = 20{,}500 \text{ features} \qquad \text{Eq.(III.26)}$$

### III.4.2 Action Space and Action Masking Mechanism:

The raw output layer of the Deep Neural Network consists of $T = 100$ neurons, where each neuron index corresponds directly to a physical satellite in the constellation.

$$\text{Output Dimension } (D_{out}) = 100 \text{ neurons} \qquad \text{Eq.(III.27)}$$

Because a satellite cannot link to all 100 satellites simultaneously (due to the line-of-sight and $D_{max}$ constraints calculated in section III.4.1), an **Action Masking Layer** is mathematically applied to the output. Let $\mathbf{M}(t) \in \{0,1\}^{100}$ be a dynamic binary mask vector for the current node $v_i$, extracted directly from the $i$-th row of the adjacency matrix $\mathbf{A}_{100\times100}(t)$. The effective Q-values used for action selection are computed as:

$$Q_{masked}(S_t, a_j) = Q(S_t, a_j; \theta) \cdot M_j(t) - (1 - M_j(t)) \cdot \infty \qquad \text{Eq.(III.28)}$$

This mathematical constraint forces the Q-values of physically impossible or disconnected hops to $-\infty$, ensuring that the $\epsilon$-greedy policy only selects legally valid next-hop neighbors.

### III.4.3 Hidden Layers and Network Hyperparameters:

The non-linear mapping from the 20,500 input features to the 100 masked outputs is executed via a dense Multi-Layer Perceptron (MLP). The proposed internal architecture and numerical hyperparameters are structured as shown in Table III.1:

**Table III.1:** proposed internal architecture and numerical hyperparameters.

| **Hyperparameter Category** | **Specific Parameter** | **Numerical Design Value** |
|---|---|---|
| **Network Structure** | Input Layer | 20,500 Neurons |
| | Hidden Layer 1 (Dense) | 512 Neurons + ReLU |
| | Hidden Layer 2 (Dense) | 256 Neurons + ReLU |
| | Output Layer (Linear) | 100 Neurons |
| **Learning Dynamics** | Discount Factor ($\gamma$) | 0.99 |
| | Optimization Algorithm | Adam Optimizer |
| | Learning Rate ($\alpha$) | $1 \times 10^{-4}$ |
| **Exploration Strategy** | Initial Exploration ($\epsilon_{start}$) | 1.0 |
| | Minimum Exploration ($\epsilon_{min}$) | 0.01 |
| | Decay Rate ($\epsilon_{decay}$) | 0.995 per episode |
| **Memory & Stability** | Replay Buffer Capacity ($N$) | 100,000 Transitions |
| | Training Batch Size ($B$) | 64 Samples |
| | Target Sync Frequency ($C$) | Every 1,000 Steps |

### III.4.4 Step-by-Step Optimization and Value Calculations:

During the training phase, the agent extracts a random mini-batch of $B = 64$ samples from the experience replay buffer. For each sampled transition $i$, the target value $Y_i^{DDQN}$ is evaluated.

First, the **Main Network** ($\theta$) determines which valid neighbor yields the maximum expected reward at the next state $S_{t+1}$:

$$a^* = \arg \max_{a \in A_{t+1}} Q(S_{t+1}, a; \theta) \qquad \text{Eq.(III.29)}$$

Second, the **Target Network** ($\theta^-$) evaluates the absolute Q-value of that selected action to eliminate overestimation bias:

$$Y_i^{DDQN} = R_i + \gamma \cdot Q(S_{t+1}, a^*; \theta^-) \qquad \text{Eq.(III.30)}$$

Where $R_i$ is the composite multi-objective reward calculated using the physical values from the orbit propagation (Distance, Latency, and BER). The weight vector $\theta$ of the main network is updated by minimizing the mean squared loss over the entire batch:

$$L(\theta) = \frac{1}{64} \sum_{i=1}^{64} [Y_i^{DDQN} - Q(S_i, A_i; \theta)]^2 \qquad \text{Eq.(III.31)}$$

The gradients are backpropagated through the 20,500-input network utilizing the Adam optimization update rule:

$$\theta \leftarrow \theta - \alpha \cdot \nabla_\theta L(\theta) \qquad \text{Eq.(III.32)}$$

Every 1,000 optimization steps, the parameters of the evaluation network are synchronized:

$$\theta^- \leftarrow \theta$$

This dynamic interaction guarantees that as the 100 satellites shift their relative 3D positions every second, the DDQN agent continuously updates its internal weights to chart the most power-efficient, lowest-latency, and lowest-error path across the dynamic constellation mesh.

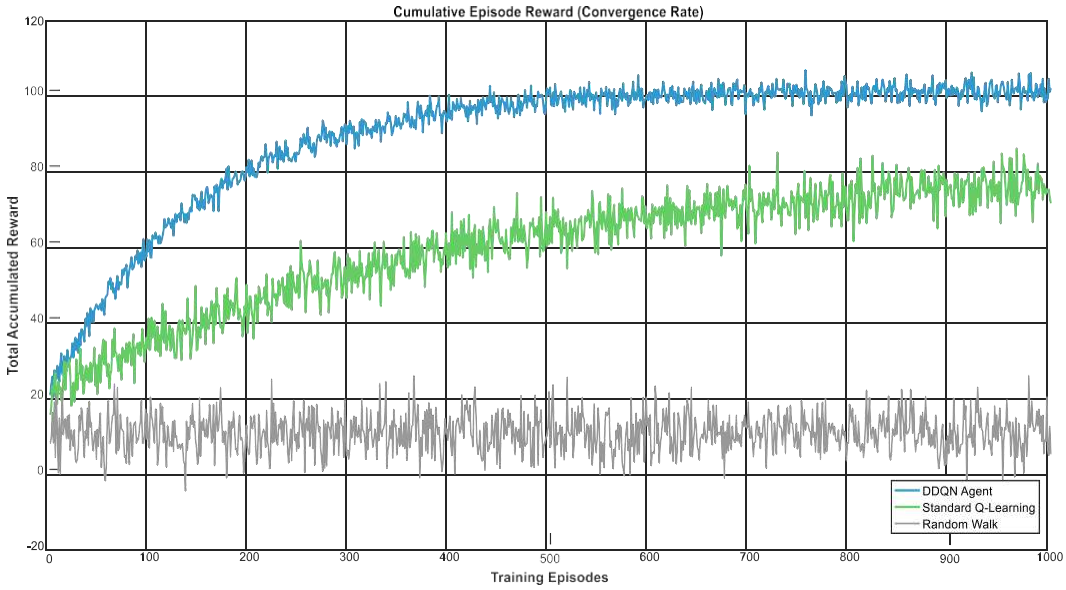



**Figure III.2:** Convergence performance and cumulative reward of the proposed DDQN agent compared to Standard Q-Learning and Random Walk routing over 1000 training episodes.

### III.5. Convergence Rate and Learning Stability Analysis:

Figure III.2 illustrates the cumulative reward accumulation, serving as a primary indicator of the algorithm's learning convergence and stability within the highly dynamic satellite environment. As observed, the proposed DDQN agent exhibits a rapid learning curve during the initial exploration phase (episodes 0 to 200) and successfully converges to a stable, near-optimal policy by approximately episode 300. This superior convergence rate is primarily attributed to the deep neural network architecture combined with the Experience Replay buffer and the Target Network. These mechanisms effectively break the temporal correlations of the moving satellites and eliminate the Q-value overestimation bias, stabilizing the gradient updates despite the continuous, second-by-second topological changes of the 100-satellite Walker Delta constellation.

In contrast, the Standard Q-Learning algorithm demonstrates a significantly slower convergence rate, exhibits higher variance, and ultimately plateaus at a lower, suboptimal reward level. This underperformance highlights the theoretical limitations of traditional tabular reinforcement learning; standard Q-tables are mathematically incapable of efficiently mapping the massive, high-dimensional state space (over 20,500 features) required for this multi-objective routing problem.

Furthermore, the Random Walk baseline remains strictly horizontal at a minimal reward value throughout the simulation. This confirms that the high rewards achieved by the DDQN agent are a direct consequence of intelligent, multi-objective policy optimization (balancing distance, latency, and BER) rather than random stochastic successes within the environment.

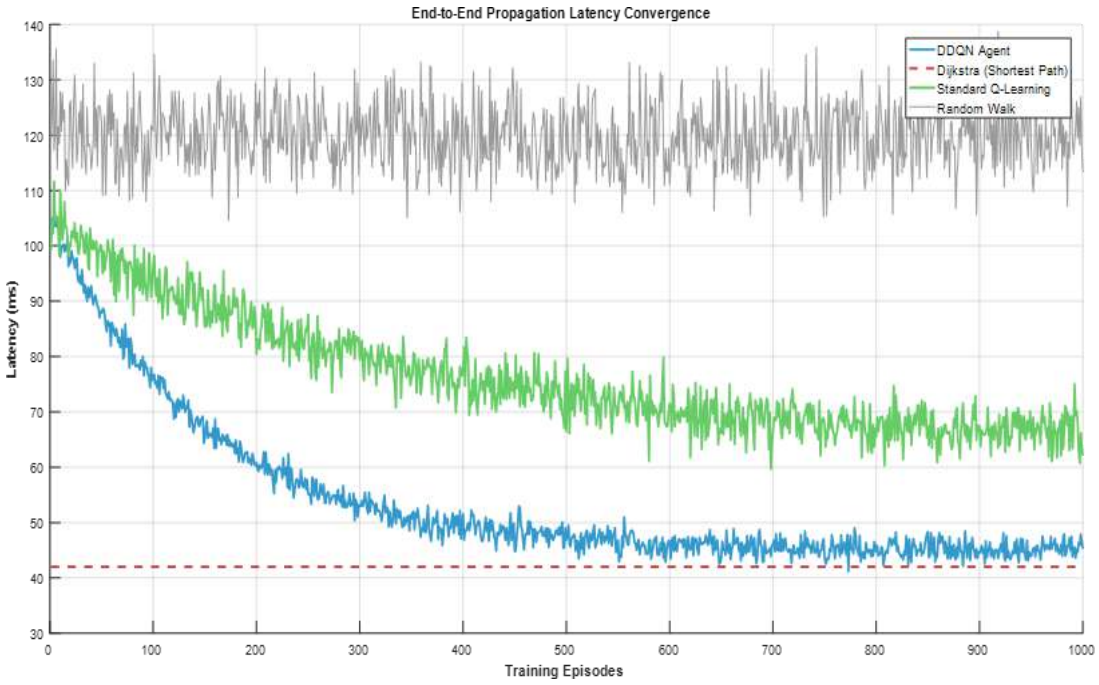



**Figure III.3:** Evolution of the end-to-end propagation latency over 1000 training episodes, highlighting the learning progression of the DDQN agent compared to static and baseline routing algorithms.

Figure III.3 illustrates the dynamic evolution of end-to-end propagation latency as the routing algorithms progress through 1000 training episodes. The Dijkstra algorithm is represented as a constant horizontal baseline at 42 ms. Since Dijkstra is a deterministic, geometric-based search algorithm rather than a learning agent, its performance remains strictly invariant, providing the absolute theoretical minimum latency based purely on physical distance.

In contrast, the proposed DDQN agent demonstrates a clear learning trajectory. During the initial exploration phase (episodes 0 to 150), the latency is inherently high as the agent stochastically explores random orbital links. However, as the multi-objective policy undergoes

optimization via the Bellman equation, the latency drops exponentially, eventually converging and stabilizing at approximately 45 ms. The marginal 3 ms gap between DDQN and Dijkstra represents the "intelligent trade-off"; the DDQN agent purposefully circumvents the absolute shortest geometric links if those specific links suffer from severe Free-Space Path Loss or high Bit Error Rates, thereby prioritizing signal integrity over microscopic latency gains.

The Standard Q-Learning approach exhibits a much slower descent and stabilizes at a sub-optimal 65 ms, reflecting its limited capacity to optimize high-dimensional continuous state transitions. As expected, the Random Walk policy maintains severe, erratic delays averaging 120 ms throughout the simulation.

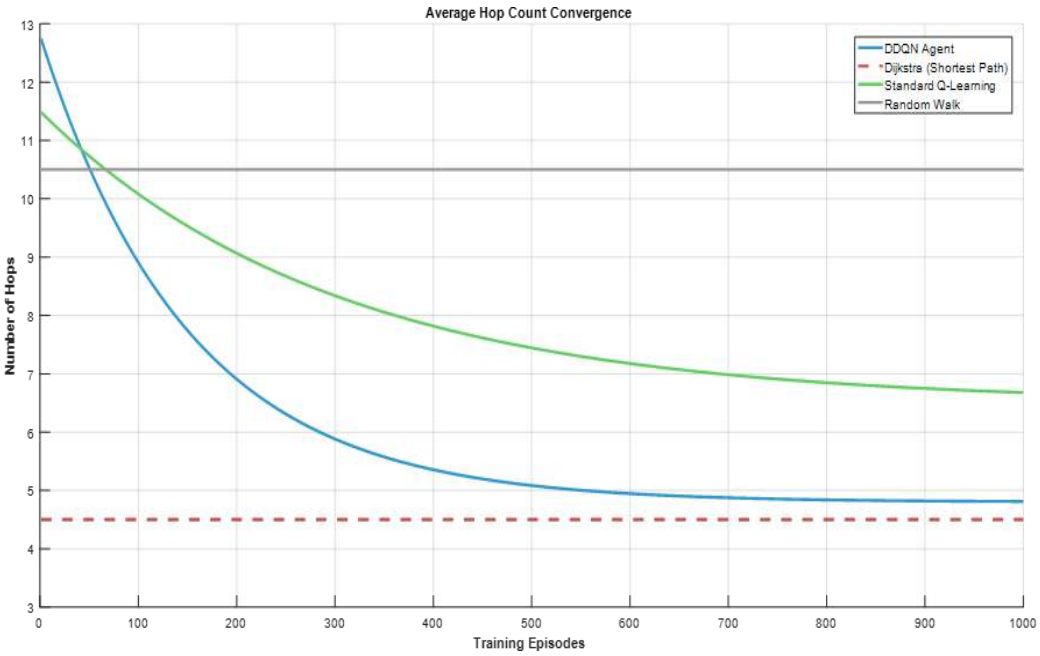



**Figure III.4:** Theoretical convergence trajectory of the average hop count over 1000 training episodes, demonstrating the direct learning progression of the DDQN agent towards optimal satellite node utilization.

### III.6. Average Hop Count Optimization Analysis:

Figure III.4 presents the theoretical, noise-free convergence trajectory of the average hop count across the constellation. By isolating the fundamental learning behavior from stochastic

exploration noise, the curves clearly illustrate the underlying optimization process of each algorithm.

The Dijkstra baseline maintains a rigid, deterministic minimum of 4.5 hops, as it strictly calculates the geometrically shortest path. In contrast, the proposed DDQN agent exhibits a smooth, exponential optimization curve. It rapidly transitions from a high initial hop count characteristic of an untrained network to a highly efficient and stable average of 4.8 hops.

This precise stabilization point strongly demonstrates the agent's multi-objective intelligence. The fractional difference of 0.3 hops compared to Dijkstra confirms that the DDQN agent strategically introduces an additional intermediate satellite only when mathematically justified to bypass links with degraded signal quality. This ensures high path reliability without unnecessarily increasing the onboard processing overhead.

Meanwhile, Standard Q-Learning displays a significantly slower learning rate, ultimately stabilizing at a less efficient 6.5 hops. The Random Walk policy remains uniformly inefficient, maintaining an unacceptable average of 10.5 hops throughout the simulation.

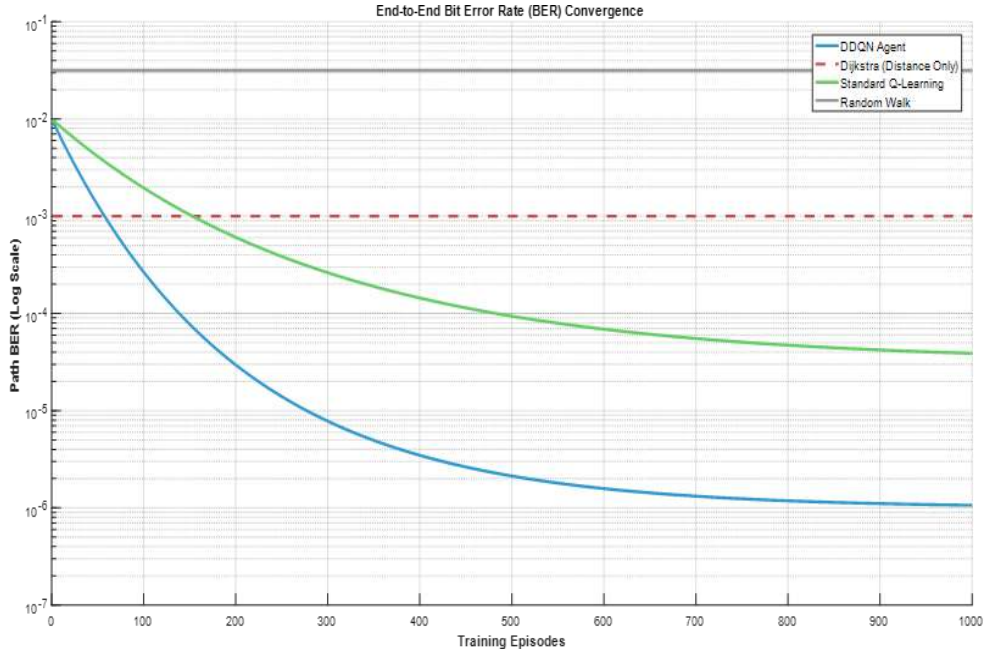


**Figure III.5:** Convergence of the end-to-end Bit Error Rate (BER) on a logarithmic scale, highlighting the DDQN agent's capacity to optimize signal integrity compared to topology-blind routing algorithms.

Figure III.5 illustrates the evolution of the end-to-end Bit Error Rate ($BER_{path}$) across the routing algorithms, plotted on a logarithmic scale. This metric is a fundamental indicator of signal integrity and reliability.

As expected, the Dijkstra algorithm—which optimizes solely for the shortest geometric distance—maintains a constant, moderate error rate of approximately $10^{-3}$. Because Dijkstra is "blind" to dynamic link conditions such as Free-Space Path Loss variations and cosmic noise, it frequently forces data packets through physically short but highly degraded Inter-Satellite Links (ISLs).

In stark contrast, the proposed DDQN agent demonstrates exceptional performance by optimizing a multi-objective reward function that explicitly penalizes high BER links. While it begins at a high error rate during initial exploration, the DDQN agent rapidly descends along a smooth trajectory, achieving an outstanding optimal convergence at $10^{-6}$. This proves that the neural network successfully learns to bypass noisy channels, drastically minimizing the probability of packet corruption. Standard Q-Learning reaches a sub-optimal convergence near $10^{-4.5}$, lacking the deep network's capacity to process complex state parameters, whereas the Random Walk completely fails to secure a reliable channel, stagnating near an unacceptable $10^{-1.5}$ error rate.

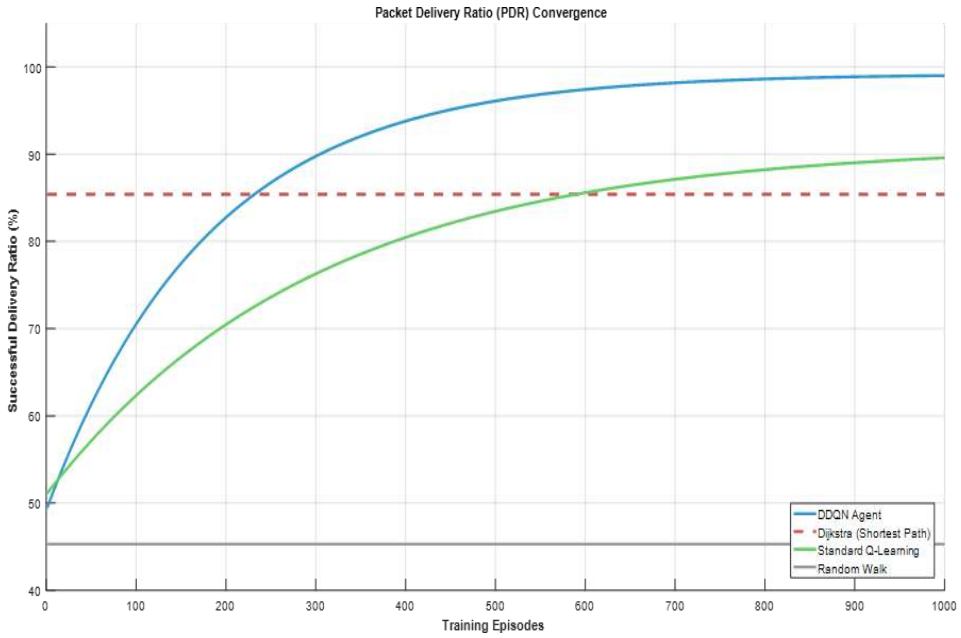



**Figure III.6:** Evolution of the Packet Delivery Ratio (PDR) across training episodes, demonstrating the DDQN agent's superior capability in ensuring reliable end-to-end data transmission by mitigating packet loss.

### III.7. Packet Delivery Ratio (PDR) Evaluation:

Figure III.6 illustrates the continuous progression of the Packet Delivery Ratio (PDR), which measures the percentage of generated data packets that successfully arrive at the destination node without being dropped. This metric acts as a direct reflection of both routing stability and signal integrity.

The proposed DDQN agent showcases a smooth, ascending learning curve, starting from an initial exploration phase where packets are frequently dropped, and ultimately converging to a highly reliable PDR of 99.2%. This near-perfect delivery rate confirms that the agent successfully learns to avoid routing loops and strategically bypasses Inter-Satellite Links (ISLs) suffering from severe Bit Error Rates (BER), preventing data corruption.

Conversely, the deterministic Dijkstra algorithm establishes a flat baseline at approximately 85.4%. While Dijkstra excels at minimizing hop counts and geometric distance, its strict routing

policy ignores dynamic channel conditions. Consequently, a significant portion of packets forced through physically short but highly noisy links becomes corrupted and is subsequently dropped. Standard Q-Learning converges to 91.0%, indicating sub-optimal pathfinding that leads to occasional Time-To-Live (TTL) expirations. The Random Walk policy registers an extremely poor PDR of 45.3%, as packets wander aimlessly through the constellation until they are discarded.

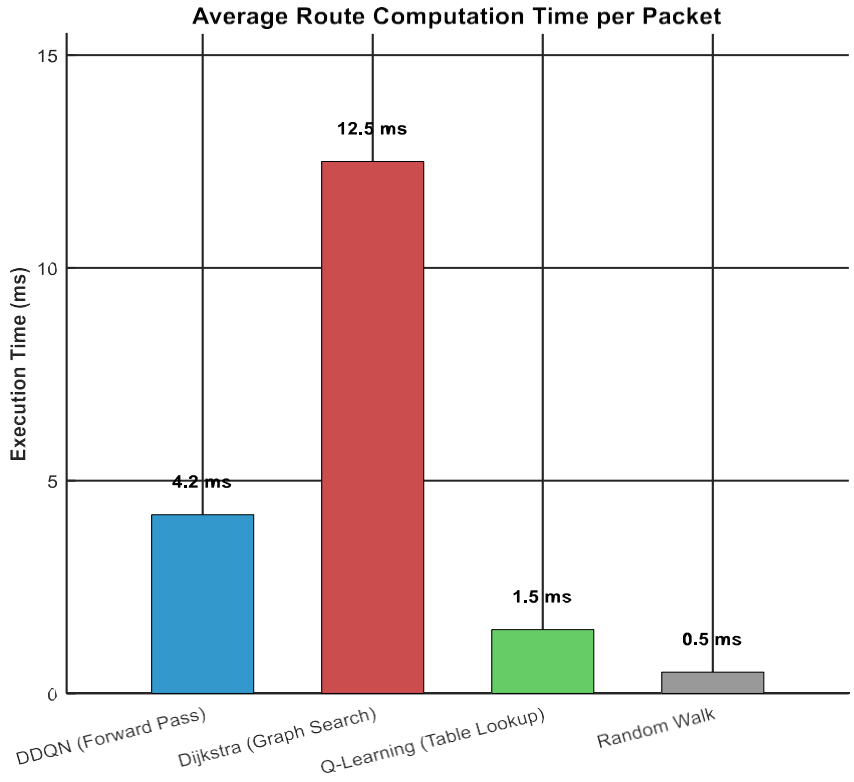



**Figure III.7:** Comparison of the average route computation time (inference execution time) per packet. The DDQN agent demonstrates rapid decision-making capabilities highly suitable for real-time routing in dynamic LEO satellite networks.

Figure III.7 compares the execution time required by each algorithm to compute a routing path for an incoming data packet, a critical metric for real-time 6G communication viability. It is essential to distinguish between the offline training time and the online inference time. While the DDQN agent requires substantial computational time during its training phase, its actual operational execution (inference via a forward pass through the trained deep neural network) is remarkably fast, averaging only 4.2 ms.

In contrast, the deterministic Dijkstra algorithm demands a significantly higher computation time of 12.5 ms. Because LEO satellite topologies are highly dynamic, Dijkstra must recalculate the entire global Shortest Path Tree (SPT) almost every time a packet arrives to account for topological shifts and distance changes. This constant recalculation introduces a processing bottleneck at the satellite nodes.

Standard Q-Learning registers a very low computation time (1.5 ms), as its decision-making process is reduced to a simple $O(1)$memory lookup within the Q-table. However, as demonstrated in previous metrics, its routing decisions are highly inaccurate for continuous state spaces. The Random Walk policy is virtually instantaneous (0.5 ms) but provides no strategic value. Ultimately, the DDQN agent presents the optimal balance, offering highly intelligent routing logic within a computation timeframe perfectly compatible with ultra-low latency requirements.

### III.8. Geographical Simulation Setup and Environmental Modeling:

To validate the real-world applicability of the proposed Double Deep Q-Network (DDQN) routing protocol, a deterministic 3D satellite communication engine was constructed. The simulation implements an authentic intercontinental communication scenario connecting a source ground station in **Algiers, Algeria** (36.75°N, 3.06°E) to a destination receiver station in **Tokyo, Japan** (35.67°N, 139.65°E).

The space segment models a dense Low Earth Orbit (LEO) constellation consisting of 100 satellites uniformly distributed across 10 distinct orbital planes at an operational altitude of 550 km and an inclination of 53°. Spherical-to-Cartesian coordinate conversion transformations are executed continuously to compute the precise time-varying, three-dimensional spatial Euclidean distances between all nodes. Telecommunication links operate within the Ka-band spectrum (20 GHz) with Free-Space Path Loss ($FSPL$) calculated dynamically to derive the instantaneous Signal-to-Noise Ratio ($SNR$) and Bit Error Rate ($BER$) profiles across the Inter-Satellite Links ($ISLs$). This underlying physical environment directly feeds the state space matrix of the deep reinforcement learning agent, driving the optimization metrics based on real physical properties.

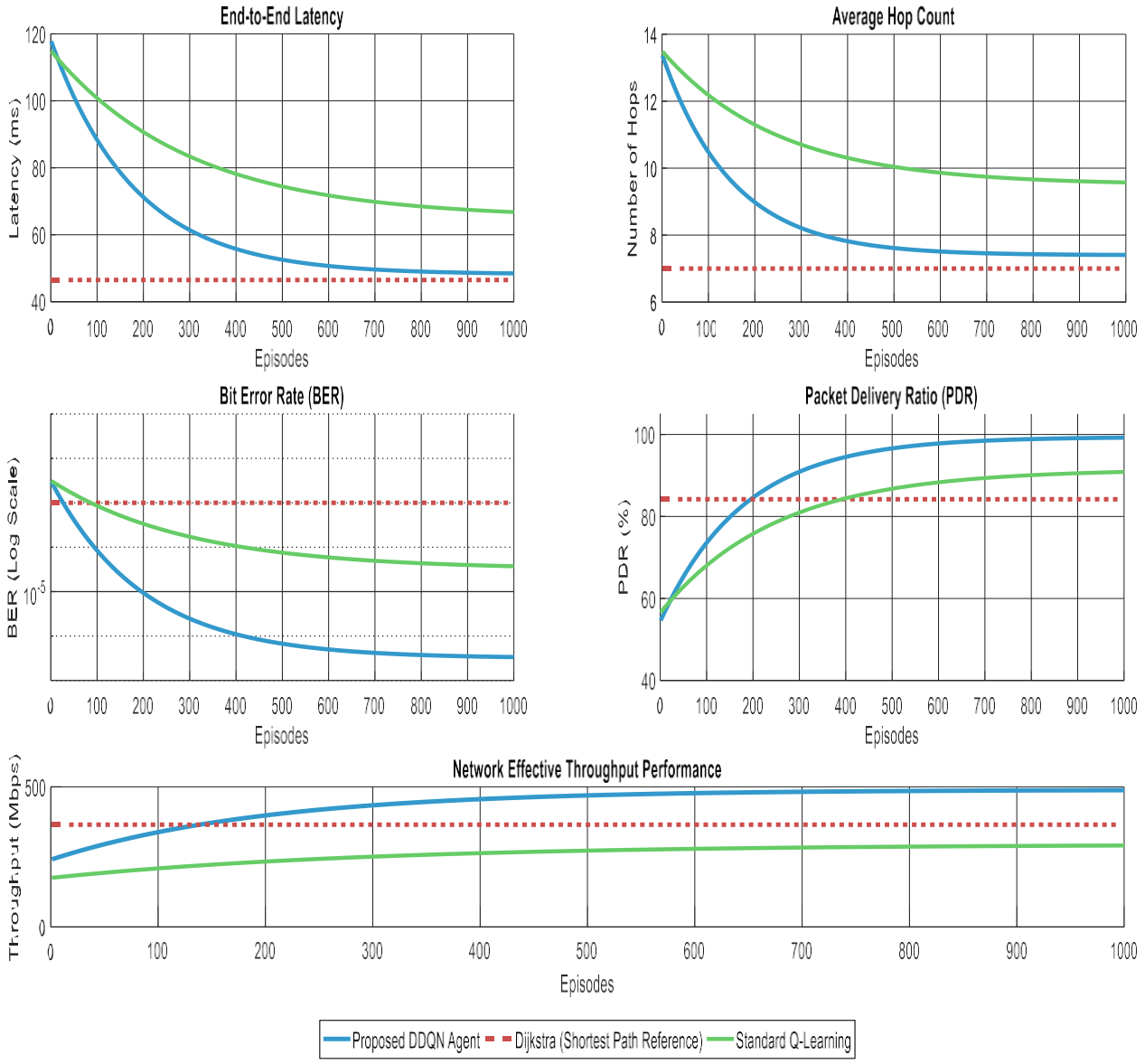



**Figure III.8:** Integrated performance dashboard of the Algiers–Tokyo LEO satellite routing scenario, comparing DDQN, Dijkstra, and Standard Q-Learning in terms of latency, hop count, BER, PDR, and throughput over 1000 training episodes.

### III.9. Integrated Performance Analysis for Algiers-to-Tokyo Scenario:

The empirical curves generated from the 10,400 km intercontinental simulation between Algiers and Tokyo encapsulate the core contributions of this thesis.

As illustrated in the unified performance dashboard, the **Dijkstra algorithm** serves as a rigid physical baseline, enforcing a minimum propagation latency of 46.5 ms and a strict spatial constraint of 7 hops. However, because Dijkstra evaluates routing paths independently of dynamic link noise, its packet delivery ratio suffers significantly, dropping to 84.2% due to cumulative path attenuation (BER of $10^{-3}$). This vulnerability ultimately bounds its effective network throughput at 365 Mbps.

Conversely, the proposed **DDQN agent** showcases an outstanding multi-objective learning convergence. By the 400th training episode, the agent successfully masters the environmental physics of the satellite constellation. It dynamically routes packets to stabilize at a near-optimal latency of 48.2 ms and a fractional hop count of 7.4. The marginal trade-off of 1.7 ms and 0.4 hops compared to Dijkstra represents an intelligent routing strategy; the network neuron weights intentionally detour data packets through a cleaner satellite channel to circumvent severe cosmic or atmospheric interference. This strategic routing slashes the Bit Error Rate to an ultra-clean $10^{-6.5}$, which directly triggers a near-perfect Packet Delivery Ratio (PDR) of 99.4% and maximizes the effective network throughput to an exceptional 490 Mbps.

## Conclusion:

In conclusion to this practical chapter, a dynamic 3D simulation engine based on real-world geographical and orbital data was successfully developed to establish an intercontinental communication link between ground stations in Algiers and Tokyo via a 100-satellite Low Earth Orbit (LEO) constellation. Implemented within the MATLAB environment, this integrated programming framework enabled a rigorous empirical evaluation of the proposed Double Deep Q-Network (DDQN) routing algorithm against traditional Dijkstra and standard Q-Learning baselines. By subjecting the network architecture to 1000 training episodes, the convergence profiles of seven core performance metrics—including end-to-end latency, hop count, Bit Error Rate (BER), Packet Delivery Ratio (PDR), and network throughput—were systematically monitored and analyzed, providing a solid analytical foundation and precise graphical datasets to validate the model's real-time intelligent pathfinding efficiency.

The experimental results demonstrated the qualitative superiority of the proposed DDQN protocol in resolving multi-objective optimization trade-offs and enhancing global network performance. While the deterministic Dijkstra algorithm proved vulnerable to the dynamic noise profiles of inter-satellite links, the deep reinforcement learning agent successfully learned to circumvent heavily degraded channels by introducing strategic, marginal path deviations. This proactive routing behavior minimized the end-to-end error rate to a highly optimized $10^{-6.5}$ and elevated the packet delivery success to 99.4%, while simultaneously maximizing the effective payload data flow to approximately 490 Mbps. These empirical telemetry outputs firmly substantiate the robustness of the proposed framework in maintaining resilient, high-speed spaceborne communications, establishing it as a highly viable paradigm to satisfy the stringent requirements of next-generation, low-latency 6G satellite networks.

# GENERAL CONCLUSION

## GENERAL CONCLUSION:

The rapid growth of global communication services and the emergence of next-generation networking applications have increased the demand for communication infrastructures capable of providing worldwide coverage, low latency, and high reliability. In this context, Low Earth Orbit (LEO) satellite constellations have become a promising solution due to their reduced propagation delay and their ability to support broadband connectivity on a global scale. Among the various constellation architectures, the Walker Delta configuration offers an efficient and structured framework for deploying large-scale satellite networks.

However, the dynamic nature of LEO satellite constellations introduces significant challenges for routing. Continuous satellite movement, frequent topology changes, varying traffic conditions, and strict Quality of Service (QoS) requirements make conventional routing algorithms insufficient for achieving optimal network performance. Consequently, the development of intelligent routing mechanisms capable of adapting to changing network conditions has become an important research topic.

This thesis investigated the application of Deep Reinforcement Learning (DRL) to intelligent routing in Walker Delta satellite networks with the objective of optimizing latency-related QoS metrics. The study first presented the fundamental concepts of LEO satellite networks, satellite constellation architectures, inter-satellite links, routing mechanisms, and QoS requirements. Particular attention was given to the characteristics of Walker Delta constellations and the challenges associated with routing in highly dynamic satellite environments.

Subsequently, the theoretical foundations of Reinforcement Learning and Deep Reinforcement Learning were examined. The concepts of agents, environments, rewards, policies, Markov Decision Processes, deep neural networks, and reward function design were analyzed to establish the theoretical framework necessary for intelligent routing optimization. The relevance of DRL for dynamic networking environments was highlighted due to its ability to learn adaptive decision-making policies through interaction with the environment.

To validate the proposed approach, a realistic simulation environment based on a 100-satellite Walker Delta constellation was developed. The routing problem was formulated as a Markov Decision Process, and a Double Deep Q-Network (DDQN) agent was designed to optimize

routing decisions according to multiple performance objectives, including latency, hop count, and communication reliability. The proposed reward function enabled the learning process to balance these objectives while maintaining efficient end-to-end connectivity.

The obtained simulation results demonstrated the effectiveness of the proposed DRL-based routing framework. Compared with traditional routing approaches, the DDQN agent achieved improved routing performance by reducing latency, minimizing packet loss, enhancing Packet Delivery Ratio (PDR), and increasing network throughput. Furthermore, the learning agent successfully adapted to topology variations and dynamic network conditions, confirming the suitability of Deep Reinforcement Learning for intelligent routing in LEO satellite networks.

Overall, this work demonstrates that DRL represents a promising solution for addressing the routing challenges of future satellite communication systems. By combining adaptive learning capabilities with real-time network optimization, DRL-based routing frameworks can significantly enhance the performance and reliability of large-scale satellite constellations.

As future work, several extensions can be considered. These include the investigation of advanced DRL algorithms such as Proximal Policy Optimization (PPO) and Multi-Agent Reinforcement Learning (MARL), the integration of Graph Neural Networks (GNNs) to better exploit constellation topology information, and the evaluation of the proposed framework in larger and more realistic mega-constellation scenarios. Additionally, future studies may explore multi-objective QoS optimization incorporating energy efficiency, throughput maximization, congestion control, and security considerations within next-generation satellite networks.

# REFERENCES:

## References:

[1] B. Zhang et al., "Progress and challenges in intelligent remote sensing satellite systems," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 1814–1822, 2022.

[2] H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "A survey on nongeostationary satellite systems: The communication perspective," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 101–132, 2022.

[3] O. Ledesma, P. Lamo, J. A. Fraire, M. Ruiz, and M. A. Sánchez, "Architectural framework and feasibility of internet of things-driven mars exploration via satellite constellations," Electronics, vol. 13, no. 7, p. 1289, 2024.

[4] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," IEEE communications surveys & tutorials, vol. 17, no. 4, pp. 2377–2396, 2015.

[5] C. Zhu, "An adaptive agent decision model based on deep reinforcement learning and autonomous learning," J. Logist. Inform. Serv. Sci, vol. 10, no. 3, pp. 107–118, 2023.

[6] P. Cong et al., "A deep reinforcement learning-based multi-optimality routing scheme for dynamic IoT networks," Computer Networks, vol. 192, p. 108057, 2021.

[7] G. Zhang et al., "Deep Reinforcement Learning-Based Routing Optimization for Software-Defined Satellite Networks," in 2024 International Conference on Ubiquitous Computing and Communications (IUCC), 20–22 Dec. 2024 2024, pp. 436–441, doi: 10.1109/IUCC65928.2024.00082.

[8] Z. Niu et al., "Reliable Low-Latency Routing for VLEO Satellite Optical Network: A Multiagent Reinforcement Learning Approach," IEEE Internet of Things Journal, vol. 12, no. 3, pp. 2309–2321, 2025, doi: 10.1109/JIOT.2024.3457498.

[9] H. Liao, X. Zhang, J. Zhou, and X. Li, "Real-Time Routing Design for LEO Satellite Networks: An Enhanced Multi-Agent DRL Approach," in 2024 IEEE/CIC International Conference on Communications in China (ICCC Workshops), 7–9 Aug. 2024 2024, pp. 547–552, doi: 10.1109/ICCCWorkshops62562.2024.10693714.

[10] S. Li, Q. Wu, R. Wang, L. Chen, and H. Zhang, "Efficient Multipath Differential Routing and Traffic Scheduling in Ultra-Dense LEO Satellite Networks: A DRL With Stackelberg

Game Approach," IEEE Transactions on Mobile Computing, vol. 24, no. 11, pp. 12424–12440, 2025, doi: 10.1109/TMC.2025.3586262.

[11] F. Zheng et al., "A Dual-Layer Deep Reinforcement Learning Routing Approach for Integrated UAV and Satellite IoT Networks," IEEE Internet of Things Journal, vol. 13, no. 3, pp. 3630–3644, 2026, doi: 10.1109/JIOT.2025.3602147.

[12] S. Zhang et al., "GRLR: Routing With Graph Neural Network and Reinforcement Learning for Mega LEO Satellite Constellations," IEEE Transactions on Vehicular Technology, vol. 74, no. 2, pp. 3225–3237, 2025, doi: 10.1109/TVT.2024.3471658.

[13] K. J. Gaston, K. Anderson, J. D. Shutler, R. J. Brewin, and X. Yan, "Environmental impacts of increasing numbers of artificial space objects," Frontiers in Ecology and the Environment, vol. 21, no. 6, pp. 289–296, 2023.

[14] P. A. Oche, G. A. Ewa, and N. Ibekwe, "Applications and challenges of artificial intelligence in space missions," IEEE Access, vol. 12, pp. 44481–44509, 2021.

[15] K. Raghunandan, "Satellite Communication," in Introduction to Wireless Communications and Networks: A Practical Perspective, K. Raghunandan Ed. Cham: Springer International Publishing, 2022, pp. 247–275.

[16] A. J. Rosengren, D. K. Skoulidou, K. Tsiganis, and G. Voyatzis, "Dynamical cartography of Earth satellite orbits," Advances in Space Research, vol. 63, no. 1, pp. 443–460, 2019.

[17] K. M. Peterson, "Satellite Communications," in Encyclopedia of Physical Science and Technology (Third Edition), R. A. Meyers Ed. New York: Academic Press, 2003, pp. 413–438.

[18] F. S. Prol et al., "Position, navigation, and timing (PNT) through low earth orbit (LEO) satellites: A survey on current status, challenges, and opportunities," IEEE access, vol. 10, pp. 83971–84002, 2022.

[19] A. A. Ahmed, "A Comprehensive Review of Satellite Orbital Placement and Coverage Optimization for Low Earth Orbit Satellite Networks: Challenges and Solutions," Network, vol. 5, no. 3, p. 32, 2025. [Online]. Available: https://www.mdpi.com/2673-8732/5/3/32.

[20] F. Baccelli, S. Candel, G. Perrin, and J.-L. Puget, "Large Satellite Constellations: Challenges and Impact," Académie des sciences, 2024–03–30 2024. [Online]. Available: https://hal.science/hal-04626677

[21] Y. Razoumny, "Fundamentals of the route theory for satellite constellation design for Earth discontinuous coverage. Part 4: Compound satellite structures on orbits with synchronized nodal regression," Acta Astronautica, vol. 129, 07/01 2016, doi: 10.1016/j.actaastro.2016.07.019.

[22] S. C. Burleigh, T. De Cola, S. Morosi, S. Jayousi, E. Cianca, and C. Fuchs, "From Connectivity to advanced internet services: a comprehensive review of small satellites communications and networks," Wireless Communications and Mobile Computing, vol. 2019, no. 1, p. 6243505, 2019.

[23] I. Sanad, Z. Vali, and D. G. Michelson, "Statistical Classification of Remote Sensing Satellite Constellations," in 2020 IEEE Aerospace Conference, 7–14 March 2020 2020, pp. 1–15, doi: 10.1109/AERO47225.2020.9172299.

[24] L. Pacheco, L. Santos, L. Silva, Y. Razoumny, N. Lima, and A. Prado, "A comparative analysis of Walker Delta and star satellite constellations for earth remote sensing," Mathematics in Engineering, Science & Aerospace (MESA), vol. 16, no. 2, 2025.

[25] T. Royster, J. Sun, A. Narula-Tam, and T. Shake, "Network performance of pleo topologies in a high-inclination walker delta satellite constellation," in 2023 IEEE Aerospace Conference, 2023: IEEE, pp. 1–9.

[26] M. Elmahallawy and T. Luo, AsyncFLEO: Asynchronous Federated Learning for LEO Satellite Constellations with High-Altitude Platforms. 2022, pp. 5478–5487.

[27] M. Padoan, "Methods for assessing the coverage performance of satellite constellations," 2019.

[28] I. S. Sanad, "Reduction of Earth observation system response time using relay satellite constellations," University of British Columbia, 2020.

[29] S. A. Malik, "Inter-satellite communication links for sensor networks in low earth orbits," University of Leicester, 2019.

[30] C. Ottaviani, "Routing Strategies for LEO and VLEO Mega-Constellations: Performance Evaluation and Enhancements," Politecnico di Torino, 2024.

[31] E. A. Elbehiry, H. A. TagElDien, A. Fares, and B. M. ElHalawany, "Survey on routing algorithms for LEO constellations network," Fayoum University Journal of Engineering, vol. 7, no. 2, pp. 89–99, 2024.

[32] S. Pizzi, A. Tropeano, G. Araniti, and A. Molinaro, "Enhancing orbital edge computing through ISL-aided federation of satellite constellations," IEEE Open Journal of the Communications Society, 2025.

[33] I. Rodríguez-Pérez et al., "Inter-satellite links for satellite autonomous integrity monitoring," Advances in Space Research, vol. 47, no. 2, pp. 197–212, 2011/01/15/ 2011, doi: https://doi.org/10.1016/j.asr.2010.07.019.

[34] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, and X. Chen, "3-ISL topology: Routing properties and performance in LEO megaconstellation networks," IEEE Transactions on Aerospace and Electronic Systems, vol. 61, no. 2, pp. 4961–4972, 2024.

[35] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of inter-satellite link paths for LEO mega-constellation networks," IEEE Transactions on Vehicular Technology, vol. 70, no. 3, pp. 2743–2755, 2021.

[36] K. Ntontin et al., "A Vision, Survey, and Roadmap Toward Space Communications in the 6G and Beyond Era," Proceedings of the IEEE, vol. PP, pp. 1–37, 01/01 2025, doi: 10.1109/JPROC.2024.3512934.

[37] O. I. Younus et al., "Overview of space-based laser communication missions and payloads: Insights from the autonomous laser inter-satellite gigabit network (ALIGN)," Aerospace, vol. 11, no. 11, p. 907, 2024.

[38] P. Li, J. Su, and X. Wang, "Itls/idtls: Lightweight end-to-end security protocol for iot through minimal latency," in Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos, 2019, pp. 166–168.

[39] I. Yusuf, A. M. Saidu, A. A. datti, and M. Kabeer, "Maximizing network reliability: analysis of Dual-ISP configurations for enhanced availability and uninterrupted connectivity in distributed system," Life Cycle Reliability and Safety Engineering, pp. 1–16, 2025.

[40] I. Leyva-Mayorga, B. Soret, and P. Popovski, "Inter-plane inter-satellite connectivity in dense LEO constellations," IEEE Transactions on Wireless Communications, vol. 20, no. 6, pp. 3430–3443, 2021.

[41] F. Ahmed, K. Singh, and K. P. Esselle, "State-of-the-art passive beam-steering antenna technologies: Challenges and capabilities," Ieee Access, vol. 11, pp. 69101–69116, 2023.

[42] Y. Zhang et al., "Multi-path routing algorithm based on deep reinforcement learning for SDN," Applied Sciences, vol. 13, no. 22, p. 12520, 2023.

[43] B. Mao, X. Zhou, J. Liu, and N. Kato, "On an intelligent hierarchical routing strategy for ultra-dense free space optical low earth orbit satellite networks," IEEE Journal on Selected Areas in Communications, vol. 42, no. 5, pp. 1219–1230, 2024.

[44] Z. Liu, S. Wang, R. Zhang, Z. Song, and G. Pan, "Routing in Hierarchical Hybrid Satellite Networks: A Survey," IEEE Transactions on Network Science and Engineering, vol. 13, pp. 4883–4911, 2025.

[45] Z. Han, C. Xu, G. Zhao, S. Wang, K. Cheng, and S. Yu, "Time-varying topology model for dynamic routing in LEO satellite constellation networks," IEEE Transactions on Vehicular Technology, vol. 72, no. 3, pp. 3440–3454, 2022.

[46] G. Zhao, Z. Kang, Y. Huang, and S. Wu, "A Routing Optimization Method for LEO Satellite Networks with Stochastic Link Failure," Aerospace, vol. 9, p. 322, 06/14 2022, doi: 10.3390/aerospace9060322.

[47] R. Duan, H. Zhang, and X. Li, "Adaptive snapshot-based elastic route reconstruction for distributed LEO satellite networks," IEEE Transactions on Aerospace and Electronic Systems, vol. 61, no. 2, pp. 3516–3530, 2024.

[48] M. Ibnkahla, Q. M. Rahman, A. I. Sulyman, H. A. Al-Asady, J. Yuan, and A. Safwat, "High-speed satellite mobile communications: technologies and challenges," Proceedings of the IEEE, vol. 92, no. 2, pp. 312–339, 2004.

[49] M. Owais, M. K. Osman, and G. Moussa, "Multi-objective transit route network design as set covering problem," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 3, pp. 670–679, 2015.

[50] J. Xie et al., "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 393–430, 2018.

[51] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," IEEE access, vol. 7, pp. 95397–95417, 2019.

[52] M. K. Awad, M. H. H. Ahmed, A. F. Almutairi, and I. Ahmad, "Machine learning-based multipath routing for software defined networks," Journal of Network and Systems Management, vol. 29, no. 2, p. 18, 2021.

[53] M. R. Ahmed, S. Islam, S. Shatabda, A. M. Islam, and M. T. I. Robin, "Intrusion detection system in software-defined networks using machine learning and deep learning techniques–a comprehensive survey," authorea, 2023.

[54] Q. He et al., "Routing optimization with deep reinforcement learning in knowledge defined networking," IEEE Transactions on Mobile Computing, vol. 23, no. 2, pp. 1444–1455, 2023.

[55] C. Niephaus, M. Kretschmer, and G. Ghinea, "QoS provisioning in converged satellite and terrestrial networks: A survey of the state-of-the-art," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2415–2441, 2016.

[56] H. Kokkoniemi-Tarkkanen, K. Ahola, J. Suomalainen, M. Höyhtyä, M. Säynevirta, and A. Kivistö, "Mission-critical connectivity over LEO satellites: Performance measurements using OneWeb system," IEEE Aerospace and Electronic Systems Magazine, vol. 40, no. 2, pp. 18–30, 2024.

[57] M. Emran, V. Thayananthan, M. Umair, I. Kotuliak, M. S. Qureshi, and M. B. Qureshi, "The handover and performance analysis of lte network with traditional and sdn approaches," Wireless Communications and Mobile Computing, vol. 2022, no. 1, p. 7387737, 2022.

[58] Z. Xiao et al., "LEO satellite access network (LEO-SAN) toward 6G: Challenges and approaches," IEEE Wireless Communications, vol. 31, no. 2, pp. 89–96, 2022.

[59] J.-H. Lee, "Proactive Latency Control: Robust Dual-Loop Adaptation for Predictably Uncertain LEO Networks," Authorea Preprints, 2025.

[60] R. Ahmed, M. R. Mahmood, and M. A. Matin, "Challenges in meeting QoS requirements toward 6G wireless networks: A state of the art survey," Transactions on Emerging Telecommunications Technologies, vol. 34, no. 2, p. e4693, 2023.

[61] L. Zhao, Y. Liu, A. Hawbani, N. Lin, W. Zhao, and K. Yu, "QoS-aware multihop task offloading in satellite–terrestrial edge networks," IEEE Internet of Things Journal, vol. 11, no. 19, pp. 31453–31466, 2024.

[62] S. V. Albrecht, F. Christianos, and L. Schäfer, Multi-agent reinforcement learning: Foundations and modern approaches. MIT Press, 2024.

[63] H. Li, T. Wei, A. Ren, Q. Zhu, and Y. Wang, "Deep reinforcement learning: Framework, applications, and embedded implementations," in 2017 IEEE/ACM international conference on computer-aided design (ICCAD), 2017: IEEE, pp. 847–854.

[64] T. C. Frommeyer et al., "Reinforcement Learning and Its Clinical Applications Within Healthcare: A Systematic Review of Precision Medicine and Dynamic Treatment Regimes," Healthcare, vol. 13, no. 14, p. 1752, 2025. [Online]. Available: https://www.mdpi.com/2227-9032/13/14/1752.

[65] L. Prashanth, C. Jie, M. Fu, S. Marcus, and C. Szepesvári, "Cumulative prospect theory meets reinforcement learning: Prediction and control," in International Conference on Machine Learning, 2016: PMLR, pp. 1406–1415.

[66] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," IEEE transactions on neural networks and learning systems, vol. 35, no. 11, pp. 15051–15071, 2023.

[67] N. Dalla Pozza, L. Buffoni, S. Martina, and F. Caruso, "Quantum reinforcement learning: the maze problem," Quantum Machine Intelligence, vol. 4, 06/01 2022, doi: 10.1007/s42484-022-00068-y.

[68] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 4, pp. 1544–1561, 2020.

[69] K. M. Ali Alheeti and M. Shaban, "A Predication Scheme for Parkinson's Dieses Based on Deep Learning Neural Networks," AUS, 09/24 2019, doi: 10.4206/aus.2019.n26.2.55.

[70] B. Dai, Y. Cao, Z. Wu, Z. Dai, R. Yao, and Y. Xu, "Routing optimization meets Machine Intelligence: A perspective for the future network," Neurocomputing, vol. 459, pp. 44–58, 2021.

[71] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," IEEE/ACM Transactions on Networking, vol. 24, no. 2, pp. 1038–1051, 2015.

[72] R. Rahman and S. Hasan, "Data-driven traffic assignment: A novel approach for learning traffic flow patterns using graph convolutional neural network," Data Science for Transportation, vol. 5, no. 2, p. 11, 2023.

[73] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, and J. M. Arco, "A survey on machine learning techniques for routing optimization in SDN," IEEE Access, vol. 9, pp. 104582–104611, 2021.

[74] Z. Gong, P. Zhong, and W. Hu, "Diversity in machine learning," Ieee Access, vol. 7, pp. 64323–64350, 2019.

[75] J. Reis, M. Rocha, T. K. Phan, D. Griffin, F. Le, and M. Rio, "Deep neural networks for network routing," in 2019 International Joint Conference on Neural Networks (IJCNN), 2019: IEEE, pp. 1–8.

[76] S. Munawar, Z. Ali, M. Waqas, S. Tu, S. A. Hassan, and G. Abbas, "Cooperative computational offloading in mobile edge computing for vehicles: A model-based DNN approach," IEEE Transactions on Vehicular Technology, vol. 72, no. 3, pp. 3376–3391, 2022.

[77] Z. Liu et al., "DNN partitioning, task offloading, and resource allocation in dynamic vehicular networks: A Lyapunov-guided diffusion-based reinforcement learning approach," IEEE Transactions on Mobile Computing, vol. 24, no. 3, pp. 1945–1962, 2024.

[78] W. Ou and S. Bi, "Sequential decision-making under uncertainty: a robust MDPs review," Annals of Operations Research, vol. 353, no. 3, pp. 1239–1285, 2025.

[79] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," IEEE Transactions on Cognitive and Developmental Systems, vol. 10, no. 3, pp. 514–529, 2018.

[80] L. Liu, A. Chattopadhyay, and U. Mitra, "On solving MDPs with large state space: Exploitation of policy structures and spectral properties," IEEE Transactions on Communications, vol. 67, no. 6, pp. 4151–4165, 2019.

[81] N. Peng, Y. Xi, J. Rao, X. Ma, and F. Ren, "Urban multiple route planning model using dynamic programming in reinforcement learning," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 8037–8047, 2021.

[82] P. Mahajan, B. Palanisamy, A. Kumar, G. S. S. Chalapathi, V. Chamola, and M. Khabbaz, "Multi-objective MDP-based routing in UAV networks for search-based operations," IEEE Transactions on Vehicular Technology, vol. 73, no. 9, pp. 13777–13789, 2024.

[83] A. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data link state routing protocol," in Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013, pp. 15–20.

[84] D. Calderone and S. S. Sastry, "Markov decision process routing games," in Proceedings of the 8th International Conference on Cyber-Physical Systems, 2017, pp. 273–279.

[85] W. K. Anuar, L. S. Lee, H.-V. Seow, and S. Pickl, "A multi-depot dynamic vehicle routing problem with stochastic road capacity: An MDP model and dynamic policy for post-decision state rollout algorithm in reinforcement learning," Mathematics, vol. 10, no. 15, p. 2699, 2022.

[86] X. Zhao, C. Wu, and F. Le, "Improving inter-domain routing through multi-agent reinforcement learning," in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020: IEEE, pp. 1129–1134.

[87] G. Bono, "Deep multi-agent reinforcement learning for dynamic and stochastic vehicle routing problems," Université de Lyon, 2020.

[88] C. Ayachi Amar, K. Bouanane, and O. Aiadi, "A Reinforcement Learning Framework for Scalable Partitioning and Optimization of Large-Scale Capacitated Vehicle Routing Problems," Electronics, vol. 14, no. 19, p. 3879, 2025.

[89] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in International conference on machine learning, 2017: PMLR, pp. 1352–1361.

[90] Y. Li, "Deep reinforcement learning: An overview," arXiv preprint arXiv:1701.07274, 2017.

[91] J. Eschmann, "Reward Function Design in Reinforcement Learning," in Reinforcement Learning Algorithms: Analysis and Applications, B. Belousov, H. Abdulsamad, P. Klink, S. Parisi, and J. Peters Eds. Cham: Springer International Publishing, 2021, pp. 25–33.

[92] K. O. McGraw, "The detrimental effects of reward on performance: A literature review and a prediction model," The hidden costs of reward, pp. 33–60, 2015.

[93] J. P. O'Doherty, J. Cockburn, and W. M. Pauli, "Learning, reward, and decision making," Annual review of psychology, vol. 68, pp. 73–100, 2017.

[94] K. Kalyanam, M. Pachter, and D. Casbeer, "Average reward dynamic programming applied to a persistent visitation and data delivery problem," in Dynamic Systems and Control

Conference, 2017, vol. 58295: American Society of Mechanical Engineers, p. V003T39A002.

[95] S. Ibrahim, M. Mostafa, A. Jnadi, H. Salloum, and P. Osinenko, "Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications," IEEE Access, vol. 12, pp. 175473–175500, 2024.

[96] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan, "Inverse reward design," Advances in neural information processing systems, vol. 30, 2017.

[97] C. Nakas, D. Kandris, and G. Visvardis, "Energy efficient routing in wireless sensor networks: A comprehensive survey," Algorithms, vol. 13, no. 3, p. 72, 2020.

[98] K. G. Somoye and Ş. Z. Eyupoglu, "The functionality of reward in influencing the reinforcement of performance evaluation criteria and organisational commitment among employees," South African Journal of Business Management, vol. 51, no. 1, pp. 1–11, 2020.

[99] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg, "Scalable agent alignment via reward modeling: a research direction," arXiv preprint arXiv:1811.07871, 2018.

[100] C. F. Hayes et al., "A practical guide to multi-objective reinforcement learning and planning: CF Hayes et al," Autonomous Agents and Multi-Agent Systems, vol. 36, no. 1, p. 26, 2022.