Towards autonomous collaboration in heterogeneous satellite systems using machine learning

PhD Research Plan - Aerospace Science and Technology PhD programme (DoCTA)

Estel Ferrer Torres
Distributed Artificial Intelligence
i2CAT Foundation
Barcelona, Spain
estel.ferrer@i2cat.net

Josep Escrig Escrig
Distributed Artificial Intelligence
i2CAT Foundation
Barcelona, Spain
josep.escrig@i2cat.net

Eduard Alarcón Cot
Department of Electronic Engineering
Universitat Politècnica de Catalunya
Barcelona, Spain
eduard.alarcon@upc.edu

Hyuk Park
Department of Physics
Universitat Politècnica de Catalunya
Barcelona, Spain
park.hyuk@upc.edu

***Abstract*—The increasing demands of modern space-based communication networks, particularly for 6G and beyond, require innovative solutions to improve the efficiency of Low Earth Orbit (LEO) satellite systems. Traditional and non-intelligent routing protocols face challenges in meeting the requirements for real-time data transmission, failing to adapt to dynamic and large-scale networks and changing traffic demands. This research aims to address these limitations by developing machine learning-based distributed routing protocols specifically designed for heterogeneous LEO satellite networks. The study will leverage reinforcement learning (RL) techniques to optimize next-hop routing decisions, accounting for the intermittent nature of inter-satellite links. Through a combination of supervised learning to predict satellite-to-satellite communication opportunities with RL-based routing, this research seeks to enable real-time, cost-efficient, and autonomous decision-making within Federated Satellite Systems.**

***Index Terms*—Low-Earth Orbits, Distributed Satellite Systems, Federated Satellite Systems, Neural Networks, Machine Learning, Supervised Learning, Reinforcement Learning**

I. Introduction

The launch of Sputnik 1 by the Soviet Union in 1957 marked the beginning of the space age, proving the feasibility of placing objects into Earth's orbit. This milestone paved the way for the growing number of satellites launched for scientific research, navigation, Earth Observation (EO), and communications. Figure 1a shows the evolution of Earth-orbiting satellites from 1957 to 2023, emphasizing the rapid expansion of the space sector, particularly in recent years. It highlights not only an exponential increase in satellite numbers but also a broader diversity of entities involved.

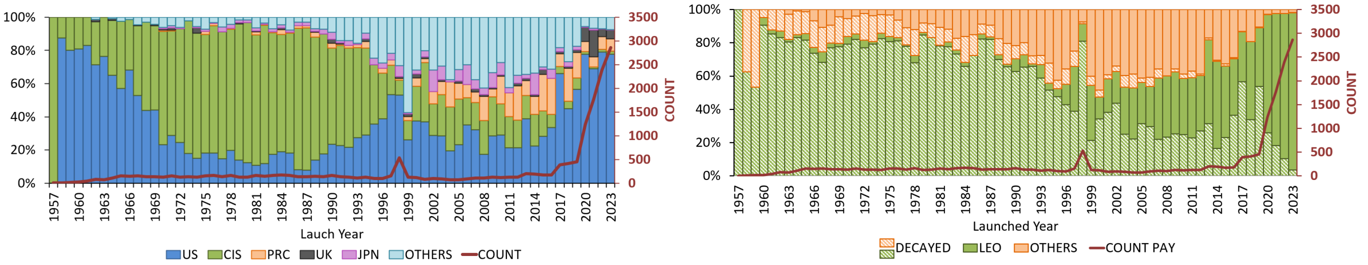
A. Monolithic satellite systems

During the 1960s and 1970s, satellite technology advanced significantly, driven by missions like Lunar 2 (the first to reach the Moon's surface), Vanguard 2 (the first photograph of Earth from space), Tiros-1 (the first weather satellite), and Intelsat I (the first commercial communication satellite). By the late 1970s, nearly 2,000 satellites had been launched, predominantly by the Soviet Union and the United States, with minor contributions from countries like the United Kingdom, France, China, and Japan. During this period, satellites were designed as monolithic systems, where all necessary functions and components were integrated into a single platform. These systems were inherently complex, requiring meticulous design and integration, often resulting in large platforms with significant resources, many of which remained underutilized. In such designs, the failure of a critical component could lead to the loss of the entire mission, as there was no redundancy beyond the single satellite. As a result, the development and launch costs were significantly high and these systems were inflexible to changing needs or mission requirements.

As shown in Figure 1b, most monolithic systems were launched into Low-Earth Orbits (LEOs), which range from 600 km to 2,000 km above the surface. LEOs are advantageous for high-resolution imaging and low-latency communications, making them particularly attractive for environmental monitoring and global internet coverage. However, LEOs suffer from short operational lifespans due to atmospheric drag, causing most satellites launched before the 2000s to be no longer operational, as shown by the hatched areas in Figure 1b. Despite these limitations, LEO remains the most densely populated orbital region today.

B. Distributed Satellite Systems

Advancements in space technology have significantly reduced development and launch costs, enabling the emergence of small, low-cost platforms with reduced mass and simplified designs, opening the door for a broader range of organizations



(a) Annual proportion of different satellite owners, including United States (US), Commonwealth of Independent States (CIS), People's Republic of China (PRC), United Kingdom (UK), and Japan (JPN).

(b) Annual proportion of LEO satellites, decayed satellites, and other non-decayed satellites

Fig. 1. Histogram illustrating the annual number of satellites launched from 1957 to 2023, detailing the distribution of satellite owners and orbital types over time. Numeric data sourced from Celestrak [1].

to participate in satellite missions. These changes have given rise to Distributed Satellite Systems (DSS) [2], consisting of multiple smaller satellites working collaboratively toward shared objectives, such as global communication, Earth observation (EO), or navigation.

The first satellite constellation, the U.S. Navstar Global Positioning System (GPS), launched in 1978 and fully operational by 1993 with 24 Medium Earth Orbit (MEO) satellites across six orbital planes. Due to its higher altitude, MEO enables global coverage with fewer satellites and longer lifespans than LEO, allowing GPS to continue providing global navigation and timing services. In the following years, Russia, China, and Europe launched their own Global Navigation Satellite Systems (GNSS), known as GLONASS, BeiDou, and Galileo, respectively.

Beyond GNSS, constellations have greatly improved communication services, especially in areas lacking traditional infrastructure. In 1998, the Iridium constellation became the first to offer global satellite phone and data services via 66 LEO satellites distributed across six polar orbital planes at 780 kilometers altitude. Equipped with Inter-Satellite Links (ISLs), these satellites enable direct communication routing without relying on ground stations.

Constellations have also become a key solution for EO applications such as weather forecasting, disaster management, agriculture, and climate research. As an example, the U.S. Planet Labs' Doves constellation, operational since 2015, comprises 200 small LEO CubeSats satellites capturing near-daily imagery for environmental monitoring. A key feature of most EO constellations is that they operate in Sun-synchronous orbits (SSOs), ensuring consistent lighting conditions for reliable data [3].

The advancements in space technology have enabled the development and launch of large numbers of satellites and constellations at reduced costs, resulting in a growing number of orbiting objects around the Earth and shaping the future of space missions. Additionally, with the definition of fifth-generation mobile communication (5G) in 2015 the satellite mission requirements evolved, especially with the definition of Non-Terrestrial Networks (NTNs) [4], to expand and complement terrestrial communication networks, enhancing connectivity for Internet of Things (IoT) devices, supporting ultra-reliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB).

These new requirements far surpass the capabilities of traditional satellite constellations, which were designed for specific missions with predefined and limited objectives. To meet this growing demand, recent DSS constellations are further increasing their capabilities, such as Iridium NEXT, Starlink [5], and OneWeb [6]. Moreover, these increasing demands are also impacting EO constellations, driving the need for real-time imagery with advanced sensing capabilities. To manage the rising data volume, ISLs are being explored as a solution, allowing direct satellite-to-satellite communication to optimize storage capacity and improve ground-link availability. This approach could greatly enhance data transfer efficiency and throughput, enabling real-time EO applications [7].

Although these DSS constellations with enhanced capabilities address most of the growing demands, they still encounter notable inefficiencies. Typically consisting of homogeneous satellites with predefined functions, these constellations struggle to adapt to real-time, dynamic, and evolving requirements. Additionally, traditional DSS often suffer from resource underutilization during duty cycle operations, leaving critical resources idle during inactive mission phases.

C. *Federated Satellilte Systems*

Federated Satellite Systems (FSS) represent a specific type of DSS where heterogeneous satellites with independent missions collaborate opportunistically to share unused resources such as power, storage, processing time, and downlink opportunities [8]. The concept was developed between 2013 and 2015 driven by the increasingly crowded near-Earth space environment and drawing inspiration from the principles of cloud computing, where nodes share resources on demand, optimizing their usage and avoiding maintaining independent and expensive infrastructure with underutilized resources. This collaborative approach not only enhances the performance of existing satellite missions but also enables the creation of new virtual missions. In that regard, FSS can better adapt to the growing demands, requirements, and emerging technologies, enhancing scalability without the need to deploy additional constellations.

The authors in [8] also present a comprehensive analysis of implementing FSS to enhance EO missions. The case study demonstrates how a customer radar altimetry satellite, along with 41 resource-supplier LEO satellites from the Iridium constellation, can significantly increase total access time, enabling near-real-time data and expanding EO mission capabilities. FSSs are also beneficial for satellite communication networks by replacing malfunctioning communication nodes, providing additional storage, creating alternative and less congested paths, and handling traffic spikes or long-term increases, resulting in a more robust and adaptable network.

The concept of FSS remains an active area of research due to the significant technical and operational challenges involved in establishing a peer-to-peer communication network between the customer and the supplier. First, the diversity of ISL technologies in this heterogeneous environment complicates the reliable establishment of the federations. Authors in [9] propose the use of Software Defined Radio (SDR) as a flexible communication system capable of dynamically adjusting the Radio Frequency (RF) characteristics. Second, the involvement of multiple stakeholders introduces legal complexities, particularly regarding the sharing of systems and sensitive data across different organizations. Security requirements and authentication protocols, as discussed in [10], are crucial to addressing these concerns. Third, the varying altitudes and inclinations of heterogeneous networks create dynamic typologies characterized by sporadic and intermittent communication links. This variability makes it challenging to establish optimal routing paths for efficient resource sharing.

While the first and second challenges focus on device standardization and protocol development, the third presents greater opportunities for research into optimizing end-to-end communication. Moreover, the emergence of sixth-generation mobile communications (6G) is placing unprecedented demands on satellite networks, requiring real-time applications with ultra-low latency, high data rates, and ultra-high frequency bandwidth for ultra-wide-area broadband access [11], [12]. In such a complex 6G environment with increasing density, heterogeneity, and traffic demands, Artificial Intelligence (AI) is seen as a crucial enable for its predictive capabilities, such as forecasting traffic patterns, user behaviors, and potential network failures. Additionally, AI-driven satellite operations allow intelligent and real-time decision-making processes with continuous adaptation to real-time conditions, ensuring optimal performance [13], [14].

Given the complexity of establishing FSS under the stringent demands of 6G, this thesis specifically focuses on addressing the challenge of developing optimal and efficient routing protocols for heterogeneous networks with intermittent ISLs. We leverage AI's predictive capabilities and real-time decision-making to ensure optimal end-to-end routing decisions.

II. Related Work

A key challenge in FSS is managing the significant relative motion between heterogeneous satellites with different orbital parameters. This dynamic environment is characterized by sporadic and intermittent ISLs, defined based on satellite proximity and antenna pointing direction. Consequently, resource sharing between neighboring satellites becomes difficult, as links may break before the federation is complete. The complexity increases when federations need to be established between distant satellites, as disruptions in any intermediate link can break the connection between the supplier and the customer.

In homogeneous systems like the Iridium constellation, satellites are positioned at uniform altitudes and inclinations. In this mesh-like configuration, intra-plane satellites maintain constant distances, while relative motion occurs only between inter-plane satellites, especially near the poles and across orbital seams. Consequently, Iridium satellites are equipped with two consistently available intra-plane ISLs and two temporal inter-plane ISLs. This homogeneous configuration significantly simplifies routing, as satellites not only share communication capabilities but also move in coordinated and predictable patterns. From the satellite's perspective, the network appears static, with only periodic disruptions of certain inter-plane links. In an ideal scenario, where node states remain unchanged, optimal end-to-end routes between satellites can be precomputed using algorithms such as Dijkstra [15]. Additionally, although the space-ground network topology is not fixed, it follows a predictable and periodic pattern of several days, allowing the precomputation of optimal paths between satellites and ground stations. However, static routing protocols are insufficient in real-world network conditions, where node states change dynamically due to factors like congestion, link failures, limited battery, or restricted bandwidth. In that sense, adaptative or dynamic routing protocols like the Extended Bellman-Ford (EXBF) [16], while more complex, must be implemented to ensure consistent network efficiency and reliability [17].

In contrast to mesh-like topologies, heterogeneous networks face unique challenges due to the diversity of orbital periods and inclinations. Although orbits in heterogeneous satellite systems are also predictable and periodic, the overall periodicity can span several days or even months. In such conditions, even assuming stable node states, static protocols are impractical because end-to-end routes with intermittent ISLs are only valid for short periods, and factors such as atmospheric drag and the Earth's non-sphericity can render these routes unusable after the system's long periodicity cycle. Moreover, as the number of satellites increases, the system periodicity extends, leading to scalability issues.

In such highly dynamic environments, traditional protocols like EXBF can struggle due to their heavy ground link dependencies, Additionally, the convergence time required by these protocols may be insufficient to handle rapid topology changes, requiring extensive end-to-end path re-computations.

Autonomous FSS offers a promising solution in this context by reducing the important ground-link dependencies that would render real-time applications unfeasible, especially in overpopulated space environments where optimal routes change rapidly. A major challenge for implementing autonomous satellite systems is the limited onboard resources, making onboard centralized computations impractical and non-scalable as the computational demands increase with the number of satellites. Moreover, traditional autonomous flooding-based routing methods for neighbor state discovery, such as Darting [18], can generate excessive overhead, consuming critical resources and reducing system efficiency.

Distributed FSS are gaining prominence as a lightweight, scalable solution for decentralized network management in resource-constrained environments. By empowering satellites to autonomously make routing decisions based solely on local information, these systems enable real-time, cost-effective operations. Although local data may not always guarantee globally optimal solutions, it enables faster and more efficient decision-making while ensuring scalability by only using local information.

A. Autonomous satellite encounter prediction

To address the challenges of autonomous and decentralized routing in FSS networks with sporadic ISLs, it is essential for satellites to predict their communication opportunities, or encounters, with their neighbors. These encounters represent the time windows during which two satellites are within range to establish a communication link. Accurately forecasting these opportunities not only reduces energy consumption by avoiding failed connection attempts but also optimizes the use of available communication resources.

Traditional methods for anticipating satellite encounters typically rely on deterministic, centralized, ground-based orbit propagation techniques [19], [20]. While on-ground approaches deliver highly accurate contact windows, their centralized nature limits scalability, rendering them impractical for large-scale satellite networks. Recent research has explored decentralized solutions to grant satellites greater autonomy [21]. However, these approaches still rely on orbital propagation, which imposes considerable processing demands on resource-constrained satellites. The authors in [22] propose an alternative cost-efficient method to predict satellite contacts without relying on extensive orbit propagation solutions. Nevertheless, this approach is constrained by some mathematical linearization, limiting its applicability to simplified environments where satellites are assumed to follow circular Keplerian orbits. As a result, it becomes unsuitable for realistic LEO networks, where near-circular orbits are affected by significant perturbations such as atmospheric drag and Earth's oblateness. Given the promising capabilities and wide-range applications of Machine Learning (ML) techniques, the authors in [23] explore the use of Graph Neural Networks (GNNs) together with Recurrent Neural Networks (RNNs) as a scalable solution for autonomously and cost-effectively modeling the temporal evolution of satellite encounters. However, these algorithms require comprehensive network information, demanding more extensive data acquisition and processing capabilities compared to methods that operate solely on local information.

Building on this, we envision the need to anticipate the contact opportunities between satellites in a distributed, cost-efficient, and autonomous manner. By leveraging ML, we can effectively capture the complex dynamics of satellite motion, including atmospheric drag and Earth's non-sphericity, while benefiting from fast and cost-effective model inferences.

B. Traditional routing protocols

Authors in [24] investigate current technologies to assess the feasibility of adapting existing routing protocols to meet the requirements of FSS. They introduce the concept of Inter-Satellite Networks (ISN), a heterogeneous network that enables sporadic end-to-end connections between distant satellites through multiple intermediate nodes. They classify existing routing protocols into four categories: LEO Satellite Network (LSN) protocols, Multi-Layered Satellite Network (MLSN) protocols, Delay-Tolerant Network (DTN) protocols, and Mobile Ad-hoc NETwork (MANET) protocols.

Protocols developed for LSNs are specifically designed for mesh-like topologies in homogeneous satellite networks, where multiple minimum-hop end-to-end paths can be easily derived by comparing the vertical and horizontal coordinates of the source and destination nodes. Among the most well-known protocols are the Explicit Load Balancing (ELB) protocol [25] and the Traffic-Light-based Routing (TLR) protocol [26]. Both are decentralized, proactive routing protocols that provide suboptimal solutions by considering only the queue length of neighboring nodes and rerouting traffic through alternative minimum-hop paths to prevent congestion. Nevertheless, these solutions are strongly bound by the mesh-like topology and are difficult to apply to the more complex and heterogeneous structure of ISNs.

In contrast, MLSN protocols integrate constellations with varying capabilities across LEO, MEO, and GEO layers, maintaining a mesh-like topology within each layer. Upper-layer satellites act as relays for lower-layer groups, managing congestion through Inter-Layer Links (ILLs). While these lower-layer groups are fixed, they are periodically managed by different upper-layer satellites due to their relative motion. Among these protocols, the Multi-Layered Satellite Routing (MLSR) algorithm [27] stands out for computing the shortest delay paths between satellites and ground gateways in a three-layer network, enhancing network capacity, reliability, and global coverage compared to single- or dual-layer protocols. However, MLSR's reliance on centralized path calculations by GEO satellites using link-state control packets exchanged across all the layers introduces scalability challenges, limiting its effectiveness in large, dynamic networks or latency-sensitive real-time applications requiring low latency and fast adaptability.

Alternatively, MANET is a network of mobile devices characterized by its dynamic and unpredictable topology, where nodes can move freely and independently, causing frequent changes in the network structure. MANET routing protocols manage this unpredictability through a discovery phase, where nodes flood the network with ”hello” packets to identify neighbors and establish routes reactively or proactively. Reactive protocols, like the Ad Hoc on-Demand Distance Vector (AODV) protocol [28], reduce overhead by discovering routes only when a packet needs to be forwarded but can introduce delays in adapting to topology changes, making them unsuitable for highly dynamic networks demanding ultra-low end-to-end latency. In contrast, proactive protocols, like the Optimized Link State Routing Protocol (OLSR) [29], ensure faster adaptation by continuously updating routes but consume significant bandwidth and battery life due to constant control packet exchanges, even when data transmission is unnecessary. For this reason, proactive protocols are also not ideal for highly dynamic and resource-limited networks, as the associated overhead can quickly deplete the nodes' battery life. More sophisticated and efficient strategies than flooding mechanisms can be employed to optimize both reaction time and bandwidth in networks with deterministic node mobility, such as ISNs.

Finally, DTNs are designed to handle long delays and frequent disconnections, using a store-and-forward approach to temporarily store data at intermediate nodes until a forwarding opportunity arises, enabling transmission even without direct paths to the destination. While unsuitable for real-time applications, DTNs excel in scenarios like deep space communications with low node density and intermittent connectivity. In large heterogeneous satellite networks, where paths between nodes usually exist, DTNs can still be valuable, especially in FSS, by prioritizing critical data while storing less urgent requests for optimal transmission opportunities. This flexibility makes DTNs a useful tool for optimizing data flow in resource-constrained environments. Contact Graph Routing (CGR) [30] is a dynamic and centralized DTN algorithm, that calculates optimal paths for each data transmission based on the time-varying network conditions modeled as contact-graph, where nodes represent communication links. While effective in space communication, CGR's scalability to larger networks is limited. The Shortest-Path Tree Approach for Routing in Space Networks (SPSN) [31] addresses this by replacing the contact-graph model with a traditional node-vertex graph, where edges represent contact opportunities. SPSN allows computing optimal routes for all destinations in a single execution, significantly reducing computational time and improving efficiency in large-scale space networks with thousands of contacts. However, SPSN is a centralized approach that still depends on complete network state information, which is often not available in real-time, limiting its applicability in dynamic, real-time environments.

The authors in [24] conclude their analysis by noting that none of the existing routing protocols fully meet the unique requirements of ISNs, including distributed, adaptive, and resource-constrained solutions. As for future research, they recommend the development of a specific routing protocol tailored for ISNs.

C. Machine Learning-based routing protocols

The authors in [24] do not provide an in-depth exploration of distributed hop-by-hop protocols or more modern approaches involving ML. On the one hand, although hop-by-hop protocols may result in sub-optimal solutions, they allow each node to make simple and cost-efficient forwarding decisions based solely on local information. These protocols are attractive candidates for large-scale, resource-constrained networks due to their fast response times and enhanced autonomy, eliminating the dependency on centralized, multi-hop ground-based computing. Furthermore, centralized routing protocols not only face challenges related to complexity and scalability but are also vulnerable to single-node link failure compromising the entire routing path. On the other hand, ML approaches offer a promising alternative for resource-constrained satellite networks by avoiding the slow convergence of metaheuristic methods like Genetic Algorithms (GA) [32] and the high computational demands of exact solutions like Dijkstra's algorithm. By leveraging offline training and onboard execution, ML approaches can adapt more quickly to network changes, making them especially valuable for highly dynamic satellite networks.

The authors in [33] provide a comprehensive survey of ML-based solutions for intelligent routing, with a focus on both multi-hop and hop-by-hop strategies. They explore how ML can be applied to optimize various aspects of end-to-end communication. The study highlights ML as a promising alternative to traditional network optimization methods, which often struggle with the high complexity and dynamic nature of 6G networks due to their limited adaptability, lack of intelligent decision-making, and slow convergence rates. The survey mainly categorizes ML-based routing approaches into two groups: Supervised Learning (SL) and Reinforcement Learning (RL).

SL-based routing approaches predict optimal paths using historical traffic patterns incorporating both past traffic flows and link-state information. SL-based methods have been shown to outperform traditional techniques by reducing computational complexity and signaling overhead [34]. However, these approaches struggle to adapt to dynamic environments, especially in heterogeneous networks characterized by high variability and complexity. Additionally, the accuracy of SL-based routing decisions is often constrained by the quality and relevance of the benchmark training data, which can degrade performance in real-time network scenarios.

RL has recently gained significant attention in the field of routing due to its ability to learn routing policies or strategies without the need for labeled training data, making it particularly well-suited for complex systems where acquiring such data is difficult or impractical. RL agents learn optimal policies by interacting with an environment, taking different actions based on the current state, and receiving reward feedback. Initially, agents take random actions to explore the environment, but as training progresses, they increasingly exploit the best actions, gradually reducing the exploration rate to learn the policy that maximizes cumulative rewards over time. Routing policies are trained on-ground with abundant resources and uploaded to resource-constrained agents. These policies can then undergo further fine-tuning with minimal exploration, enabling the agents to make real-time decisions based on the current state of the environment while continuously adapting their strategies according to the received rewards, offering significant advantages against dynamic scenarios and unexpected conditions.

In the LSN context, the environment is the LEO satellite network, characterized by a given topology and dynamism. This environment can accommodate either a single learning agent or multiple agents learning collaboratively or competitively in a decentralized manner. Additionally, the RL agents can select either the next-hop node to forward the packet (hop-by-hop protocols) or the entire routing path (multi-hop protocols). Hop-by-hop protocols rely on local information to determine the next node for packet forwarding, offering faster adaptation with lower computational demands. In contrast, multi-hop protocols calculate entire routing paths based on global network states. While multi-hop protocols ensure global optimality, they require extensive resources and complete network information, and are more sensitive to single-node failures or congestion, rendering the entire path unusable. Instead, hop-by-hop protocols are better suited for resource-constrained environments, providing quicker responses to network dynamics. While decentralized approaches may not guarantee global optimality, they are more adaptable and scalable in dynamic satellite networks.

In this context, we reviewed a range of recent studies on decentralized RL-based routing protocols with hop-by-hop decision-making strategies. We categorized these protocols according to the network topology, distinguishing between static and dynamic.

1) Static topology: In a static topology, nodes remain fixed, either because they are stationary or because they all move with the same velocity and pattern. For instance, in a satellite constellation network like Iridium, topology can be considered static, even though there is some relative motion between inter-plan satellites in both cross-seams and polar regions.

In [35], the authors propose a decentralized routing protocol for packet transmission between ground stations through a polar satellite network comprising 150 satellites in a static topology. The relative motion between satellites and the ground stations can be assumed null during the transmission time. Each satellite makes distributed next-hop forwarding decisions based on local information from its four consistent neighboring nodes. The state space is modeled with discrete values representing the congestion level of each node, and a tabular RL algorithm, Q-learning, is applied. The reward for each action is based on the queuing delay of the next-hop node and the distance to the destination. When the next-hop node has already been visited, the reward for that action is heavily penalized. Successfully forwarding the packet to the destination yields a large positive reward.

Similarly, [36] applies Q-learning for a decentralized routing between satellites in a Walker Delta constellation with thousands of nodes. Apart from evaluating the average end-to-end delay, they measure the packet drop rate, defined as the ratio of packets not received before the Time-to-Live (TTL).

Both [35] and [36] demonstrate that decentralized, cost-efficient Q-learning-based routing (Q-routing), despite utilizing simplified state representations, can dynamically adapt to local state information, achieving performance comparable to traditional centralized Shortest-Path (SP) algorithms. Under high traffic demands, these approaches have been shown to outperform centralized algorithms by minimizing end-to-end delays and effectively preventing congestion and packet loss.

Other research addresses distributed routing in satellite networks using more advanced RL algorithms, known as Deep Reinforcement Learning (DRL) capable of handling continuous state spaces through Neural Networks (NNs).

Authors in [37] apply a DRL approach for decentralized routing in a polar satellite network with 48 LEO satellites evenly distributed across eight orbital planes. Leveraging NNs, each satellite can take next-hop forwarding decisions based on a continuous state. While this approach is more resource-intensive during both training and inference, it can improve performance by incorporating more realistic and detailed information about neighboring nodes. The state space includes metrics such as Signal-to-Noise Ratio (SNR), queuing delay, bandwidth, and distance to the destination. The results demonstrate improved average end-to-end delays compared to two benchmark decentralized protocols, which also rely on local information but select the next hop solely based on the highest node weight, without incorporating feedback from current or past actions.

Similarly, the authors in [38] train a decentralized DRL-based routing algorithm for a Starlink network consisting of 140 satellites. The local continuous state space considers only the queue length of neighboring satellites and their distance to the destination. However, the reward is dynamically adjusted based on the congestion level of the next-hop node, considering the propagation delay when the queue is relatively free, or the transmission time when the queue is near congestion. The results demonstrate that the DRL-based routing outperforms decentralized protocols, with lower delivery times, reduced packet loss rates, and higher throughput, particularly under heavy traffic loads. Furthermore, the DRL-based routing demonstrates superior performance compared to Q-routing with discrete state space, emphasizing the importance of utilizing realistic information.

The authors in [39] apply a decentralized DRL-based routing strategy to the Iridium constellation. This appraoch incorporates the Depth of Discharge (DoD) of neighboring satellites into the state space, allowing for a traffic balance by forwarding packets to satellites with higher battery capacity, extending the operational lifespan of the satellite network. Additionally, the strategy ensures timely packet delivery by incorporating the remaining propagation time of the packet into the state space. The results demonstrate more balanced power consumption compared to SP-based routing, while still meeting end-to-end delay constraints.

Alternatively, the DRL-based decentralized routing in [40] considers data from two-hop neighbors to anticipate and avoid congestion before it occurs. Particularly, the state only considers the queue length and the congestion level, classified as free, busy, or congested. During free or busy queues, the reward is proportional to the distance between the next hop and the destination. When the next hop is congested, the reward is penalized with a negative value. This approach outperforms ELB, achieving a significantly lower drop rate, reduced end-to-end delay, and increased throughput.

These studies highlight the effectiveness of DRL-based decentralized routing in static satellite networks. The diversity and complexity of the state space representations showcase its flexibility in capturing various link state attributes, such as queue length, capacity, and battery levels, while enabling the reward function to be tailored to specific optimization goals. Compared to traditional SP routing methods, DQN-based approaches demonstrate significant reductions in end-to-end delays, packet loss rates, and traffic imbalances, particularly under heavy traffic loads. This illustrates the effectiveness of RL-based decision-making techniques in addressing the growing requirements in 6G networks in terms of ultra-low latency and increased traffic demands. Nevertheless, their applicability has been shown solely for mesh-like architectures, and their performance in more complex environments with dynamic topologies remains unproven.

2) Dynamic topology: Some studies have focused on addressing the routing challenges in dynamic topologies using RL approaches. Dynamic topologies are characterized by relative movement between nodes, which can be either predictable or random. In the context of FSS, the ISN discussed in [24] features a highly dynamic yet predictable topology network, where heterogeneous satellites move at different velocities, creating sporadic links and a constantly changing number of neighboring nodes.

Authors in [41] applied a DRL approach to manage traffic routing in a Space-Air-Ground Integrated Network (SAGI-Net), where data packets are transmitted from ground users (GUs) and unmanned aerial vehicles (UAVs) to either base stations (BS) or satellites. UAVs also act as intermediaries, assisting the network when GU-to-BS or GU-to-satellite links become congested or suffer from reduced speeds. The DRL model is structured as a cooperative multi-agent system, where GUs and UAVs coordinate their actions to forward packets while optimizing communication energy efficiency. However, the effectiveness of the algorithm is demonstrated on a small-scale network, comprising only one satellite, three UAVs, one base station, and five ground users.

In [42], the authors apply decentralized Q-learning for routing in a satellite network composed of two mesh-like layers: 126 satellites in Very LEO (VLEO) and 50 in LEO. VLEO satellites are organized into static groups, each group communicating with the nearest LEO satellite acting as the group manager. Due to the relative motion between the layers, the LEO manager of each VLEO group changes dynamically. The Q-learning-based routing algorithm directs traffic through less congested routes using inter-orbital links. Results show improved traffic distribution, significantly lower drop rates, and enhanced throughput compared to SP, particularly in the event of satellite failure. However, the environment is relatively simple, consisting of just two homogeneous layers characterized by a small topology periodicity of only one day. In this setup, each LEO satellite can pre-compute the joining and departure times for each VLEO group. Therefore, this approach would not scale well in larger and more dynamic heterogeneous networks.

The authors in [43] utilize different link connections within the NSFNet dataset, featuring 13 nodes and arbitrary links, to model a snapshot of a heterogeneous satellite network with varying numbers of neighbors. To handle the fluctuating number of neighboring satellites, they employ a GNN to aggregate local information of each satellite, such as link capacity and queue delay of its neighbors. This aggregated information remains consistent in size, regardless of the number of neighbors, and serves as the state representation for the DRL algorithm. Based on this information, the RL agent makes next-hop routing decisions aimed at minimizing end-to-end delay and maximizing throughput. Results demonstrate that RL-based routing outperforms the SP algorithm in both throughput and end-to-end delays. However, the action space is fixed and may not fully capture all possible next-hop link connections in large-scale network topologies. Additionally, scalability issues may arise in the GNN state representation phase when applied to larger networks.

The authors in [44] propose a distributed multi-agent deep reinforcement learning (MADRL) approach to enhance routing performance in MANET consisting of 12 nodes. These networks are highly dynamic, with unpredictable node mobility, requiring flooding mechanisms for neighbor discovery. In this approach, each node functions as an autonomous agent, optimizing next-hop forwarding decisions based on local observations of the network state. During training, all nodes share

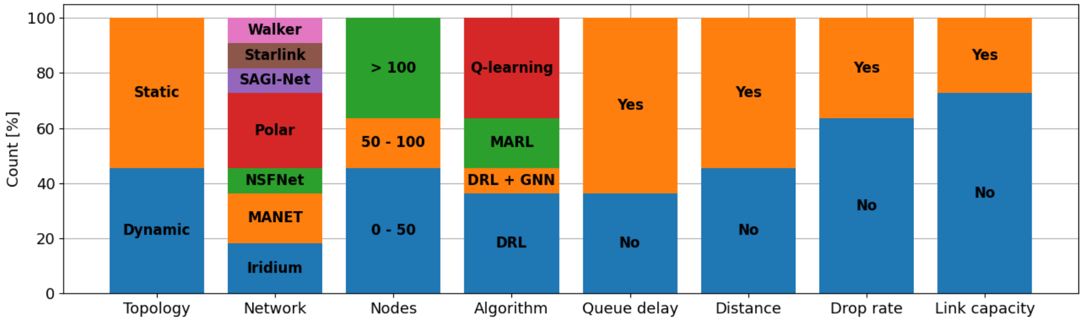


Fig. 2. RL-based routing in [35]–[45] classified based on different categories.

and update the same routing policy and receive a common reward based on the global state space and joint actions of all agents. Each node maintains a routing table for each neighbor-destination pair. Depending on the confidence levels in the table entries, the agent decides whether to forward packets via broadcast or unicast. The method demonstrates robustness and scalability, maintaining strong performance even under conditions beyond the original training scenarios, such as larger networks of up to 30 nodes and more dynamic mobility patterns.

Similarly, the authors in [45] present a Q-learning-based routing algorithm designed for MANETs with 20 to 40 nodes. This approach leverages Q-learning to indirectly estimate node behavior, selecting the optimal next-hop to minimize transmission delays while optimizing route length and stability. The algorithm dynamically adjusts to network changes based on local node information. Simulations demonstrate its superior performance compared to traditional MANET routing protocols, particularly in terms of end-to-end delay and packet delivery ratio.

Nevertheless, both [44] and [45] are specifically tailored to MANETs, which limits their applicability in satellite systems. Despite the dynamic nature of heterogeneous satellite systems, their node mobility is predictable, and neighbor discovery based on flooding mechanisms must be avoided to optimize the onboard resources and increase the overall routing performance.

Previous works demonstrate the applicability of RL-based routing protocols in dynamic topology networks. While static topology methods are unsuitable for heterogeneous networks, dynamic topology methods are well-suited for networks with relative node mobility. However, the analyzed studies have primarily focused on specific configurations, either involving unpredictable node motion, such as MANETs, or predictable motion with limited dynamism, such as small satellite networks structured in mesh-like layers. Figure 2 provides a summary of the related works on RL-based routing, classifying them into different key categories: topology (static or dynamic), network type, number of nodes, the specific RL algorithm applied, and the four most commonly used state representations (queueing delay, distance to destination, drop rate, and link capacity).

Nevertheless, none of the aforementioned studies directly address the unique challenges presented in large-scale, heterogeneous satellite systems, particularly within the context of FSS. These systems are characterized by varying satellite capabilities and frequent link disruptions, creating a far more complex environment that remains largely unexplored. Although existing RL-based methods for dynamic topologies could potentially be applied, their performance is likely to be suboptimal in such scenarios. This underscores the necessity for a distributed routing protocol specifically tailored to heterogeneous satellite networks. Such a protocol must effectively manage intermittent ISLs, accommodate the dynamic nature of neighboring satellites, and leverage the predictable orbital motion inherent to these systems. To the best of our knowledge, no ML-based approaches have yet been applied to address these challenging scenarios.

III. Thesis Objectives

I aim to contribute to the definition of a routing protocol to be applied in the context of FSS, characterized by a highly dynamic and heterogeneous environment, with resource-constrained nodes. To address the limitations of centralized solutions, this protocol will rely on distributed, autonomous decision-making, enabling each satellite to select the optimal next hop based on its local network state.

To achieve this goal, the thesis is structured around the following objectives:

- **Objective 1 (O1). Predict satellite-to-satellite encounters using Supervised Learning.**
  This objective involves developing an algorithm capable of forecasting future contact opportunities between any two LEO satellites from any constellation. The algorithm must account for the key perturbations present in LEO space and provide a cost-effective solution in terms of time and resources, suitable for deployment on resource-constrained satellites. Supervised Learning is proposed due to its efficiency, rapid inference capabilities, and ability to learn complex orbital dynamics from large datasets.

- **Objective 2 (O2). Develop a Single-Agent RL-Based Routing Protocol for Heterogeneous LEO Satellite Networks.**
  This objective focuses on learning an optimal routing policy for next-hop decisions in a dynamic, heterogeneous satellite network. The policy will rely on local information from neighboring satellites, including queue length, position, and link characteristics. It must effectively balance network traffic to prevent congestion while minimizing end-to-end delay and ensuring each packet meets its delivery time constraints. Additionally, the policy will integrate the satellite encounter prediction framework from Objective 1 to provide real-time information about neighboring satellites at each time step.
- **Objective 3 (O3). Develop a Multi-Agent RL-Based Routing Protocol for Heterogeneous LEO Satellite Networks.**
  Building on Objective 2, this objective addresses the limitations of the single-agent approach by enabling multiple agents to collaborate toward a shared goal. The protocol will utilize centralized training, leveraging complete information about the network state and the actions taken by each agent. This collaborative approach aims to enhance overall network performance and coordination among the satellites.
- **Objective 4 (O4). Disseminate research findings through publications and conferences.**
  This objective focuses on sharing the outcomes of the research with the scientific community through high-impact publications in peer-reviewed journals and presentations at relevant conferences. It involves documenting key advancements in the development of the routing protocol, including the SL-based satellite encounter predictions and both single-agent and multi-agent RL approaches for distributed routing. This dissemination will contribute to advancing knowledge in the field of FSS and networking in resource-constrained environments.

IV. Methodology and Resources

A. Methodology

- **Literature Review.** The first phase of the research consists of an extensive review of LEO satellite networks, routing protocols, and RL techniques. This phase aims to identify the state of the art, major challenges, and research gaps in these areas.
- **Problem Definition.** Based on the findings from the literature review, a formal problem statement is developed to address key challenges in LEO satellite routing, with a focus on scalability, adapting to dynamic network topologies, and achieving efficient routing under various network constraints and requirements. This phase also involves defining performance metrics and benchmarks for evaluating the proposed solutions.
- **Algorithm development.** The core research effort involves creating novel routing algorithms using reinforcement learning to address the identified challenges. Python will be the primary programming language due to its rich ecosystem of libraries for machine learning and deep learning (e.g., Keras, TensorFlow, and PyTorch).
- **Validation and testing.** After training, the algorithms will be validated and tested through different realistic LEO network conditions. Key performance metrics will be used to compare our solution against existing routing protocols. Performance will also be evaluated across different network topologies and traffic patterns.
- **Periodic meetings.** Regular meetings will be held to ensure steady progress and collaborative discussion. Weekly meetings with the supervisor will provide feedback on recent developments, while biweekly meetings with the co-supervisor and tutor will refine the problem formulation, review intermediate results, and discuss state-of-the-art research.
- **Publications.** Research findings will be documented and disseminated through publications in high-impact journals and conferences. This will not only contribute to the academic community but also allow for critical peer feedback and validation of the work.

B. Resources

- **i2CAT Foundation facilities.** Access to office space and essential equipment, including a personal laptop and peripherals. Additionally, it has a dedicated server for running computationally intensive simulations and experiments.
- **GitLab.** Local GitLab server for version control, enabling efficient code management, collaboration, and continuous integration throughout the project.
- **Jira.** A tool used to manage and structure projects. Moreover, Jira can be used to implement agile methodologies, with task tracking, sprint planning, and progress monitoring to ensure efficient time management and milestone achievement throughout the research.
- **Overleaf.** A collaborative LaTeX editor for writing and organizing academic documents, such as the thesis, research papers, and reports. Overleaf will also allow for seamless collaboration with supervisors.
- **Pycharm.** The integrated development environment for Python, essential for developing and debugging the Python-based algorithms and simulations used in the research. PyCharm's extensive support for machine learning libraries like TensorFlow and PyTorch will streamline the development process.
- **UPC Campus Nord Facilities and Labs**. Access to the laboratories and other research facilities at UPC's Campus Nord, providing a conducive environment for research, experimentation, and testing.
- **i2CAT Foundation Projects**. Participation in ongoing i2CAT projects, which provide financial support for publications, conference presentations, and travel. These projects will also offer opportunities for collaboration with external parties or internal departments such as Space Communication.

- **UPC Projects**. The ”Design of Low Earth Orbit Satellite Communication System” project is a collaborative project between Ajou University and UPC NanoSatLab aims to develop core technologies for an optimized LEO satellite-air-ground network, addressing the specific characteristics of LEO satellite networks and payload variations. The ultimate goal is to design a CubeSat that integrates these advanced technologies.
- **DSS-Simulator**. This simulator is a framework that enables to assess novel communications protocols in satellite networks. It is based on an event engine that is representative for networking events, and it has been extended to support satellite dynamics representation [46]–[48].

V. Tasks definition and work plan

A work package (WP) is defined for each of the objectives O1, O2, and O3. An additional WP is allocated for overall thesis-related tasks. To achieve objective O4, at least one milestone (M) is included in each WP.

WP1. SL-based Satellite Encounter Prediction

- **T1.1. Related works.**
  Conduct a comprehensive analysis of the SoA in encounter anticipation methods for heterogeneous networks.
- **T1.2. Data set creation.**
  Develop three distinct datasets for training, validation, and testing. To mitigate issues related to insufficient data, both the training and validation datasets will consist of synthetic data. The testing dataset, however, will be based on real-world data sourced from Celestrak. Apply normalization preprocessing to the datasets to facilitate the training process.
- **T1.3. SL Architecture Design.**
  Create a configurable framework for tuning hyperparameters of a fully connected neural network, including the number of layers and neurons per layer, to allow for easy adjustments during experimentation.
- **T1.4. Training.**
  Select an appropriate loss function to minimize during the training phase. Additionally, consider other evaluation metrics to assess final performance. Use a trial-and-error approach to test various configurations of the SL architecture.
- **T1.5. Validation and testing.**
  Evaluate the model’s performance using the validation and testing datasets, ensuring the final solution meets the required standards.
- **M1. Publications.**
  Produce at least one presentation in a relevant conference and one publication in a relevant journal, detailing the problem, methodology, and results obtained using the SL-based encounter anticipation model. Publications must demonstrate an accuracy higher than 90% and show faster inference times compared to SoA methods based on orbital propagation.

WP2. Single-Agent RL-based Distributed Routing Protocol.

- **T2.1. Related works**.
  Conduct a comprehensive analysis of the SoA in distributed RL-based routing within satellite networks, as well as other relevant domains.
- **T2.2. Static topology environment.**
  Simulate a simple satellite network with a static topology, where the number of orbits and satellites per orbit can be configured. Define the state space, action space, and reward functions that the agent needs to explore the environment and learn optimal routing policy.
- **T2.3. RL algorithm.**
  Implement a DQN with configurable parameters to ensure robustness and adaptability for use in various environments.
- **T2.4. Training and testing the DQN within the static topology environment**
  In addition to the total reward per episode, consider additional evaluation metrics such as average end-to-end delay and packet drop rate. Utilize a trial-and-error process to fine-tune the hyperparameters for the specific scenario. Train the agent by forwarding one packet at a time. Then, stress-test the policy by forwarding multiple packets simultaneously. Assess the model’s robustness by testing it in a larger satellite network than the one used during training. Compere the testing results with a benchmark method.
- **T2.5. Dynamic topology environment.**
  Building upon the environment from T2.2, extend the simulation by adding more satellites in various orbits to create a heterogeneous network. Redefine the state space to include communication encounter windows, leveraging the SL model developed in WP1. Modify the action space to accommodate varying numbers of neighbors dynamically.
- **T2.6. Training and testing the DQN within the dynamic topology environment**
  Similar to T2.4, perform a trial-and-error process to determine the optimal hyperparameters for the dynamic environment. Train the agent by forwarding single packets initially, and then stress-test by forwarding multiple packets simultaneously. Evaluate the model’s robustness by testing it in a larger, more dynamic, and heterogeneous satellite network. Compare the testing results with a benchmark method.
- **M2. Publications.** Deliver at least one presentation at a relevant conference and publish at least one article in a reputable journal, detailing the problem, methodology, and results of the RL-based decentralized routing in dynamic topology. The publications should demonstrate that the proposed method outperforms traditional routing algorithms in terms of end-to-end delay and packet drop rate, while also showcasing high robustness, scalability, and cost-efficiency suitable for resource-limited satellites.

WP3. Multi-Agent RL-based Distributed Routing Protocol.

- **T3.1. Multi-agent static topology environment.**
  Extend the environment described in T2.2 to accommodate multiple agents. This will primarily involve redefining the state space and reward function to reflect the multi-agent setting.
- **T3.2. Mutli-agent RL algorithm.**
  Enhance the DQN implementation from T2.3 to support centralized learning with decentralized execution. In this configuration, all satellites modify a shared policy, but the policy is learned based on the entire network's state and actions.
- **T3.3. Training and testing the DQN within the multi-agent static topology environment.**
  Fine-tune the hyperparameters using a trial-and-error approach tailored to the specific scenario. Train the agents in an environment where multiple forwarding decisions are made simultaneously by different agents. Test the model's robustness by evaluating it on a larger satellite network than the one used during training. Evaluate multiple metrics such as the average end-to-end delay and the packet drop rate. Compare the testing results with a benchmark method.
- **T3.4. Multi-agent dynamic topology environment.**
  Expand the environment described in T2.5 to support multiple agents. The same state space and reward function defined in T3.3 should be used in this scenario, with the assumption that each agent is aware of its neighbors at each time step.
- **T3.5. Training and testing the DQN within the multi-agent dynamic topology environment.**
  As in T3.3, use a trial-and-error approach to optimize the hyperparameters for the dynamic environment. Compare the testing results with a benchmark method.
- **M3. Publications.** Present at least one paper at a relevant conference and publish at least one article in a reputable journal, outlining the problem, methodology, and results of the multi-agent RL-based decentralized routing in dynamic topologies. The publications should demonstrate that agent collaboration outperforms single-agent RL methods, such as the model obtained in WP2, particularly in terms of traffic distribution and packet prioritization.

- **M4. Simulator.** Develop a Python-based open-source simulator on GitLab with configurable network topologies (static or dynamic), customizable satellite characteristics, flexible reinforcement learning methods (single-agent or multi-agent), and adjustable traffic demands.

WP4. Thesis

- **T4.1. Related works.**
  Conduct a comprehensive review of existing literature relevant to the research topic to establish the state of the art and identify research gaps.
- **T4.2. Research Plan writing.**
  Develop a detailed research plan outlining the related works in T4.1, the objectives, and the tasks to accomplish them, together with the timeline and milstones.
- **T4.3. Thesis writing.**
  Write the PhD thesis, presenting the research problem, methodology, results, and conclusions in a structured and coherent document.
- **T4.4. Corrections.**
  Revise the thesis based on feedback from the supervisor, co-supervisors, and tutor to ensure accuracy and completeness.
- **T4.5. Viva preparation.**
  Prepare for the oral defense of the thesis, including a presentation of the research and responses to potential questions.
- **T4.6. Research carrier planning.**
  Develop a plan for the future research career, identifying potential opportunities for postdoctoral work, publications, and collaborations.
- **M5. Publication.**
  Publish a journal article based on the survey of related works analyzed in T4.1, highlighting the significance of FSS and the need for specialized routing protocols to address challenges posed by dynamic topologies and high communication demands in 6G networks.

A timeline work plan to complete each WP with its associated tasks and milestones is detailed in the Gantt diagram in Figure 3.

VI. Data Management Plan

The Data Management Plan (DMP) outlines the strategies and processes for efficiently organizing, sharing, and preserving the data used in this thesis for training, validating, and testing ML models.

For WP1, the training data for the SL model will be synthetic and self-generated, simulating a heterogeneous environment with customizable satellite configurations and orbital parameters, distributed across multiple constellations. To test the SL model for encounter prediction, publicly available datasets from Celestrak will be utilized. These datasets consist of Two-Line Element (TLE) sets, which provide regularly updated information on the current positions and orbits of active satellites around Earth.

In WP2 and WP3, synthetic traffic data will be generated based on historical traffic patterns to train and evaluate the RL-based routing protocol, particularly to simulate and manage congestion states.

Throughout the research, all data will be stored on i2CALC, the local server at the i2CAT Foundation, which provides 11 TB of shared storage with the Distributed Artificial Intelligence (DAI) department. Upon completion of the research,

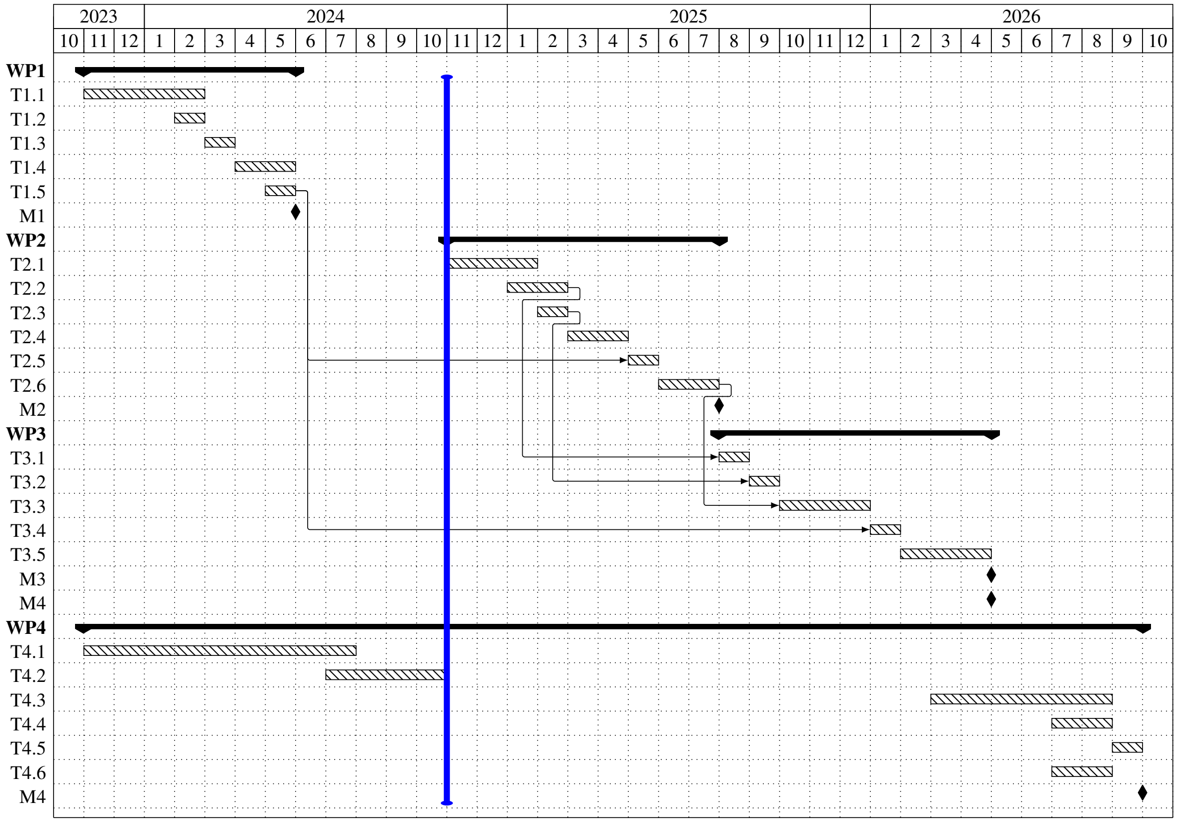


Fig. 3. Gantt diagram. Vertical blue line indicate the current month.

the data will be retained on the server for at least five years, with public repositories considered for long-term storage and accessibility.

DMP aligns with the FAIR principles, ensuring that the data is Findable, Accessible, Interoperable, and Reusable. All datasets will be properly documented with descriptive metadata, including their origin, purpose, and usage instructions, and will feature unique identifiers (e.g., DOIs) to ensure they are easily findable. To ensure accessibility, data will be published to open-access repositories like Zenodo or UPC's repository. Additionally, some open-source code and simulation environments will be uploaded to GitHub. Standardized formats such as TXT, CSV, JSON, and HDF5 will ensure interoperability and facilitate seamless integration with other systems. To enhance reusability, metadata will include clear citation guidelines and instructions, promoting adherence to scientific standards, maximizing visibility and research impact, and fostering broader collaboration.

VII. Training Plan

Some training activities are planned to be followed during the 3-year PhD program. Some additional activities may be added depending on the needs that might arise during the research period.

- Activity 1 [1st year]: Scientific writing, by Gavin Lucas. Provider: i2CAT Foundation. Dedicated time: 16 hours.
- Activity 2 [1st year]: Referencing and avoiding plagiarism. Provider: i2CAT Foundation. Dedicated time: TBD.
- Activity 3 [1dt year]: Effective communication, by Imma Ripoll. Provider: i2CAT Foundation. Dedicated time: 10 hours.
- Activity 4 [1st year]: Machine Learning Engineering for Production (MLOps), by Andrew Ng. Provider: Coursera. Dedicated time: 65 hours.
- Activity 5 [1st year]: Public Speaking and Presentations, by Nacho Téllez. Provider: i2CAT Foundation. Dedicated time: 20 hours.
- Activity 6 [2nd year]: Research communication. Provider: i2CAT Foundation.
- Activity 7 [2nd year]: Time and Stress Management. Provider: i2CAT Foundation.
- Activity 8 [2nd and 3rd years]: Conducting laboratory sessions at UPC as a teaching assistant, focused on my area of expertise, mainly AI and programming.
- Activity 9 [3rd year]: Knowledge transfer, the protection of research results, entrepreneurship, and technological start-ups. Provider: i2CAT Foundation.
- Activity 10 [3rd year]: Post-doctoral research programs and career advice.

References

[1] D. T. Kelso. (1985) Celestrak. [Online]. Available: https://celestrak.org/

[2] C. Araguz López, “In pursuit of autonomous distributed satellite systems,” 2019.

[3] R. J. Boain, “Ab-cs of sun-synchronous orbit mission design,” 2004.

[4] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, “Cellular, wide-area, and non-terrestrial iot: A survey on 5g advances and the road toward 6g,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1117–1174, 2022.

[5] S. Cakaj, “The parameters comparison of the “starlink” leo satellites constellation for different orbital shells,” *Frontiers in Communications and Networks*, vol. 2, p. 643095, 2021.

[6] Y. Henri, “The oneweb satellite system,” in *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*. Springer, 2020, pp. 1091–1100.

[7] P. Wang, H. Li, B. Chen, and S. Zhang, “Enhancing earth observation throughput using inter-satellite communication,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 7990–8006, 2022.

[8] A. Golkar and I. L. i Cruz, “The federated satellite systems paradigm: Concept and business case evaluation,” *Acta Astronautica*, vol. 111, pp. 230–248, 2015.

[9] R. Akhtyamov, I. L. i Cruz, H. Matevosyan, D. Knoll, U. Pica, M. Lisi, and A. Golkar, “An implementation of software defined radios for federated aerospace networks: Informing satellite implementations using an inter-balloon communications experiment,” *Acta Astronautica*, vol. 123, pp. 470–478, 2016.

[10] O. von Maurich and A. Golkar, “Data authentication, integrity and confidentiality mechanisms for federated satellite systems,” *Acta Astronautica*, vol. 149, pp. 61–76, 2018.

[11] M. Giordani and M. Zorzi, “Non-terrestrial networks in the 6g era: Challenges and opportunities,” *IEEE network*, vol. 35, no. 2, pp. 244–251, 2020.

[12] X. Zhu and C. Jiang, “Integrated satellite-terrestrial networks toward 6g: Architectures, applications, and challenges,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 437–461, 2021.

[13] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, “The roadmap to 6g: Ai empowered wireless networks,” *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[14] M. Banafaa, I. Shayea, J. Din, M. H. Azmi, A. Alashbi, Y. I. Daradkeh, and A. Alhammadi, “6g mobile communication technology: Requirements, targets, applications, challenges, advantages, and opportunities,” *Alexandria Engineering Journal*, vol. 64, pp. 245–274, 2023.

[15] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, p. 269–271, dec 1989.

[16] C. Cheng, R. Riley, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves, “A loop-free extended bellman-ford routing protocol without bouncing effect,” *SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 4, p. 224–236, aug 1989.

[17] L. Franck and G. Maral, “Static and adaptive routing in isl networks from a constellation perspective,” *International journal of satellite communications*, vol. 20, no. 6, pp. 455–475, 2002.

[18] K. Tsai and R. P. Ma, “Darting: a cost-effective routing alternative for large space-based dynamic-topology networks,” in *Proceedings of MILCOM'95*, vol. 2. IEEE, 1995, pp. 682–686.

[19] D. Fischer, D. Basin, K. Eckstein, and T. Engel, “Predictable mobile routing for spacecraft networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1174–1187, 2012.

[20] J. A. Fraire and J. M. Finochietto, “Design challenges in contact plans for disruption-tolerant satellite networks,” *IEEE Communications Magazine*, vol. 53, no. 5, pp. 163–169, 2015.

[21] W. Su, J. Malaer, O. Cho, and K. Suh, “Using mobility prediction to enhance network routing in leo crosslink network,” 2019, international Astronautical Congress (IAC).

[22] J. A. Ruiz-De-Azua, V. Ramírez, H. Park, A. C. AUG, and A. Camps, “Assessment of satellite contacts using predictive algorithms for autonomous satellite networks,” *IEEE access*, vol. 8, pp. 100 732–100 748, 2020.

[23] G. Casadesus and E. Alarcón, “Toward autonomous cooperation in heterogeneous nanosatellite constellations using dynamic graph neural networks,” in *International Astronautical Congress*, 2023.

[24] J. A. R. de Azúa, A. Calveras, and A. Camps, “Internet of satellites (iosat): Analysis of network models and routing protocol requirements,” *IEEE access*, vol. 6, pp. 20 390–20 411, 2018.

[25] T. Taleb, D. Mashimo, A. Jamalipour, N. Kato, and Y. Nemoto, “Explicit load balancing technique for ngeo satellite ip networks with on-board processing capabilities,” *IEEE/ACM transactions on Networking*, vol. 17, no. 1, pp. 281–293, 2008.

[26] G. Song, M. Chao, B. Yang, and Y. Zheng, “Tlr: A traffic-light-based intelligent routing strategy for ngeo satellite ip networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3380–3393, 2014.

[27] I. F. Akyildiz, E. Ekici, and M. D. Bender, “Mlsr: A novel routing algorithm for multilayered satellite ip networks,” *IEEE/ACM Transactions on networking*, vol. 10, no. 3, pp. 411–424, 2002.

[28] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (aodv) routing,” Tech. Rep., 2003.

[29] T. Clausen and P. Jacquet, “Optimized link state routing protocol (olsr),” Tech. Rep., 2003.

[30] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, “Contact graph routing in dtn space networks: overview, enhancements and performance,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[31] O. De Jonckère and J. A. Fraire, “A shortest-path tree approach for routing in space networks,” *China Communications*, vol. 17, no. 7, pp. 52–66, 2020.

[32] A. Lambora, K. Gupta, and K. Chopra, “Genetic algorithm-a literature review,” in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 380–384.

[33] F. Tang, B. Mao, Y. Kawamoto, and N. Kato, “Survey on machine learning for intelligent end-to-end communication toward 6g: From network access, routing to traffic control and streaming adaption,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1578–1598, 2021.

[34] F. Geyer and G. Carle, “Learning and generating distributed routing protocols using graph-based deep learning,” in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, 2018, pp. 40–45.

[35] B. Soret, I. Leyva-Mayorga, F. Lozano-Cuadra, and M. D. Thorsager, “Q-learning for distributed routing in leo satellite constellations,” in *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 2024, pp. 208–213.

[36] Y. Huang, W. Shufan, K. Zeyu, M. Zhongcheng, H. Huang, W. Xiaofeng, A. J. Tang, and X. Cheng, “Reinforcement learning based dynamic distributed routing scheme for mega leo satellite networks,” *Chinese Journal of Aeronautics*, vol. 36, no. 2, pp. 284–291, 2023.

[37] P. Zuo, C. Wang, Z. Yao, S. Hou, and H. Jiang, “An intelligent routing algorithm for leo satellites based on deep reinforcement learning,” in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[38] G. Xu, Y. Zhao, Y. Ran, R. Zhao, and J. Luo, “Spatial location aided fully-distributed dynamic routing for large-scale leo satellite networks,” *IEEE Communications Letters*, vol. 26, no. 12, pp. 3034–3038, 2022.

[39] J. Liu, B. Zhao, Q. Xin, J. Su, and W. Ou, “Drl-er: An intelligent energy-aware routing protocol with guaranteed delay bounds in satellite mega-constellations,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 2872–2884, 2020.

[40] C. Wang, H. Wang, and W. Wang, “A two-hops state-aware routing strategy based on deep reinforcement learning for leo satellite networks,” *Electronics*, vol. 8, no. 9, p. 920, 2019.

[41] H. Zhang, H. Tang, Y. Hu, X. Wei, C. Wu, W. Ding, and X.-P. Zhang, “Heterogeneous mean-field multi-agent reinforcement learning for communication routing selection in sagi-net,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[42] X. Shi, P. Ren, and Q. Du, “Heterogeneous satellite network routing algorithm based on reinforcement learning and mobile agent,” in *2020 IEEE Globecom Workshops (GC Wkshps*. IEEE, 2020, pp. 1–6.

[43] Y. Shi, W. Wang, X. Zhu, and H. Zhu, “Low earth orbit satellite network routing algorithm based on graph neural networks and deep q-network,” *Applied Sciences*, vol. 14, no. 9, p. 3840, 2024.

[44] S. Kaviani, B. Ryu, E. Ahmed, K. A. Larson, A. Le, A. Yahja, and J. H. Kim, “Robust and scalable routing with multi-agent deep reinforcement learning for manets,” *arXiv preprint arXiv:2101.03273*, 2021.

[45] A. Ghaffari, “Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms,” *Wireless Networks*, vol. 23, pp. 703–714, 2017.

[46] J. A. Ruiz-de Azúa, C. Araguz, A. Calveras, E. Alarcón, and A. Camps, “Towards an integral model-based simulator for autonomous earth ob-

servation satellite networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 7403–7406.

[47] C. Araguz López, J. A. Ruiz De Azúa Ortega, A. M. Calveras Augé, A. J. Camps Carmona, and E. J. Alarcón Cot, "Simulating distributed small satellite networks: A model-based tool tailored to decentralized resource-constrained systems," in *Proceedings of the 70th International Astronautical Congress*, 2019, pp. 1–10.

[48] J. A. Ruiz-De-Azua, A. Calveras, and A. Camps, "A novel dissemination protocol to deploy opportunistic services in federated satellite systems," *IEEE access*, vol. 8, pp. 142 348–142 365, 2020.