UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2025-129
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 14/11/2025 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Loïck, Pierre CHOVET
Born on 19 December 1998 in Essey-lès-Nancy, (France)

DECENTRALIZED AUTONOMY IN PLANETARY MULTI-ROBOT TEAMS: TRUSTWORTHY TASK NEGOTIATION, RESILIENT COMMUNICATION, AND ADAPTIVE OPTIMIZATION

Dissertation defence committee

Dr Miguel Angel Olivares-Mendez, dissertation supervisor
Associate Professor, Université du Luxembourg

Dr Kazuya Yoshida
Professor, Space Robotic Lab, Tohoku university

Dr Gilbert Fridgen, Chairman
Professor, Université du Luxembourg

Dr Carlos Pérez del Pulgar
Professor, Universidad de Malaga

Dr Jose Luis Sanchez-Lopez
Research Scientist, Université du Luxembourg

Affidavit / Statement of originality

I declare that this thesis:

- is the result of my own work. Any contribution from any other party, and any use of generative artificial intelligence technologies have been duly cited and acknowledged;
- is not substantially the same as any other that I have submitted, and;
- is not being concurrently submitted for a degree, diploma or other qualification at the University of Luxembourg or any other University or similar institution except as specified in the text.

With my approval I furthermore confirm the following:

- I have adhered to the rules set out in the University of Luxembourg's Code of Conduct and the Doctoral Education Agreement (DEA)[1], in particular with regard to Research Integrity.
- I have documented all methods, data, and processes truthfully and fully.
- I have mentioned all the significant contributors to the work.
- I am aware that the work may be screened electronically for originality.

I acknowledge that if any issues are raised regarding good research practices based on the review of the thesis, the examination may be postponed pending the outcome of any investigation of such issues. If a degree was conferred, any such subsequently discovered issues may result in the cancellation of the degree.

Approved on 2025-10-10

[1] If applicable (DEA is compulsory since August 2020)

Acknowledgements

THIS dissertation is the result of a tumultuous journey, an exploration in uncharted territories only guided by my own light. Here I am, as this thesis is taking its final shape, looking for the right words to express these four years and all the people that mattered and impacted it achievement.

While he was still a master student when I started, **Dave** has been one of the pillars of my progress. I cannot count the number of hours he invested in reviewing every one of my papers along with this work, correcting my spelling and typography, and always suggesting the most relevant changes. His sharp but benevolent humour was a real source of motivation to come to the office, and I can't count the number of passionate discussions we had, exploring science, new opportunities, expanding my field of view, but also offering new ideas to grow.

Arriving slightly later as my intern, **Gabriel** soon rejoined us as a PhD student, making us the trio of planetary robotics. He was a source of motivation and dedication to his work, always proposing innovative ideas and a new point of view on ideas. He would drag me out to the most unexpected places, such as a nuclear reactor, a radioactive mine or an RC car race.

With three different approaches to work, we managed to balance each other and reach a reasonable work–life rhythm... at least as much as possible. Although our work was not directly related, we supported each other throughout these four years.

I also want to express my sincere gratitude to **Miguel** for his unwavering support throughout the thesis, especially during its darkest times. He did his best to secure me an exchange stay in Japan and helped reorient my thesis when it was needed most. I want to thank **Abhishek** for his support through half of my thesis. Despite personal difficulties, he helped me forge a strong backbone for this work. I thank Professor **Yoshida** for welcoming me to Tohoku University and allowing me to spend three months in Japan, a childhood dream come

true. There, I had the chance to work on homemade robots and even run experiments on Sendai's beach. I would like to thank **Renan** for our common work on the FiReSpARX project, but also for igniting that flame in me that led me to make my future project more concrete.

But a thesis is not all about the work, its about keeping the right morale through the 4 years, like a marathon, most is in the head. And while she was running her own race, **Nolwenn** was still undefectively here for me every single day of this thesis, especially when I needed someone. One of the dearest people I met over the course of these four years is **Evelin**. While we couldn't meet often, she quickly became one of my closest friends, helping me to persevere through the darkest times and evolve, allowing me to see things from a different perspective. When it's about friends, how to not thank **Jeremy**, **Simon**, **Jolan**, **Yannick**, **Loraly**, **Aurel**, **Stella**, **Laurent**, **Salomé** for the parties, the board games nights and all the activities we have done. It's these kind of things that helps you refresh your mind and keep up. To my parents **Karine and Oliver** and brother **Matteo**, who were here to support me even in the most complicated moments for all of us. Finally, I want to thank **Rishekesh**, who made my future adventure possible, through his work and dedication, the future is very promising.

I'm finally putting the last words of this thesis, ready to start my next adventure, in the same team, with different goals, but with an ever-growing passion and enthusiasm and a bright future ahead.

Loïck Chovet

Pour Lydia,

dont le sourire ne nous quittera jamais.

Pour Jean-Pierre,

qui a toujours su stimuler mon esprit critique.

Index

List of Figures

List of Tables

Glossary

AODV Ad-hoc On-demand Distance Vector. 37–39, 166

B.A.T.M.A.N Better Approach To Mobile Ad-hoc Networking. 37–39

CADRE Cooperative Autonomous Distributed Robotic Exploration. 11, 31

DDS Data Distribution Service. xiv, xvii, 28, 29, 36, 119, 120, 125, 126, 128, 129, 135, 138, 139, 142, 145, 150, 153, 154, 160–162

DEM Digital Elevation Map. ix, 17, 19, 167, 170, 173–175, 192

DLT Distributed Ledger Technology. vii, viii, 32, 33, 68–70, 72, 96, 98, 99, 112, 113, 193

ESA European Space Agency. xii, 11, 18, 44, 55, 56, 64, 65, 124–126

ESRIC European Space Resources Innovation Centre. xii, 11, 18, 44, 55, 64, 65, 124–126

FBA Force Based Approach. ix, 41, 176, 178, 180, 181, 192

GNSS Global Navigation Satellite Systems. 166, 185

GPU Graphical Processing Unit. 60, 185

HWMP Hybrid Wireless Mesh Protocol. 38, 39, 128, 133

HWMP+ Hybrid Wireless Mesh Protocol +. 38, 39, 120, 122, 127, 128, 133, 134, 145, 162, 166, 182

IMU Inertial Measurement Unit. 48, 51

IPFS InterPlanetary File System. 76, 78, 86, 95, 98, 105

ISRU In-Situ Resources Utilisation. 7, 11, 15, 31, 33, 44, 65, 68–70, 72, 73, 80, 82, 84, 92, 98, 99, 104, 112, 165

KPI Key performance indicator. 135, 138, 139, 147, 152

LOS Line-Of-Sight. 18, 19, 40, 61, 131, 135, 137–142, 144, 146, 161, 167

LRV Lunar Roving Vehicle. 9

MANET Mobile Ad-Hoc Networks. 34, 37

MQTT Message Queuing Telemetry Transport. 161

MRS Multi-Robot Systems. vii, xi–xiii, xvi, 10–13, 15, 17–19, 22–37, 39, 41, 44, 58–60, 62, 65, 66, 68–72, 79–82, 88–92, 94, 107, 111, 117, 119, 120, 125–127, 129–132, 146, 162–165, 167, 175, 182, 186–188, 191–195

MT Multi-Task Robots. 86

NFT non-fungible tokens. vii, 33, 76, 78, 81, 85, 86, 95, 98, 105, 113

NLOS Non Line-Of-Sight. 18, 131, 135, 137–142, 144, 146, 161

OLSR Optimized Link State Routing. 38, 166

OSI Open Systems Interconnection. xi, 35–37, 119, 134, 166

PER Packet Error Rate. 38

QOS Quality of Service. 11, 22, 28, 36, 125, 126, 148

REALMS Resilient Exploration And Lunar Mapping System. xi, xii, 44, 46–56

REALMS2 Resilient Exploration And Lunar Mapping System 2. viii, xii, xiii, 58–60, 63–65, 68, 70, 120, 123, 124, 126

RERR Route Error. 37

RMW ROS MiddleWare. xiv, xv, xvii, 128–130, 134, 145, 147–150, 152–162

ROI region of interest. 45, 55, 58

ROS Robot Operating System. xi, 11, 28, 29, 60, 119, 130, 183–185

ROS 2 Robot Operating System version 2. viii, xi, xvii, 11, 15, 17, 18, 22, 28–31, 35, 36, 44, 59–62, 66, 119, 120, 124–131, 134, 135, 137, 144–146, 148, 151, 157, 158, 160–162, 164, 182, 183, 185, 194

RREQ Route Request. 37

RRER Route Reply. 37

RSSI Received Signal Strength Indicator. 39

RTK Real-Time Kinematic. 166, 185

SAS Space Applications Services. 63

SLAM Simultaneous Localisation And Mapping. 60, 185

SOA Service Oriented Approach. 25

SP Service Provider. xiii, xvii, 70, 71, 80, 81, 88–90

ST Single Task Robots. 86

UAV Unmanned Air Vehicule. 39

vSLAM Visual Simultaneous Localisation And Mapping. 48, 49, 51

“The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom.”

— Isaac Asimov

PROLOGUE

Chapter 1

Introduction

"The Earth is the cradle of humanity, but one cannot live in the cradle forever"

– Konstantin Tsiolkovsky

1.1 Why Going To The Moon?

1.1.1 Historical context of Lunar exploration

The first steps of Lunar exploration began as a geopolitical race between the United States and the Soviet Union during the Cold War. Known as the Space Race, this period saw intense competition in achieving significant scientific and technological milestones, with the first manned Moon landing as the goal. The Soviet Union achieved most of the early milestones in this race. The first dog in space, Laika, in November 1957, the first man in space, Yuri Gagarin, in April 1961, and the first woman in space, Valentina Tereshkova, reaching orbit in June 1963. With the Luna program, the USSR landed on the Moon in 1959 and was the first to land a spacecraft in 1966. However, the NASA Apollo program focused on crew missions, culminating on July 20, 1969, with the first human step on the Moon.

Although the Apollo 11 landing in July 1969 marked a historic milestone, it was only the beginning of crewed exploration on the lunar surface. Over the next three years, five additional

Figure 1.1: Left: Valentina Tereshkova, the first woman in space (Vostok 6, 1963). Right: Neil Armstrong taking the first step on the Moon during Apollo 11 (1969).

Apollo missions (12, 14, 15, 16, and 17) successfully brought astronauts to the Moon. Each mission expanded scientific objectives: deploying seismometers and retroreflectors, conducting geological surveys, and collecting over 380 kilograms of lunar rock and soil samples. Later missions introduced the Lunar Roving Vehicle, which allowed astronauts to travel further from the landing site and conduct more complex fieldwork.

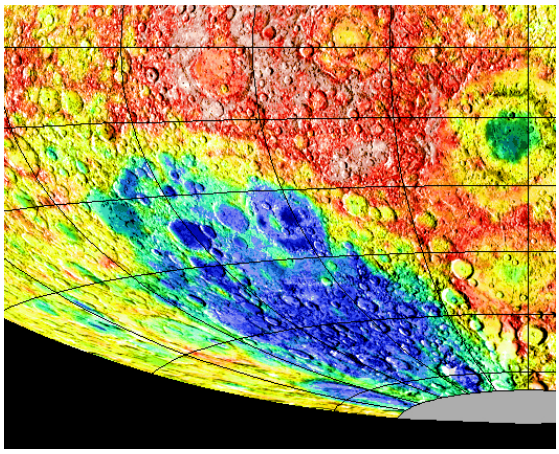
Despite these achievements, sustaining crewed lunar exploration presented mounting challenges. The Apollo program was extremely expensive, consuming billions of dollars annually at its peak. With the United States facing domestic pressures such as the Vietnam War and social unrest, public and political support for large-scale space spending declined sharply in the 1970s. In addition, once the symbolic goal of “first footsteps on the Moon” had been achieved, the missions began to receive less media coverage, diminishing their impact on public opinion.

Robotic missions, by contrast, offered a more cost-effective and sustainable path forward. They could operate for months or years without life-support requirements, withstand harsher environments, and deliver high-quality data through advanced sensors. The Soviet Luna program, followed later by NASA’s Surveyor and Lunar Orbiter series, demonstrated the

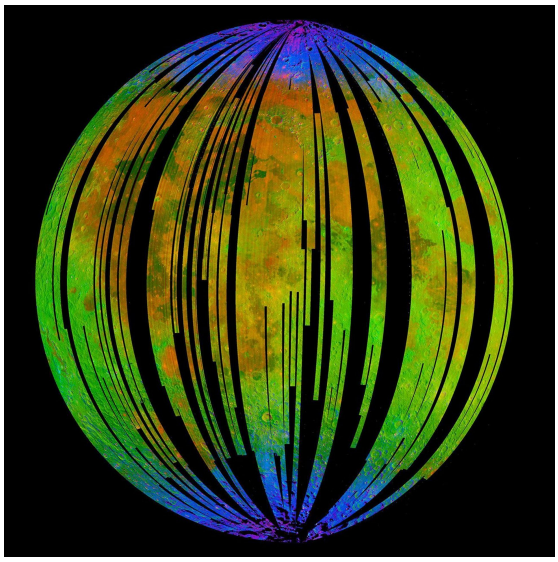
value of robotic exploration in mapping landing sites, testing surface operations, and returning samples. As space science matured, robotics also opened possibilities for international participation and technological innovation without the immense risks associated with human spaceflight.

This transition did not end aspirations for human exploration but rebalanced priorities toward science, automation, and long-term sustainability. It laid the foundation for modern lunar exploration, where robotic systems conduct reconnaissance, resource prospecting, and technology demonstrations, preparing the ground for the eventual human return within a broader, international framework.

1.1.2 Current Motivations for Lunar Exploration



(a) Global multispectral mosaic of the Moon from Clementine UV-Vis data, showing mineralogical and albedo variations.



(b) Distribution of surface water ice at the Moon's poles, as detected by the Moon Mineralogy Mapper (M3) on Chandrayaan-1 [1].

Figure 1.2: Remote sensing data revealing the mineral composition and volatile distribution on the lunar surface.

Although interest in crewed lunar exploration declined after the Apollo era, several robotic missions continued to expand scientific knowledge of the Moon. Orbital missions such as NASA's Clementine (1994) and Lunar Prospector (1998) renewed global mapping efforts and

hinted at the possible presence of polar ice ([2], as visible in Fig. 1.2a). This momentum culminated in the early 2000s with Chandrayaan-1 (India, 2008) and NASA's Lunar Reconnaissance Orbiter (LRO, 2009), which produced detailed topographic and mineralogical maps.

This discovery was later reinforced in 2009 by NASA's LCROSS mission, which detected water vapour and hydroxyl in the debris plume generated from a targeted impact [3], reframing the Moon as a potential resource site rather than a barren world. In 2019, the Moon Mineralogy Mapper (M3) onboard India's Chandrayaan-2 spacecraft [1] definitely proved the existence of surface water ice as visible in Fig. 1.2b. Global perception of the Moon shifted to a potential resource hub to support long-duration exploration and the development of a lunar economy.

The implications for In-Situ Resources Utilisation are deep. The water ice could be harvested to sustain life support systems, produce breathable oxygen, and be split into hydrogen and oxygen to serve as rocket propellant. As hydrogen is usable as a rocket propellant, this would transform the Moon into a fuel depot for deep-space missions, which would critically reduce the cost of interplanetary travel.

Another factor in lowering the cost of lunar and deep-space missions is the rise of reusable launch systems. Traditionally, access to space was prohibitively expensive, with launch costs exceeding $70000 per kilogram during the Space Shuttle era. However, the rise of reusable launch vehicles—most notably SpaceX's Falcon 9—has critically changed this ecosystem. In 2023, the cost of launching to low Earth orbit (LEO) with a Falcon 9 is approximately $2700 per kilogram, representing a reduction of more than an order of magnitude [4]. This drastic cost reduction is revolutionising space logistics, making cargo delivery, orbital infrastructure, and lunar surface operations far more economically viable. It also paves the way for a new space economy in which commercial actors, resource utilisation, and data-driven services play a central role in lunar development.

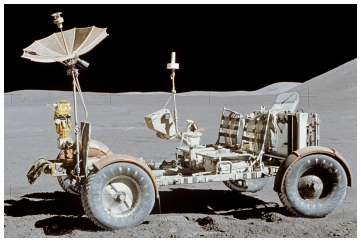
The new space economy is driven not only by cost reductions and technology but also by strategic, scientific, and economic motivations. One of the first is the desire for geopolitical independence and strategic presence. More nations aim to land on the Moon as proof of independence, seeking to break free from the dominant political schemes. Beyond demon-

strating geopolitical presence, many nations view lunar missions as a way to strengthen domestic innovation ecosystems and gain access to the emerging cislunar economy. Developing autonomous landing, navigation, and resource-extraction technologies provides both scientific prestige and commercial opportunity. The lunar south pole, with its permanently shadowed regions and potential water ice, is emerging as a key strategic and economic zone [5]. Initiatives like the Artemis Accords and China's International Lunar Research Station reflect parallel visions for setting normative and operational standards on the Moon [6, 7].

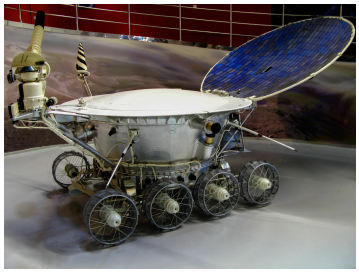
Economic opportunity is another major driver. As detailed previously, the Moon is host to significant amounts of water ice but also oxygen-containing regoliths, and trace quantities of economically interesting elements such as rare earth elements and Helium-3 [8]. Analyses of Apollo regolith samples indicate ^{3}He concentrations in the range of 2–10 ppb by mass, with the highest values, around 9.5 ppb, measured in titanium-rich mare basalts from Mare Tranquillitatis [8]. These concentrations are strongly correlated with ilmenite abundance, suggesting that high-Ti mare regions host the largest ^{3}He reserves, while remote sensing by Chang'e-1 supports this spatial distribution. Although the total lunar inventory has been estimated at roughly 10^5–10^6 tonnes, the economically recoverable fraction remains highly uncertain. Nevertheless, such resources, combined with the potential to extract metals and oxygen from regolith, could underpin future In-Situ Resources Utilisation (ISRU) operations and support a sustainable cis-lunar economy.

These resources could support both local infrastructure and broader space operations. ISRU would reduce the mass launched from Earth and potentially create new markets in lunar construction, fuel production, and material processing. Researchers have explored metal extraction (e.g. iron, aluminium, titanium) from regolith and the use of solidified regolith for structural elements [9]. Various studies show that sintered or bonded regolith can reach compressive strengths comparable to or exceeding terrestrial concrete, making it a candidate for load-bearing habitat walls [10]. Additionally, compressed regolith layers can provide radiation shielding and thermal insulation, helping stabilize interior temperatures and protect against cosmic rays [11].

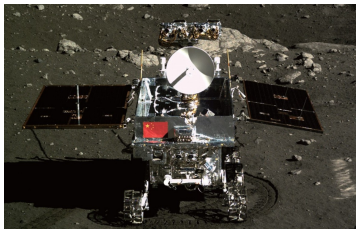
Lunar exploration is also motivated by infrastructure development. The sustained pres-



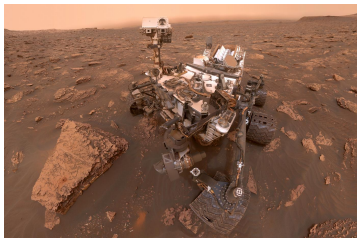
(a) Apollo Lunar Roving Vehicle (NASA, 1971).



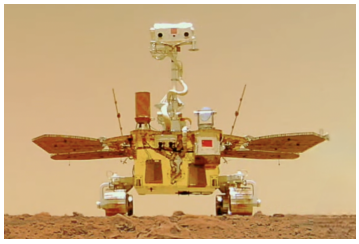
(b) Lunokhod 1 (Soviet Union, 1970).



(c) Yutu rover from Chang'e 3 (China, 2013).



(d) Perseverance rover (NASA, 2021).



(e) Zhurong rover from Tianwen-1 (China, 2021).

(f) Canadarm2 on the ISS (Canada, from 2001).

Figure 1.3: Examples of lunar and planetary robotic systems from different space agencies.

ence on the Moon requires logistic systems such as power grids, mobility platforms, refuelling stations, and communication relays. These enable science missions and commercial operations that range from robotic prospecting to human habitats. The construction of such infrastructure, through NASA's Artemis Base Camp, the Lunar Gateway, and private ventures like Intuitive Machines and Astrobotic, is seen as the backbone of a cislunar economy.

Scientific research remains a critical motivation. The Moon serves as a natural laboratory. Its lack of atmosphere and geological activity preserves ancient surfaces and records while offering unique environmental conditions. Furthermore, the lunar far side provides a unique radio-quiet environment ideal for low-frequency astronomy [12].

1.2 What Do Robots Bring?

The story of robotic systems is deeply tied to space exploration. In 1957, Sputnik was the first man-made system in orbit. Despite its limited functionality, which was restricted to sending a single ping, it was effectively the first autonomous system in space. Even early manned

missions, such as the Apollo lunar landings, were supported by robotic components, including the Lunar Roving Vehicle (LRV) (Fig. 1.3a), a teleoperated system that greatly expanded the mission's range.

As crewed programs receded, robotic lunar missions carried the arc of discovery forward—first through teleoperation, then with growing autonomy. The Soviet Lunokhod-1/2, visible in Fig. 1.3b, demonstrated months-long surface operations and remote science (imaging, soil mechanics, in-situ X-ray fluorescence), proving that complex traverses and instrument deployment were feasible without crew. China's Yutu-1/2, visible in Fig.1.3c, brought expanded autonomy, with autonomous obstacle avoidance and local planning. It also contributed science with a ground-penetrating radar (GPR), producing the first meter-scale subsurface stratigraphy from the Moon and constraining regolith thickness and basalt flow layering [13]. Yutu-2 is also the first rover to traverse the far side of the moon. India's Pragyan (Chandrayaan-3) delivered in-situ LIBS and APXS measurements at the south-polar region, reporting elemental abundances (including sulfur) directly relevant to ISRU feedstocks and polar resource models.

On Mars, surface missions progressively increased autonomy while transforming planetary science. Sojourner validated mobile manipulation and hazard-aware driving under severe bandwidth/latency, establishing the rover paradigm. Spirit and Opportunity then delivered stratigraphic and mineralogical evidence for past aqueous environments (jarosite, hematite "blueberries," silica deposits), reframing Mars' paleoenvironmental history. Curiosity confirmed an ancient habitable lacustrine system in Gale crater (clays, redox couples, organics) and pioneered routine visual–odometry–based autonav for multi-sol traverses. Perseverance (Fig. 1.3d, [14]) added fault-tolerant autonomy (onboard map building and safe-route selection), began sample caching for return, and demonstrated MOXIE oxygen production—an operational ISRU milestone [15]. China's Zhurong (see Fig.1.3e) independently validated long-range traverse planning and conducted climate/regolith studies on Utopia Planitia, broadening the architectural diversity of surface systems. Collectively, these missions show a clear trend: from teleoperation toward onboard decision-making and sustained surface logistics, with direct implications for multi-robot coordination and ISRU-ready autonomy.

All of those robotics systems were used to explore the surfaces of Mars and the Moon, collect data and perform scientific experiments. Additionally, the Canadarm2 (Fig. 1.3f) on the International Space Station (ISS) demonstrated as a vital part for servicing, assembling and payload operations in orbit.

Robots offer several essential capabilities for space missions; however, traditional single-robot missions face limitations. They lack adaptability as adding more equipment requires increasing the robot's size, weight and consequently its price. For security reasons, the speeds of space robots are strictly limited. As an example, the Curiosity rover is moving at $0.04\,\mathrm{m/s}$. As of mid-August 2025, since its landing in August 2012, Curiosity travelled $35.4\,\mathrm{km}$ [16]. This highlights one of the biggest issues with single-robot systems: coverage capabilities.

1.2.1 The Emergence of Multi-Robot Systems (MRS)

Traditional exploration missions were optimised for targeted scientific analysis rather than broad-scale prospection. A single rover can study a site in depth but cannot simultaneously survey multiple regions, map volatile deposits, or sample geological diversity over large distances. Moreover, communication latency and limited data bandwidth constrain real-time control, forcing conservative operations that reduce temporal efficiency. These factors limit not only the quantity of scientific return but also the capacity to identify and evaluate in-situ resources essential for sustained human or robotic presence.

To address these limitations, the field of Multi-Robot Systems is focusing on space missions research. MRS architectures offer strong advantages:

- **Price reduction**: In MRS, each robot can be more specialised, simpler, hence cheaper
- **Coverage**: Despite their slow speed, each robot in a MRS can explore its own region of interest or even collaborate
- **Specialisation**: Some robots can be dedicated to very specific tasks, ensuring perfectly optimised robots and a wider array of possible capabilities with a lower cost.

- **Redundancy**: As the system is composed of various robots, we avoid scenarios where one component is critical, leading to the failure of the whole mission.

These benefits are particularly important for ISRU, where robots must explore, map, and analyse the lunar or Martian surface collaboratively. Despite these benefits, no real multirobot missions landed on any astral body. However, Missions are starting to be planned such as NASA's Cooperative Autonomous Distributed Robotic Exploration (CADRE) project. The CADRE mission will deploy a group of small, autonomous rovers to the lunar surface to demonstrate coordinated exploration and task sharing without direct human control. CADRE aims to prove that decentralised robotic teams can efficiently map and adapt to unknown terrain, setting the stage for future scalable ISRU missions. In parallel, Challenges such as the ESA-ESRIC Space Resources Challenge are pioneering these ideas, demonstrating the viability and importance of decentralised MRS in practice, while leading to the development of innovative space MRS.

To support this transition, new software architectures and networking paradigms are being adopted. A key enabler is the **ROS 2**, which provides a middleware framework inherently designed for decentralised systems. Unlike its predecessor ROS, ROS 2 is pushed by the industry and institutions like NASA as a robotic middleware transportable to any proprietary application. It removes the need for a central master node, relying instead on DDS-based discovery and Quality of Service (QOS) policies, allowing autonomous agents to communicate over dynamic and constrained networks—a crucial feature for off-world deployments.

However, effective coordination also depends on the ability of robots to remain connected in extreme environments. Here, **mesh networks** have emerged as a robust solution. In such networks, each robot serves as both an agent and a relay node, dynamically adjusting to network topology changes. Mesh topologies offer redundancy, fault tolerance, and adaptability, especially when using protocols such as HWMP+, AODV, or BATMAN. When combined with ROS 2, mesh networks enable autonomous robots to share data, synchronize plans, and execute tasks collectively.

Together, **ROS 2 and mesh networking form a critical backbone** for resilient, scalable MRS deployments, where each agent operates semi-independently yet cooperatively within a

dynamically evolving communication topology.

With growing interest, successful demonstrations of decentralised MRS and additionnal cost reduction, we expect that future space missions will include various multi-robot teams across the Moon, Mars and other celestial bodies. These various MRS will be owned by different organisations, such as space agencies, private companies, or research institutions. Although these MRS could operate independently from each other, some might want to collaborate, to share tasks, resources, or knowledge. This introduces a complex mechanism called **Coopetition**, where robotic agents both compete and collaborate depending on mission goals and economic incentives.

In the following section we will present a plausible scenario based on these findings and identify the research questions that need to be assessed in such future.

1.3 Scenario and research questions

The growing awareness of these lunar resources has not only renewed scientific curiosity but also sparked strong industrial interest. Space agencies now view the Moon as a technological testbed and a stepping stone toward Mars, while private companies envision commercial opportunities in transportation, communication, construction, and resource utilisation. Programs such as NASA's *Commercial Lunar Payload Services* (CLPS) or ESA's *Moonlight* initiative illustrate a gradual shift from government-led exploration toward a mixed ecosystem where public institutions and private actors share both infrastructure and data. In this emerging landscape, access to local resources—water ice for fuel production, oxygen and metals from regolith, or strategic south-polar sites—becomes a driver for collaboration and competition alike, setting the stage for new operational models of lunar activity. Robotic systems lie at the centre of this transition: they are the first agents to map, extract, and process these resources, operating where human presence remains costly or impossible.

In the near future, various companies and institutions will operate decentralised heterogeneous MRS. However, some robots might not be used all the time, and some missions might require capabilities that their robotic team do not own. In classical terms, *cooperation* refers

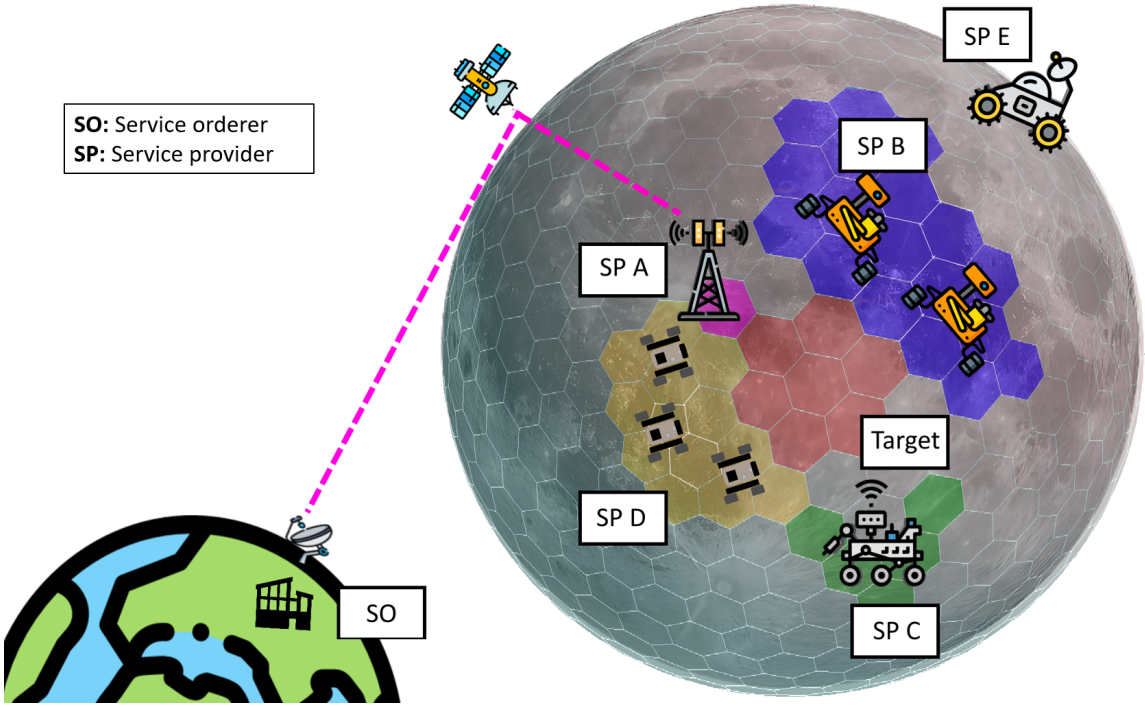


Figure 1.4: Scenario of the thesis, with various MRS on the Moon

to systems jointly pursuing a shared goal, while *collaboration* implies deeper coordination and information exchange among partners to achieve that goal. By contrast, **Coopetition** describes a more complex dynamic where agents both compete and cooperate depending on context. In such an environment, each MRS seeks to maximise its own objectives—scientific return, profit, or operational efficiency—while still engaging in temporary alliances for mutual benefit, such as sharing data, resources, or task execution. This interplay between autonomy, competition, and collaboration defines the emerging paradigm for future space robotics and lunar operations, a **Coopetitive framework**.

Consider the following scenario on the Moon as visible in Fig. 1.4. A research organisation on Earth wants to perform analysis on the red zone and collect samples. While *Company D* offers a team of fast scouting robots, they do lack the possibility to collect samples. *Company C*, on the other hand, controls only one robot with sampling capabilities, but only requesting their help would increase the mapping time. Instead of making a compromise between

mapping speed and capabilities, the research organisation could outsource the task to both companies. In this situation, both companies would benefit while remaining competitors, minimising costs and maximising mission return.

From such a scenario, we identify three arising issues that will be our research questions.

- **Autonomous collaboration and trust in a coopetitive environment.** In a decentralised lunar economy, robots from different organisations must negotiate task execution without a central authority (such as a bank or government on Earth), and without assuming shared incentives. Each robot acts on behalf of its operator and must decide whether to accept, outsource, or reject tasks. This demands new mechanisms to build trust between untrusted entities, establish fair exchanges, and autonomously manage task execution, even when the other party is a direct competitor. Traditional market-based approaches do not address this complexity. Hence, novel coordination strategies must be designed for autonomous agents to interact economically and cooperatively. From this challenge derive the following research question :

  RQ-1: *How do robots collaborate autonomously with trust in a coopetitive system?*

- **Communication efficiency in heterogeneous, dynamic, and constrained environments.**

  Even if robots are capable of autonomous cooperation and economic negotiation, such capabilities are ineffective without a robust and adaptive communication infrastructure. In the context of a decentralised lunar economy, where robots from multiple stakeholders must interact in real-time, communication becomes a critical enabler of collaboration. These robots will operate on unstructured, large-scale terrains with no fixed infrastructure, using heterogeneous hardware and software configurations. Ensuring interconnectivity over mesh networks subject to range limits, topology changes, and strict energy constraints is essential for supporting a coopetitive behaviour. Without efficient communication protocols and resilient network structures, even the most advanced decision-making mechanisms will fail to scale or adapt. Thus, communication is

not only a technical bottleneck but a prerequisite for enabling trusted collaboration in coopetitive MRS.

RQ-2: *How efficiently can heterogeneous Multi-Robot Systems communicate for space missions?*

- **Network topology optimisation for resilience and task success.**
  Building on the previous challenges, if robots establish a coopetitive system over a mesh network. Dynamically maintaining a connected and resilient network as robots move across the environment presents a challenge. Since communication relies on peer-to-peer and relay-based links, a robot moving too far away could become unreachable, disrupting task coordination and data delivery. To prevent such issues, robots must proactively adapt their spatial configuration and routing responsibilities, integrating network awareness into their decision-making. Therefore, the MRS must not only solve their own tasks efficiently but also reason about the collective network structure to ensure long-term operational success. This problem is called Connectivity maintenance.

RQ-3: *How can the robots optimise the network topology?*

1.4 Contributions of the thesis

This thesis proposes a **decentralised and heterogeneous MRS framework for a coopetitive space exploration scenario** to address the three research questions, with the following contributions, detailed in Fig. 1.5.

- **REALMS1 and 2: A decentralised MRS based on ROS 2, demonstrated in a lunar analogue challenge.** This MRS enables efficient multi-robot mapping missions in a lunar environment, with enhanced resilience. It also demonstrates the effectiveness of MRS in a mission of ISRU together with the use of mesh networks.

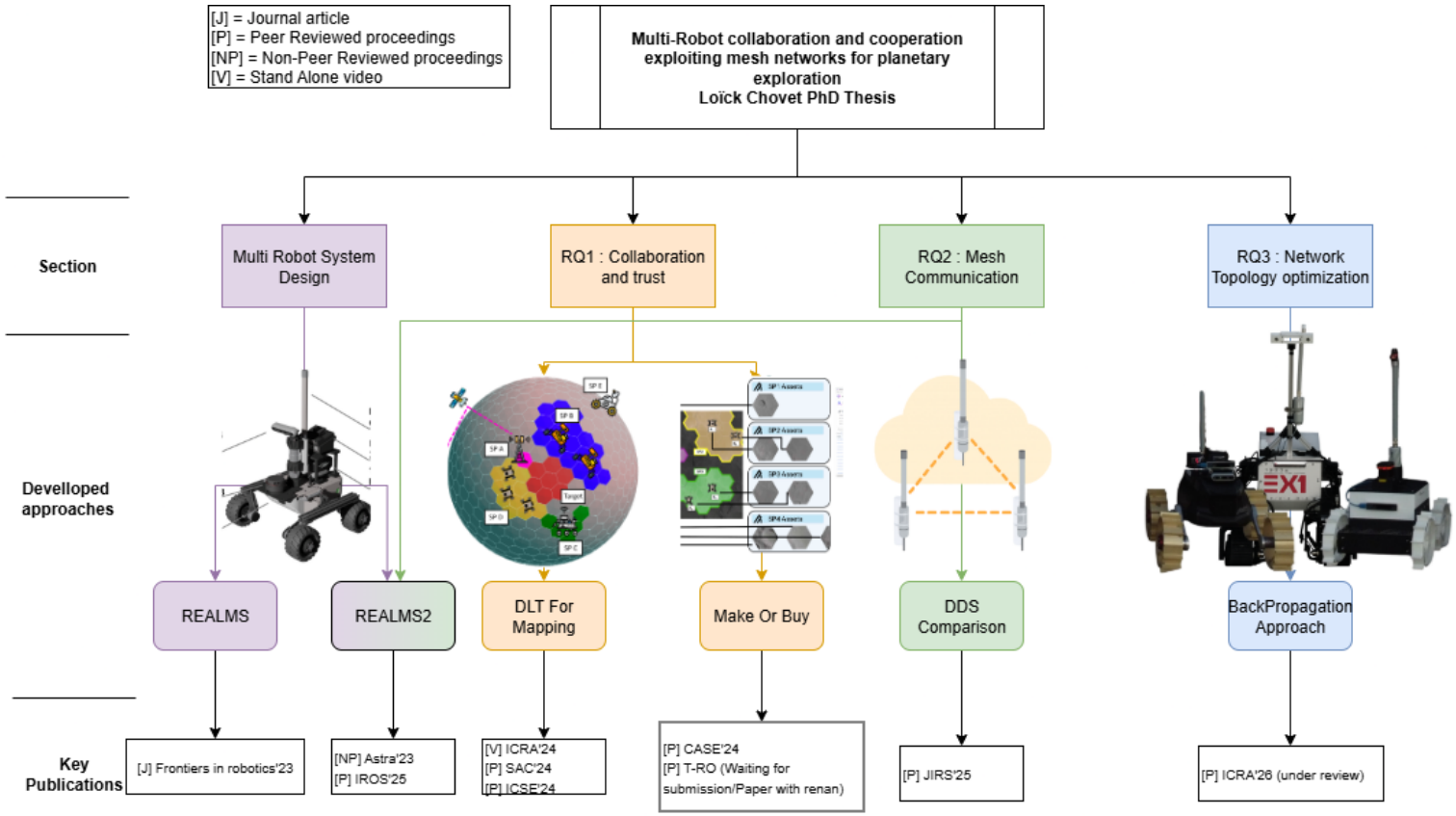


Figure 1.5: Thesis Structure

- **An approach for a market-based, robot-agnostic, task-agnostic coopetitive system**. The *Make or Buy* approach enables robots of various competing owners to determine the optimal way to perform a task, be it by creating it in-house or outsourcing it. This approach comes with a cost function designed for evaluating the price to perform any type of task, taking into acount the capabilities and characteristics of each robot.
- **A detailed comparison of middleware approaches for optimal ROS 2 communication over mesh networks.** This work benchmarks the main ROS 2 middleware options (FastDDS, CycloneDDS, Zenoh) in a realistic lunar exploration setup using dynamic mesh topologies. By testing reachability, delay, overhead, and resource usage across multiple message sizes, it identifies the trade-offs of each middleware. Zenoh stands out for low CPU usage and high network resilience, making it a strong candidate for space-grade multi-robot deployments.
- **An approach for connectivity maintenance over mesh networks for space exploration.** The BackPropagation approach is proposed as a solution to have a robot go further away from the communication reach, while some relay robots adapt their position in order to ensure permanent connection. The approach take in count preliminary knowledge of the environment through the use of pre-made DEM, but also the local knowledge of each robot. It also takes into account the characteristics of each robot, such as their speeds, to adapt the behaviours, mitigating potential issues.

1.5 Thesis outline

This thesis is organized as follow:

- Chapter 2 offers a literature review of the state of MRS along with their most important needs. It also develops the state of space robotics and the future applications expected. It then examines the existing collaboration mechanisms and the concept of coopetitive environments. In order to dig into decentralised MRS competition, it details the existing market-based approaches. Finally it investigates the concept of networked robot, with

details about the mesh technology and connectivity maintenance approaches.

- Chapter 3 introduces REALMS 1 and 2, our MRS developed for the ESA-ESRIC Space Resources Challenge. We present this decentralised robotic system and its evolution to a system that served as a basis for the following sections. This chapter includes the results of the first and second trial of the ESA-ESRIC Space Resources Challenge.

- Chapter 4 details the reasoning that led to the adoption of the *make-or-buy* framework as a response to our first research question. We introduce a novel architecture enabling a coopetitive MRS to execute agnostic tasks in exchange for cryptocurrency-based compensation autonomously. This architecture is evaluated through simulated experiments and benchmarked against both upper and lower performance bounds. In addition to the simulation results, the proposed mechanisms are implemented on a real-world robotic system (REALMS2) to assess their feasibility and validate their effectiveness in practice.

- Chapter 5 explores the use of mesh networks for MRS in the context of space exploration. It evaluates the performance of ROS 2-based communication systems over such networks and investigates how to fine-tune their behaviour for optimal results. In this context, we examine the ROS 2 middleware stack and investigate *Zenoh*, a novel middleware designed for efficiency in dynamic and resource-constrained environments. The evaluation is based on two real-world experiments under different networking conditions. The first involves static robots operating in both Line-Of-Sight (LOS) and Non Line-Of-Sight (NLOS) scenarios. The second focuses exclusively on a NLOS scenario, where a moving robot causes changes in the network topology. Results show that *Zenoh* consistently outperforms traditional ROS 2 middleware implementations, making it a compelling solution for meshed MRS, particularly in the demanding context of space exploration.

- Chapter 6 presents our solution to the connectivity maintenance challenge: the *Back-Propagation* approach. This decentralised strategy draws from a range of existing connectivity maintenance techniques, aiming to ensure reliable communication with

minimal energy consumption. Using prior knowledge such as DEM and robot-generated maps, it is able to maintain LOS communication links. The proposed method is validated through a series of simulated experiments across multiple scenarios, and benchmarked against existing approaches. In addition to simulation, the approach is demonstrated in a real heterogeneous MRS in collaboration with the Space Robotics Laboratory (SRL) at Tohoku University in Japan.

- Chapter 7 closes this thesis with a summary of the research along with a discussion of the future research to extend this work.

1.6 Publications

This thesis is based on the following peer-reviewed publications, classified by authorship contribution and by the Qualis PPGCC 2023–2024 ranking of the publication venue. Within each category, works are ordered from highest to lowest Qualis tier.

As main author

- van der Meer, D.†, Chovet, L.†, Garcia, G.†, Bera, A., Olivares-Mendez, M. A. **"REALMS2 - Resilient Exploration And Lunar Mapping System 2 – A Comprehensive Approach"**, To appear in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2025. (In Press)
  Qualis ref.: A1

- Chovet, L., Kern, J. M., Bera, A., Santra, S., Olivares-Mendez, M. A., Yoshida, K. **"Robust Connectivity Maintenance for Heterogeneous Multi-Robot Systems for Planetary Exploration"**, Under publishing process for the IEEE International Conference on Robotics and Automation (ICRA), 2025.
  Qualis ref.: A1

- Lima Baima, R.†, Chovet, L.†, Olivares-Mendez, M. A., Fridgen, G.**"Towards Space Machine Economy: Assessing Non-proprietary-market-based Multi-Robot Coordi-**

nation" Aimed to be submitted IEEE Transactions on Robotics (T-RO).
Qualis ref.: A1

- Chovet, L.†, Lima Baima, R.†, Hartwich, E., Bera, A., Sedlmeir, J., Fridgen, G., Olivares-Mendez, M. A. **"Trustful Coopetitive Infrastructures for the New Space Exploration Era"**, Proceedings of the 2024 ACM Symposium on Applied Computing (SAC24), 2024. https://hdl.handle.net/10993/60481
Qualis ref.: A2

- Chovet, L.†, Garcia, G.†, Bera, A., Richard, A., Yoshida, K., Olivares-Mendez, M. A. **"Performance Comparison of ROS2 Middlewares for Multi-Robot Mesh Networks in Planetary Exploration"**, Journal of Intelligent and Robotic Systems, 2024. (Accepted/In Press). https://hdl.handle.net/10993/64125
Qualis ref.: A2

- Chovet, L., Lima Baima, R., Bera, A., Olivares-Mendez, M. A., Fridgen, G. **"Coopetitive Lunar Mapping: A distributed non-proprietary Multi-Robot Coordination using Blockchain-based Cost Optimisation"**, IEEE Conference on Automation Science and Engineering (CASE), 2024. https://hdl.handle.net/10993/62339
Qualis ref.: A3

As co-author

- Lima Baima, R., Chovet, L., Sedlmeir, J., Olivares-Mendez, M. A., Fridgen, G. **"Designing Trustful Cooperation Ecosystems is Key to the New Space Exploration Era"**, Proceedings of the 2024 IEE/ACM International Conference on Software Engineering (ICSE), New Ideas and Emerging Results (In Press). https://doi.org/10.48550/arXiv.2402.06036
Qualis ref.: A1

- van der Meer, D., Chovet, L., Bera, A., Richard, A., Sanchez-Cuevas, P. J., Sánchez-Ibáñez, J. R. Olivares-Mendez, M. A. **"REALMS: Resilient Exploration and Lunar**

Mapping System", Frontiers in Robotics and AI, 2023 https://hdl.handle.net/10993/57382

Qualis ref.: A3

Along with the following peer reviewed abstract :

- van der Meer, D.†, Chovet, L.†, Garcia, G.†, Bera, A., Olivares-Mendez, M. A. **"REALMS2 – Resilient Exploration And Lunar Mapping System 2"**, Astra, 2023. https://hdl.handle.net/10993/57411

† These authors contributed equally to this work.

Chapter 2

Literature Review

"Any sufficiently advanced technology is indistinguishable from magic."

– Arthur C. Clarke

2.1 Introduction

This chapter surveys the state of the art in designing, coordinating, and deploying MRS for space and other extreme environments. We begin with foundational architectures, from centralised and hierarchical to decentralised and hybrid, arguing that delay, intermittency, and risk push practice toward decentralised autonomy for resilience and scale. We then move from classical space robotics to the middleware that makes these choices executable in the field, with a focus on ROS 2 and its DDS-backed QOS, security, and composability. Building on this stack, we examine coordination in coopetitive, multi-stakeholder settings and surface the resulting communication and incentive gaps. Because effective coopetition depends on robust links, we next detail networked-robot topologies and mesh routing under motion, and close with connectivity maintenance—methods that keep teams connected so planning, control, and markets remain viable in realistic lunar analogs and future interplanetary missions.

2.2 Multi-Robot Systems (MRS)

MRS have been extensively studied in various fields such as agriculture [17], search and rescue [18], and increasingly for space exploration. As explained in [19] and visible in Fig. 2.1, four primary architectures are distinguished:

- **Centralised**: A single point of coordination gathers all data and plans for the entire fleet. It allows simpler robot designs and highly optimised solutions but introduces a single point of failure and limited fault tolerance [20].
- **Hierarchical**: A supervisory tree where higher-level units oversee teams of robots. This adds fault tolerance by decentralising authority but still relies on top-level stability [21].
- **Decentralised**: Robots act independently but share information to reach a common goal. This increases fault tolerance but often leads to suboptimal or delayed coordination. The ALLIANCE architecture [22] is a key example.
- **Hybrid**: A combination of centralised and decentralised mechanisms, allowing robustness and scalability. Examples include architectures such as [23].

Decentralised and hybrid approaches have gained traction for their fault tolerance and flexibility. This is particularly useful in complex environments or scenarios involving multiple stakeholders. The notion of “coopetition” [24]—a blend of cooperation and competition—arises when robots owned by different companies need to collaborate while maintaining autonomy. In such cases, only decentralised or hybrid architectures offer the necessary independence.

The authors in [25] provide a comprehensive survey of cooperative, heterogeneous, decentralised robotics. They highlight the following challenges:

- **Control architecture**: Coordinating global behaviour from local data (Reactive, Deliberative, and Hybrid control models).
- **Consensus**: Reaching decisions without central leadership.
- **Containment**: Reducing interconnectivity to prevent cascading failures.

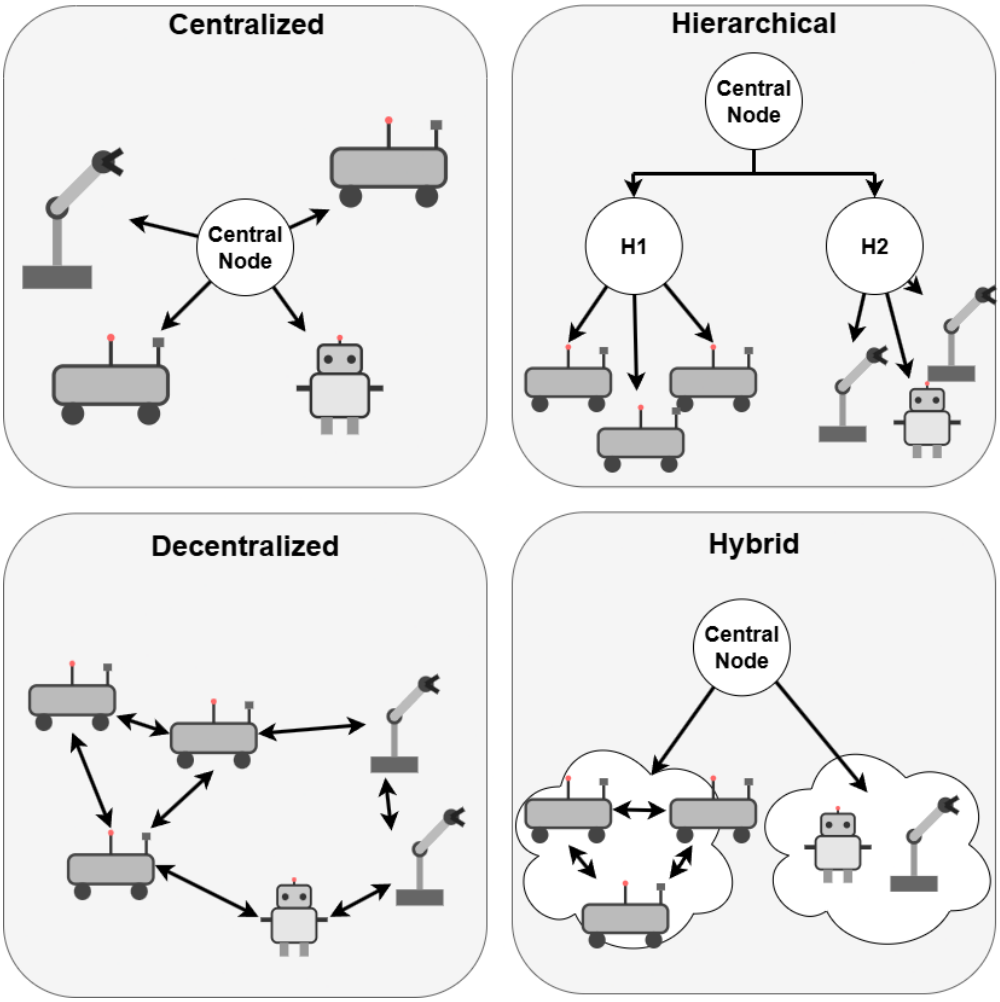


Figure 2.1: Visual representation of the four main MRS architectures.

- **Formation**: Coordinated group movement.
- **Task Allocation**: Deciding the most suited robot to perform a task.
- **Intelligence**: Sharing training data and learning models.
- **Optimisation**: Countering the inherent inefficiencies of decentralised systems.
- **Communication**: Intrinsic (via sensors) vs. explicit (dedicated channels).

They particularly emphasise the need for improved communication solutions in decentralised MRS, identifying four key challenges:

1. Efficiency
2. Environment tolerance
3. Consensus protocols
4. Speed and energy optimisation

One of the earliest decentralised MRS frameworks is ACTRESS [26], based on the Universal Modular ACTOR formalism [27]. In this architecture, robots called “robotors” operate as independent agents that communicate through peer-to-peer message passing. Each agent is modelled as an actor with perception, planning, and execution modules, allowing local autonomy while still contributing to a collective objective. This early work already captured the essence of modern decentralised robotics: modularity, self-contained reasoning, and distributed cooperation.

Building on this foundation, the ALLIANCE framework [22] introduced adaptive task reallocation and explicit fault tolerance. In ALLIANCE, each robot continually evaluates its own motivation to perform available tasks based on internal state and perceived team performance. When a teammate fails or communication is lost, other robots autonomously reassign tasks to preserve mission objectives. This concept of *behaviour-based cooperation* inspired many later hybrid architectures where deliberative and reactive layers coexist to ensure robustness.

Through the 2000s, decentralised approaches matured beyond predefined behaviour sets. Probabilistic reasoning, distributed consensus, and market-based coordination began to dominate the literature, allowing MRS to make complex joint decisions under partial information [28]. At the same time, the increasing heterogeneity of robots (rovers, flyers, crawlers, and orbiters) highlighted the need for modular communication interfaces capable of spanning hardware and mission boundaries.

The Service Oriented Approach (SOA) paradigm emerged as a response to this growing complexity, promoting loosely coupled, discoverable services instead of monolithic control [29]. In this view, robots expose capabilities such as navigation, sensing, or manipulation as

network-accessible services—enabling concepts like *Robot-as-a-Service* and *Sensing-as-a-Service*. Such architectures are particularly promising for remote or harsh environments like the Moon or Mars, where robotic systems must dynamically share resources and data without central supervision. Service-oriented interaction also naturally aligns with coopetitive missions, where heterogeneous agents belonging to different stakeholders may contribute specific services to a shared exploration task.

A contemporary illustration of these principles can be seen in the DARPA Subterranean (SubT) Challenge [30]. Competing teams deployed heterogeneous fleets of ground and aerial robots to explore complex, GPS-denied underground environments autonomously. Success relied on decentralised mapping, multi-modal communication relays, and adaptive task allocation—mirroring the challenges faced in planetary exploration. The winning CERBERUS team, for instance, implemented a layered architecture where local autonomy on each robot was combined with mesh-based coordination and opportunistic data sharing. When connectivity was lost, robots continued exploring independently and re-synchronised upon reconnection. This demonstrated, at the field scale, the viability of decentralised and hybrid coordination under severe communication constraints.

In summary, decentralised MRS research has evolved from actor-based prototypes to robust, service-oriented systems capable of sustained autonomy and cooperation in dynamic environments. The lessons from terrestrial initiatives like the DARPA SubT Challenge directly inform upcoming planetary missions, where similar architectural principles—distributed decision-making, adaptive networking, and modular service composition—will underpin resilient robotic teams operating far from human supervision.

As autonomy, heterogeneity, and decentralisation have deepened, so too has the need for flexible communication frameworks capable of supporting them. This evolution has led to a new generation of middleware built to enable resilient, scalable, and interoperable coordination across distributed robotic systems.

Robot Operating System 2 (ROS 2) and Middleware for MRS

Middleware emerged as the connective tissue of multi-robot systems—bridging the gap between hardware diversity, distributed computation, and coordinated behaviour. Early frameworks such as Player/Stage [31] and CARMEN [32] adopted a client–server model in which a central process handled sensor abstraction and message routing. They simplified experimentation and code reuse but required stable connectivity and manual configuration, which limited scalability beyond laboratory networks.

Subsequent platforms—YARP [33], OpenRTM [34], and Open-RMF [35]—pushed toward more modular and distributed operation. YARP introduced a lightweight publish/subscribe layer and dynamic port connections across processes, favouring flexibility over strict timing guarantees. OpenRTM formalised component lifecycles and interface descriptions, offering real-time support and standardised data types at the cost of greater complexity and tighter coupling to specific execution environments. More recently, Open-RMF extended this modular approach to the orchestration of entire robot fleets, introducing high-level coordination for scheduling, traffic management, and task allocation across heterogeneous robots and facilities. It builds on existing middleware layers but remains dependent on central brokers and stable communication infrastructure.

In parallel, lightweight network protocols such as MQTT [36] popularised asynchronous publish/subscribe messaging with low bandwidth overhead, particularly suited for IoT and embedded devices. MQTT's simplicity and broker-based delivery, however, lacked the deterministic timing, discovery, and security controls required for safety-critical robotics.

Across these systems, several common traits stand out: they provide abstraction from hardware drivers, modular task decomposition, and message-based communication. Yet they differ in how they handle discovery, timing, and fault tolerance. Most rely on central servers or brokers; few support real-time quality-of-service control or peer-to-peer discovery; and almost none offer built-in security or interoperability across middleware implementations.

As robots became increasingly heterogeneous, mobile, and autonomous, these limitations became critical. Large, decentralised teams required middleware capable of dynamic

discovery without a master node, tunable communication reliability, and real-time determinism across heterogeneous networks. Addressing these challenges demanded a fundamental redesign of the robotics communication stack—culminating in modern, DDS-based frameworks that underpin ROS 2.

The ROS 2 has become the middleware of reference for modern robotic applications, particularly in distributed and muMRS. While the first version of ROS served as the foundational open-source framework for robotics over the past decade, it exhibited fundamental limitations in scalability, robustness, and real-time communication, making it inadequate for mission-critical deployments such as planetary exploration or heterogeneous MRS. These limitations stemmed largely from its reliance on a centralised architecture, with a single master node responsible for coordinating node discovery and message exchange—a design that created single points of failure and constrained the autonomy of agents in dynamic environments.

Key improvements of ROS 2 over ROS include:

- **Decentralised Discovery and Communication**: Enabled by DDS, ROS 2 eliminates the need for a master node. Nodes can independently discover each other, reducing the risk of complete system failure due to a single point of failure.
- **QoS-Driven Messaging**: ROS 2 introduces fine-grained QOS policies for each topic, including parameters like reliability (reliable vs. best-effort), durability, deadline enforcement, and liveliness. This allows MRS to adapt communication behaviours to environmental constraints (e.g., lossy lunar networks) or application-specific requirements (e.g., sensor data vs. control loops).
- **Real-Time Capabilities**: Thanks to DDS and improvements in the ROS 2 executor and `rclcpp` layer, it is possible to write deterministic, low-latency code for time-sensitive tasks such as motion control or inter-robot coordination.
- **Security and Interoperability**: ROS 2 supports Secure DDS extensions and is compatible with multiple DDS vendors (e.g., Cyclone DDS, Fast DDS), enhancing both flexibility and potential deployment on mixed fleets of robots.

- **Modular and Containerised Deployment**: With native support for component composition, lifecycle management, and containerised nodes, ROS 2 eases integration into modern DevOps pipelines and embedded platforms.

These architectural improvements are not only theoretical; they translate into measurable performance gains. As shown in Fig. 2.2, adapted from [37], ROS 2 exhibits higher throughput and lower jitter than ROS across different message sizes, while introducing slight overhead due to DDS serialisation. The graph compares latency, reliability, and message drop behaviour under network stress, highlighting the improved resilience of ROS 2 for distributed robotic architectures.

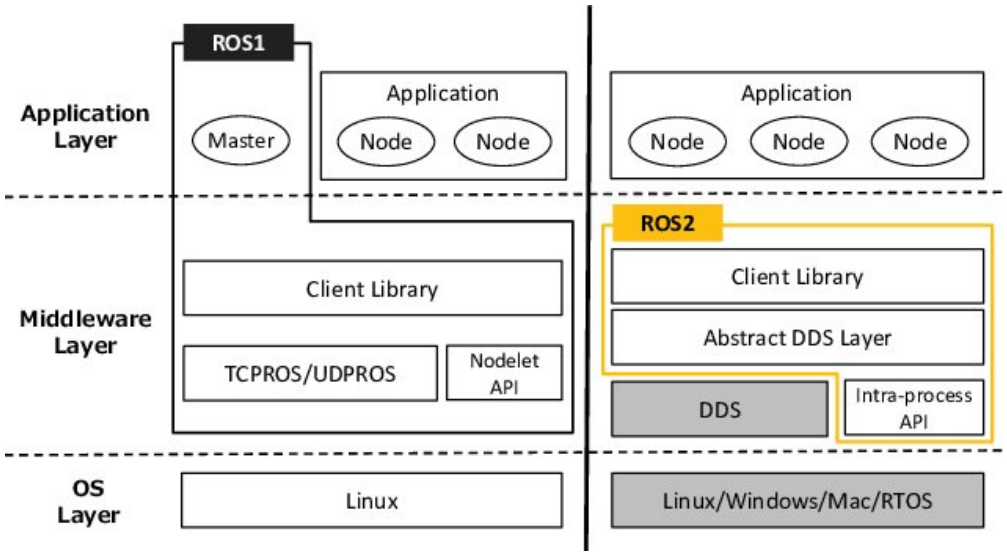


Figure 2.2: ROS vs. ROS 2 communication architecture and performance comparison, adapted from [37].

Nevertheless, ROS 2 is not without challenges. The use of DDS introduces non-negligible configuration complexity and computational overhead, particularly on resource-constrained platforms such as microcontrollers or embedded systems. Furthermore, most MRS research leveraging ROS 2 still assumes ideal communication conditions (e.g., no significant delays, stable links), which are unrealistic in extreme environments like the lunar surface. This thesis aims to bridge this gap by explicitly addressing the impact of limited bandwidth, variable topology, and connectivity loss on ROS 2-based MRS deployments.

In particular, we explore the integration of ROS 2 with mesh networking protocols and blockchain-based task coordination, leveraging its decentralised and modular nature to support robust communication and cooperation among heterogeneous robots. These contributions are evaluated in realistic lunar analog scenarios, where connectivity, data integrity, and autonomous decision-making are critical.

2.3 Space Robotics and Future Applications

Traditional planetary missions used a single multipurpose robot equipped with several sensors and actuators. Examples include Lunokhod 2 [38], Yutu-2 [39], Sojourner [40], and Curiosity [41]. Missions such as Spirit and Opportunity [42] deployed twin rovers, but each operated independently. Similarly, Perseverance and Ingenuity [43] did not exhibit cooperative behaviours while still operating a MRS, composed of a rover and a small helicopter.

These missions relied heavily on operator supervision due to limited mobility and high risks. For instance, Perseverance's top speed of 0.042 m/s [44] reflects the trade-off between coverage and safety. This limits area exploration and mission autonomy.

The landscape is changing. New commercial efforts [45, 46] aim to deploy lunar rovers with tight budgets and high expectations for efficiency. These missions will benefit from deploying MRS for resilience and scalability.

Unique constraints in space missions—like communication delays between Earth and celestial bodies—rule out centralised architectures. A master node would become a single point of failure and is impractical due to latency [47]. Consequently, decentralised or hybrid MRS designs are the most suitable.

The LUNARES project [48] is a key demonstration of a heterogeneous robotic team performing autonomous lunar sampling. Related projects like RIMRES [49] focus on even more complex robot assemblies. While [50] examined satellite constellations, the focus is now shifting to surface exploration.

The NASA Perseverance–Ingenuity pair marked a first step toward multi-agent systems on Mars. However, their actions remained uncoordinated. A more advanced mission is

NASA's CADRE[1], a technology demonstrator investigating how a homogeneous fleet of autonomous rovers can collaborate to share tasks, data, and decision-making responsibilities. CADRE showcases the potential of scalable, decentralised exploration tailored to space communication constraints and marks a milestone in collaborative autonomy for future lunar or Martian missions.

As this field evolves, future MRS space missions will require robust network architectures to coordinate multi-agent behaviour under harsh conditions. The following chapters will explore such architectures and the implications for lunar exploration using ROS 2-based MRS communication systems.

2.4 Collaboration mechanisms and Coopetitive environments

Coordinating multiple robots in real-world scenarios has long required scalable and resilient collaboration mechanisms. Traditional centralised coordination strategies tend to simplify robot design by offloading the decision-making to a main node. However, this comes at the cost of reduced fault tolerance and limited adaptability in dynamic or multi-stakeholder environments.

To overcome these limitations, decentralised and hybrid architectures have been proposed, particularly in contexts where robots operate across organisational or national boundaries. In these scenarios, collaboration must be achieved despite independent ownership and differing incentives—a situation described as *coopetition*, a blend of cooperation and competition [24]. This paradigm is especially relevant in space exploration and ISRU, where multiple actors may jointly explore, map, or exploit the same terrain while preserving autonomy.

MRS researchers have increasingly turned to market-based approaches for resource allocation, task distribution, and coordination in such coopetitive environments [28, 51]. These methods typically use auction mechanisms, where tasks or resources are bid upon by agents, allowing for scalable and emergent coordination. Auction-based systems can be more efficient and fault-tolerant than centralised planners, and have been explored for space

[1] `https://www.jpl.nasa.gov/missions/cadre/`

exploration scenarios like decentralised mapping [52].

Centralised approaches often fall short in accounting for the complexities of cross-organisational interactions and economic independence [53]. Distributed and decentralised market-based architectures, by contrast, offer greater autonomy, adaptability, and robustness—features critical for future lunar and planetary missions. Moreover, as planetary exploration missions become more industrial and commercially driven, it becomes vital to support autonomy not only at the technical level but also in the economic interaction between agents.

Emerging research proposes integrating contractual models into auction systems to further improve efficiency in space industry contexts [54, 28]. When combined with secure digital infrastructure—particularly decentralised ones—these systems enhance agent autonomy while reducing the risk of censorship, manipulation, or single points of failure [55, 56].

Ultimately, these developments highlight the need for new coordination paradigms capable of supporting heterogeneous, self-interested agents collaborating in harsh and dynamic environments. This thesis explores the design of such paradigms, focusing on the intersection of MRS, coopetition, and digital markets enabled by distributed ledger technologies.

2.5 Distributed Ledger Technology (DLT) and Market-based approaches

DLT provides decentralised transaction recording and validation across a network of nodes, offering similar capabilities to centralised platforms but without requiring a central operator [57]. As a subset of DLT, blockchain technology is characterised by its replicated, append-only, and hash-linked data structures [58], and supports various levels of openness and participation—from public and permissionless to private and permissioned networks [59].

The first major application of blockchain was Bitcoin [60], which popularised decentralised, trustless digital interactions. Today, blockchain technologies are used not only for currency, but also to encode business logic and agreements through smart contracts. These programs are stored and executed across the blockchain network, allowing tamper-resistant automation

of tasks such as payments, service execution, and compliance checking [61, 62].

Smart contracts enable novel forms of coordination in MRS by encoding market rules directly into a digital infrastructure shared among agents. Standards such as ERC20 and ERC721 facilitate both fungible and NFT, allowing robots to exchange not just currency, but also digital assets such as maps, task assignments, or performance guarantees [63, 64].

Decentralised MRS architectures can leverage blockchain to implement trustless, auditable, and automated coordination mechanisms [65, 66, 67]. Projects such as AIRA [68] introduced the idea of "Robonomics," where heterogeneous robots and humans interact via smart contracts, allowing service agreements and payments to be managed on-chain.

In the space domain, applications of blockchain have already been suggested or demonstrated in use cases such as orbital asset tracking, mining rights, decentralised identity, and even decentralised satellite constellations like SpaceChain [69, 70]. These platforms envision smart contracts managing everything from resource allocation to autonomous mission governance.

Despite these advancements, most applications still assume centralised ownership or homogeneous deployments. Very few address the unique challenges of decentralised, coopetitive, and dynamic MRS missions, especially in harsh planetary conditions. The integration of DLT into such scenarios presents new challenges: bandwidth limitations, energy constraints, real-time latency, consensus reliability, and the lack of flexible and expressive task representation formats [71, 72, 73].

There remains a significant research gap in designing DLT-based frameworks for autonomous coordination in truly decentralised and coopetitive MRS settings, particularly in ISRU contexts [74]. While many technical features of these platforms could theoretically be replicated by centralised infrastructure [75], doing so would reintroduce the very risks that DLT aims to eliminate—censorship, monopolisation, and single points of failure.

Our work addresses this gap by investigating how DLT can support automated market-based coordination in multi-stakeholder space missions. By leveraging smart contracts for secure task exchange, robot service provisioning, and reputation-based decision-making, we envision a decentralised economic system where autonomous agents cooperate, compete,

and evolve in a transparent, auditable, and fault-tolerant ecosystem—ultimately laying the foundation for a truly *coopetitive* MRS economy in space. This sets the stage for our **make-or-buy** approach as detailed in 4.3.

2.6 Networked robots

Mesh networks represent a network topology in which every node can act as a relay, further propagating the network. They are quite common in large-scale infrastructures and are the default topology constructed by separate entities connecting to each other. In the scenario of different robots owned by companies that share interests but not property over the system, the default architecture constructed by connecting all those companies will result in a mesh network.

It was introduced in the IEEE 802.11s norm [76] as a new topological approach to wireless networks. This approach to networking has often been shown to be the best solution for Mobile Ad-Hoc Networks (MANET) [77]. According to [78], the resulting network topology is the best trade-off between security, neutrality and the following points:

- **Robustness and Reliability**: Mesh networks are more robust and reliable due to their decentralised nature. If one node fails or becomes disconnected, the network can dynamically reroute traffic through alternative paths, ensuring reliable connectivity.
- **Scalability**: Mesh networks can easily scale up by adding additional nodes without causing significant disruptions or requiring extensive reconfiguration. Each new node increases the network's capacity and coverage area.
- **Redundancy**: With multiple paths available for data transmission, mesh networks provide built-in redundancy. This enhances fault tolerance and minimises the impact of node or link failures, improving performance and operational time.
- **Extended Coverage**: Particularly suited to large or challenging environments where traditional networks encounter limitations. Nodes communicate directly, facilitating broader coverage without a centralised infrastructure.

- **Flexibility and Adaptability**: Easily deployed and reconfigured to suit various scenarios, including dynamic environments where nodes may be mobile or the network's structure changes.

MRS can operate in various and challenging environments that require higher flexibility than traditional centred networks. The extended coverage and resilience of the network are crucial for MRS, as each robot is considered an antenna expanding the communication zone. Signal attenuation by terrain (e.g., rock formations) is mitigated by the complex topology adaptability of the mesh. Furthermore, this flexibility allows robots to be automatically reconnected to their peers during a running experiment if the network topology changes.

The OSI Model and its Relevance to MRS Networks

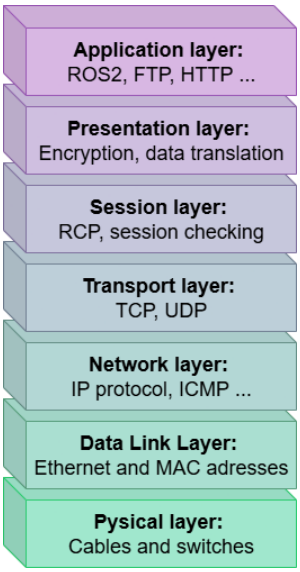


Figure 2.3: The OSI model applied to robotic communication systems. Each layer represents a conceptual boundary, from physical hardware to high-level application frameworks such as ROS 2. The colour gradient from purple to green visually encodes the abstraction level, from high (Application) to low (Physical).

Understanding mesh routing protocols requires a grounding in the OSI model, a conceptual framework that standardises the functions of a telecommunication or computing system into seven abstraction layers as visible in Fig.2.3:

1. **Physical Layer**: Manages the transmission of raw bitstreams over a physical medium (e.g., radio signals in wireless networks).
2. **Data Link Layer**: Ensures reliable transmission between directly connected nodes, managing MAC addresses and error detection.
3. **Network Layer**: Handles routing and forwarding of data packets across the mesh, using IP addresses and routing protocols.
4. **Transport Layer**: Guarantees end-to-end data integrity and flow control (e.g., TCP/UDP).
5. **Session Layer**: Manages sessions or connections between applications.
6. **Presentation Layer**: Translates data formats between application and network (e.g., encryption, compression).
7. **Application Layer**: Interfaces with user-level software, such as ROS 2 nodes communicating over DDS.

DDS operates primarily at the Application Layer of the OSI model, offering a high-level publish/subscribe communication abstraction. It also handles session and presentation responsibilities such as data encoding, QOS enforcement, and connection management, but always relies on underlying transport protocols for actual data transmission.

This layered model is crucial for understanding interoperability, routing decisions, and how different mesh protocols integrate within MRS systems. It also highlights a core advantage of layered design: each layer operates independently. As a result, the functioning of high-level frameworks such as ROS 2 remains unaffected by changes in the underlying layers—as long as the transport layer continues to deliver packets. This abstraction allows roboticists to experiment with or optimise lower-layer elements (such as mesh routing protocols) without modifying the application logic or middleware architecture.

Mesh routing protocols

In the following subsections, the most commonly used mesh routing protocols are introduced.

AODV

The Ad-hoc On-demand Distance Vector (AODV) protocol was introduced in 2003 as a reactive routing solution tailored for MANET [79]. AODV operates at the network layer (Layer 3) of the OSI model, ensuring dynamic and efficient route discovery between nodes without relying on pre-established infrastructure. Unlike proactive protocols that maintain full route tables at all times, AODV initiates route discovery processes only when communication is required. This on-demand approach significantly reduces routing overhead and conserves energy and bandwidth—two critical factors in mobile and resource-constrained environments such as planetary exploration scenarios.

When a node needs to communicate with another node for which it has no known route, it broadcasts a Route Request (RREQ) packet across the network. Intermediate nodes either reply with a Route Reply (RRER) if they have a valid route or propagate the RREQ further. Once the route is established, it is maintained only as long as it is needed. If a link break is detected, a Route Error (RERR) message is sent to inform affected nodes, triggering a new route discovery if necessary.

Due to its lightweight nature and minimal state requirements, AODV remains an attractive solution for decentralised MRS, particularly in exploration missions where the network topology is highly dynamic and nodes may frequently enter or leave the communication range. However, its simplicity comes at the cost of suboptimal link quality estimation and potential latency during initial route setup, which can impact real-time data sharing in mesh networked robotic teams. Operating at the network layer (Layer 3) of the OSI model, AODV is particularly suited for dynamic and mobile MRS where minimising routing overhead is essential, though it may lack awareness of link quality.

B.A.T.M.A.N Advanced

Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N) is a low-level decentralised routing protocol operating on the second OSI layer. This protocol is part of the Linux Kernel and only needs to be activated as a module to run on every device, even low-end ones [80].

Each Linux device with this module activated becomes a node equipped with a network switch.

B.A.T.M.A.N was derived from the Optimized Link State Routing (OLSR), which, like AODV, is a proactive protocol. The main difference lies in how information is passed: OLSR and AODV use multipoint relays, whereas B.A.T.M.A.N routes data through available neighbors. As a Layer 2 protocol, B.A.T.M.A.N Advanced offers robust and decentralised routing with minimal configuration, making it attractive for resilient mesh deployments on heterogeneous or low-power robotic platforms.

HWMP and HWMP+

Hybrid Wireless Mesh Protocol (HWMP), described in the IEEE 802.11s standard [81], is the default routing protocol for mesh networks and uses the airtime metric to assess the best communication paths. The airtime cost C_a is calculated as:

$$C_a = \left[O + \frac{B_t}{r}\right] \cdot \frac{1}{1 - PER} \tag{2.1}$$

$$PER = \frac{FrameResent}{FrameSent} \tag{2.2}$$

Where O is channel access overhead, B_t is the standard frame size, and r is the transmission rate.

Hybrid Wireless Mesh Protocol + (HWMP+) was introduced as an enhancement, improving the Packet Error Rate (PER) estimation and accounting for traffic load on the link to improve routing decisions [82]. The authors highlight HWMP+ achieves 3%-6% more packet delivery and reduces routing overhead by 2.8%-10.4%. Integrated into the IEEE 802.11s standard, HWMP and its enhancement HWMP+ operate at the data link layer (Layer 2), providing better link quality estimation and packet delivery performance in wireless mesh environments where interference and load balancing are key concerns.

Comparative performance Compared to AODV and B.A.T.M.A.N, HWMP is tied to IEEE 802.11s standards. AODV and B.A.T.M.A.N are more flexible in terms of device compatibility. However, they do not estimate link quality, limiting their efficiency in variable wireless environments.

The authors of [82] emphasise that most mesh metrics are poorly adapted to dynamic wireless conditions, especially when affected by noise, interference, and energy constraints. HWMP+ was designed with these constraints in mind.

2.7 Connectivity maintenance

The concept of computing the optimal position for keeping the network connectivity has been explored over the years. Existing works, such as communication-aware motion planning [83], connectivity preservation [84], and connectivity maintenance [85], described various approaches to maintain the connectivity between the robots of a MRS. Many authors suggests using a fleet of Unmanned Air Vehicule (UAV) to maintain direct connection [86]. However, such an application seems unrealistic for space missions where rovers are limited in speed and missions are expected to last too long for the UAV batteries.

These existing approaches can be categorised into the following main families:

1. **Graph-theoretic approaches:** Model communications as a graph and preserve connectivity by optimising properties like the Fiedler value (second-smallest Laplacian eigenvalue) [87].

2. **Geometric or force-based approaches:** leverage virtual attractive/repulsive forces (e.g., spring or potential-field analogies) to keep robots within communication range [88]. These approaches are also used to connect multiple MRS together with relay rovers.

3. **Signal-quality-based approaches:** relies on real-time measurements of the Received Signal Strength Indicator (RSSI) to adapt robot behaviour and avoid disconnection due to signal degradation. The authors of [89] present the creation of a network map.

Such an approach is less applicable to space systems, where motion is expensive and dangerous.

4. **LOS based methods:** ensure connectivity by maintaining geometric visibility, especially relevant when communication is obstructed by terrain [90]. Though relevant, these approaches target obstacle-dense settings like urban environments.

5. **Environment-aware or space-specific approaches:** explicitly incorporate terrain features, planetary propagation models, or mission-specific constraints into the connectivity strategy [91, 92].

6. **Intermittent connectivity approaches:** relax the requirement of continuous links by ensuring periodic rendezvous through temporal constraints [93]. Despite their efficiency, these methods are unsuitable for space missions where contact loss is unacceptable.

Many recent methods combine elements from multiple categories. For instance, the work of [87] integrates **environment-aware** modelling with **line-of-sight constraints** and **graph-theoretic reasoning**, to enable connectivity-aware path planning for lunar rovers. The authors consider a homogeneous fleet of lunar robots tasked with maintaining contact with a lander during exploration. Robots are modelled as nodes in a communication graph, where links are enabled based on environment-dependent thresholds. Connectivity is assessed via the Fiedler value (λ), with $\lambda > 1$ enforced to avoid disconnection. While effective, maintaining a high Fiedler value requires additional robots to act as relays, increasing energy consumption and reducing the system's effective range.

It is important to notice that the space environment implies various design choices. The DARPA Subterranean Challenge invited teams to offer solutions to explore subterranean environments. The winning team dropped regular relay points and added a routine to check that if a robot lost contact with the users [30], it returned to the last known point. In open-field exploration, beacons offer a too-limited range and efficiency while representing an important load on the robots. We also consider a system that focuses on a decentralised or hybrid approach to networking since a centralised approach reduces the individual capabilities of each robot.

Another state-of-the-art approach is presented as the FBA in [88]. It relies on the Fruchterman approach [94], combining **Graph theory** and **Force Based** approaches.

The Fruchterman approach is designed for graph drawings with two simple rules:

- Vertices that are neighbours should be drawn near to each other.
- Vertices should not be drawn too close to each other.

This proposed approach assumes that every link between the nodes acts as a spring mechanism, applying a repulsive force if two nodes are too close and an attractive force if they are too far. Every node receives forces from its neighbours, the sum of these forces leading to a total force on each node, moving the nodes toward their optimal position.

The authors assume that every robot is a node added to the graph that needs to respect the previously mentioned rules. Finally, It is worth mentioning that most of the literature focuses on simulated work, with limited validation on heterogeneous MRS.

Given the limitations of the aforementioned approaches and the lack of a space-dedicated approach, we next describe our mission scenario and formulate the connectivity maintenance problem while focusing on resource-constrained scenarios, such as analogue and planetary environments.

THE MAIN CONTRIBUTIONS

Chapter 3

Designing a Multi-robot system

Related Publications

- van der Meer, D., Chovet, L., Bera, A., Richard, A., Sanchez-Cuevas, P. J., Sánchez-Ibáñez, J. R. Olivares-Mendez, M. A. **"REALMS: Resilient Exploration and Lunar Mapping System"**, *Frontiers in Robotics and AI*, 2023 `https://hdl.handle.net/10993/57382`
- van der Meer, D.[†], Chovet, L.[†], Garcia, G.[†], Bera, A., Olivares-Mendez, M. A. **"REALMS2 – Resilient Exploration And Lunar Mapping System 2"**, *Astra*, 2023. `https://hdl.handle.net/10993/57411`
- van der Meer, D.[†], Chovet, L.[†], Garcia, G.[†], Bera, A., Olivares-Mendez, M. A. **"REALMS2 - Resilient Exploration And Lunar Mapping System 2 – A Comprehensive Approach"**, *To appear in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2025. (In Press)*

[†] These authors contributed equally to this work.

"I do not fear computers. I fear the lack of them"

– Isaac Asimov

3.1 Introduction

As space exploration shifts from singular governmental missions to a more pluralistic and competitive landscape, there is a growing necessity for robotic systems that are not only autonomous and robust, but also interoperable across organisational boundaries. The

emergence of a new space economy demands a rethink of how robotic systems are deployed, coordinated, and trusted in extreme environments.

To address this shift, we developed a novel experimental MRS, deliberately built around a set of foundational principles: decentralisation and resilience. Implemented using ROS 2 and operating on a mesh communication network, this MRS was designed to simulate ISRU scenarios.

The system was initially developed as a solution for the ESA-ESRIC Space Resources Challenge, as described in Section 3.2. Qualification for the final round of the challenge, along with insights gained from the first field test, motivated the design of an upgraded version, presented in Section 3.3.

This system, along with the lessons learned throughout the ESA-ESRIC Space Resources Challenge, forms the foundation for the rest of this thesis and directly motivates the formulation of **RQ-1**, **RQ-2**, and **RQ-3**.

3.2 REALMS: Resilient Exploration and Lunar Mapping System

This section introduces REALMS, MRS developed as a solution for the ESA-ESRIC Space Resources Challenge. This work serves as a foundation for all of the future work in the thesis.

3.2.1 The first trial of the ESA-ESRIC space resource challenge

The first trial of the ESA-ESRIC Space Resources Challenge challenged 13 teams to perform an ISRU mission in a lunar analogue facility. The objective consists of gathering visual data and generating a 3D map of an unknown environment. The environment contains a lunar illumination and lunar communication delays as expected during a space mission. In the challenge stage, the illumination was set up in a dark hall with black curtains and an array of bright spotlights to replicate sunlight with a low incidence angle, similar to the lunar south pole. The communication delay is achieved by using the ESA delayed communication system to simulate the delay between the Earth and the Moon at a software level. The round-trip delay consists of five seconds in total. Additionally, it is expected that the proposed system should

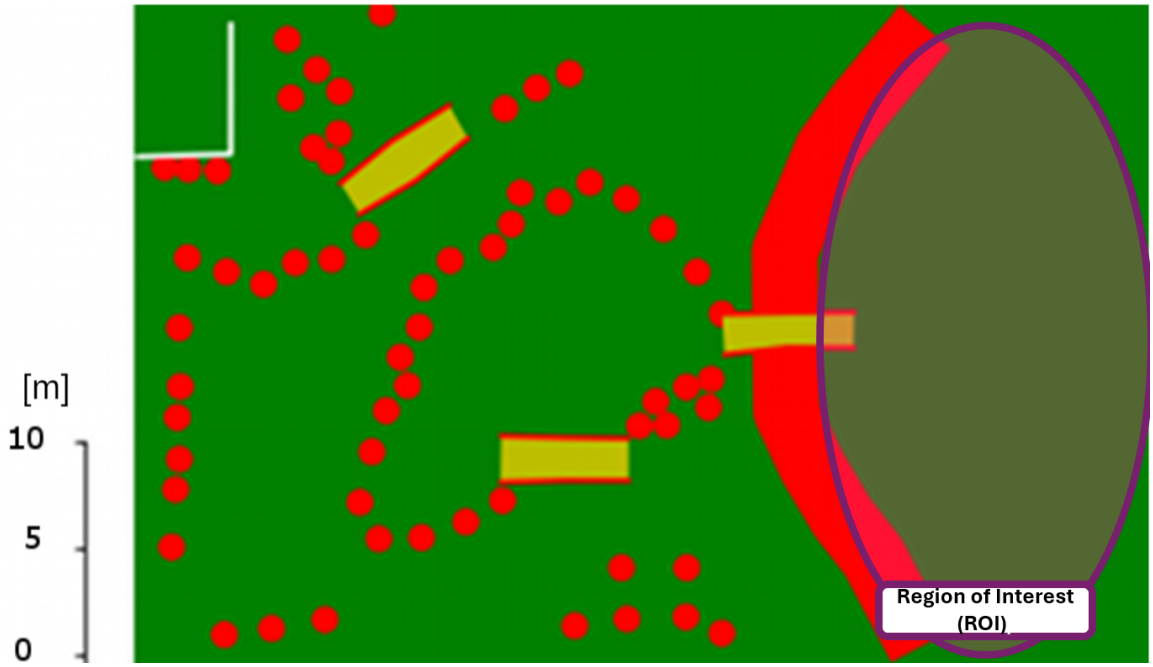


Figure 3.1: Provided map of the environment

be able to operate with occasional and eventual communication blackouts. A preliminary map is provided to the teams (see Fig.3.1) The environment is a flat concrete surface with several obstacles, such as rocks and ramps. The goal is to reach a region of interest (ROI), representing a large crater, filled with small rocks on the soil. It contains larger rocks that need to be analysed by the research teams. The ROI can be accessed through a ramp.

Then, taking into account the challenge description, the following requirements are identified:

1. The system must map as much as possible of the 2500 m² area in 2.5 h.
2. The system must be able to move and explore a lunar surface analogue zone and navigate through rocks and slopes.
3. The system must be impervious to a five-second delay, unpredictable blackouts and a limited bandwidth.
4. The system must be resilient to partial system failure, allowing it to finish the mission

even when parts of the system fail.

3.2.2 Proposed solution

The implemented system consists of two identical rovers controlled by two identical ground stations over a delayed network. This entire system can be extended to accommodate any number of rovers and ground stations, depending on the available bandwidth. This section explains the whole REALMS architecture composed of n rovers and ground stations, the Earth-Moon-Earth delay simulator and the lunar testing environment as shown in Fig.3.2. The control room with the ground stations to control the rovers is connected to the lunar testing environment through an Earth-Moon-Earth Delay Simulator. Inside the lunar testing environment, the rovers are connected through a wireless connection. The rovers can communicate with their respective ground station through the delay simulator, with a communication delay of 2.5 s in each direction of the data transmission.

This setup with the control room, the delay simulator, and the lunar testing environment is replicated in the LunaLab [95]. The LunaLab is the lunar analogue facility of the University of Luxembourg, a $8\,\mathrm{m}$ by $11\,\mathrm{m}$ ($88\,\mathrm{m}^2$ room containing 20 tons of basalt focusing on the optical fidelity with respect to lunar environments

The next sections elaborate on the different components in more detail. First, we describe the Earth-Moon-Earth delay simulator that adds communication delay in the network. Second, we explain the ground station setup. Third, we present the rovers, and finally, we present the multi-master architecture.

Earth-Moon-Earth Delay Simulator

Fig. 3.3 describes the developed network architecture of the lunar delay network [96] to test the performance of the proposed system. The system represented here connects the control room to a network in the lunar analogue facility by introducing a delay of 2.5 s in each direction of the connection. The delay computer has a 3.0 GHz Intel Core $i7$ generation 8 processor, and 8GB of RAM. The operating system that we use is FreeBSD 12.2. The delay computer has two separate network interfaces, $ue0$ and $ue1$, as described in Fig. 3.3. There are two

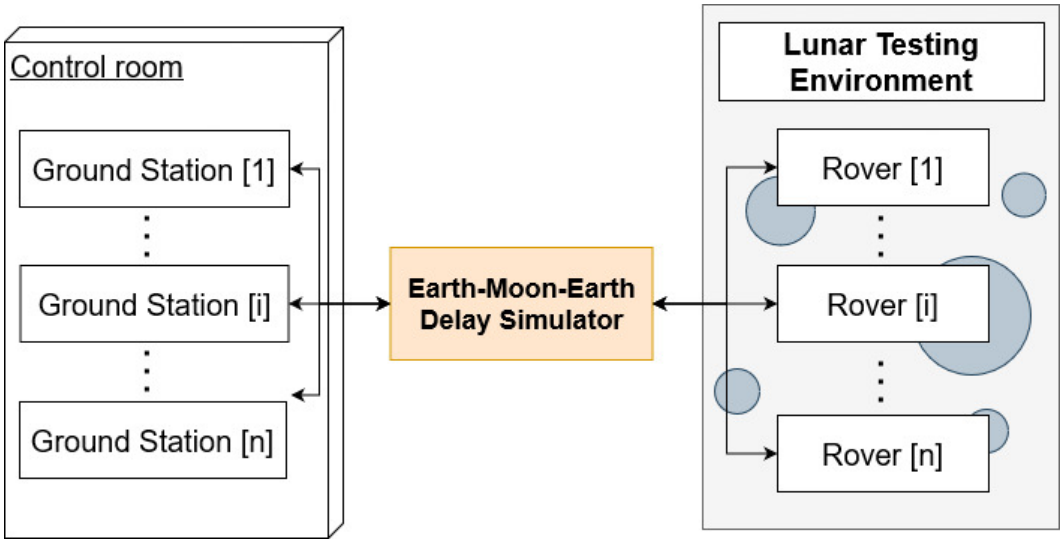


Figure 3.2: Overview of the REALMS architecture showing how multiple ground stations connect to multiple rovers through the Earth-Moon-Earth
Delay Simulator

routers, Delay Router and LunaLab Router, connected to $ue0$ and $ue1$, respectively. All the remote computers controlling the navigation and movement of the rovers are connected via Ethernet cable to the Delay Router. Also, the REALMS rovers are connected to LunaLab Router via 2.4 GHz Wi-Fi signal. In order to emulate an end-to-end delay between the remote computer and the rovers, there is a bridge, called $bridge0$, between $ue0$ and $ue1$. Therefore, all the traffic passes through the bridge between the control room and the LunaLab. Finally, two rules are set for the outgoing traffic from each network interface that is connected to the bridge ($ue0$ and $ue1$) using the “ipfw” command to introduce the specific delay.

Ground stations

The rovers are controlled through computers that serve as a ground station. Fig. 3.4 shows the functions of the ground stations. They include visualising the rovers and their environment, the possibility of giving navigation goals to the rovers and the ability to teleoperate them. The communication between the ground stations and the rovers is established through FKIE multimaster nodes. Each ground station is running a ROS Master.

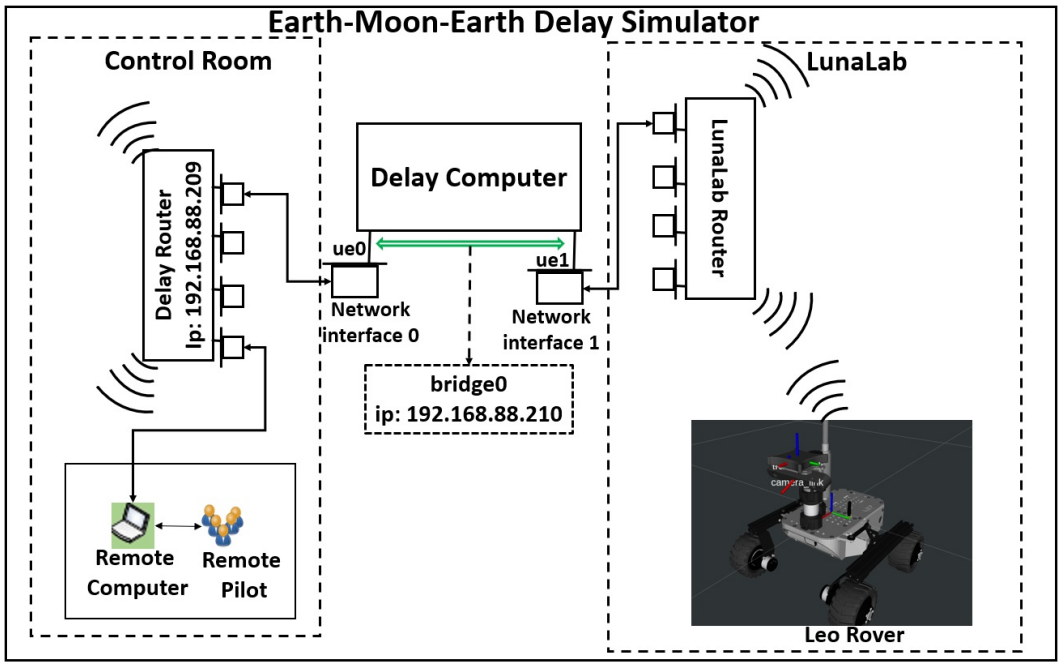


Figure 3.3: Delay Network Architecture connecting the rovers to the ground stations of REALMS by delaying all network traffic by a pre-defined amount of time

RViz is used as a user interface and allows sending a navigation goal to the rovers. The FKIE multimaster software enables RViz to control the rovers despite the presence of network delay. Each robot is unaware of the other robots in the network allowing for easy scaling of the network and reducing interference between the robots. Additionally, the ground station can switch to manual mode for teleoperation of the robot via input devices.

Rover

The rovers used for REALMS are two off-the-shelf robots modified according to the needs of lunar exploration and mapping in lab conditions. Each rover is a modified version of a Leo Rover [97]. Fig.3.5 shows a Leo Rover used for REALMS with all the relevant components, such as the cameras, the communication antenna, the Nvidia Jetson, the lights and the wheels. Each robot is running a Visual Simultaneous Localisation And Mapping (vSLAM) algorithm that uses the visual input of the RealSense camera and its Inertial Measurement

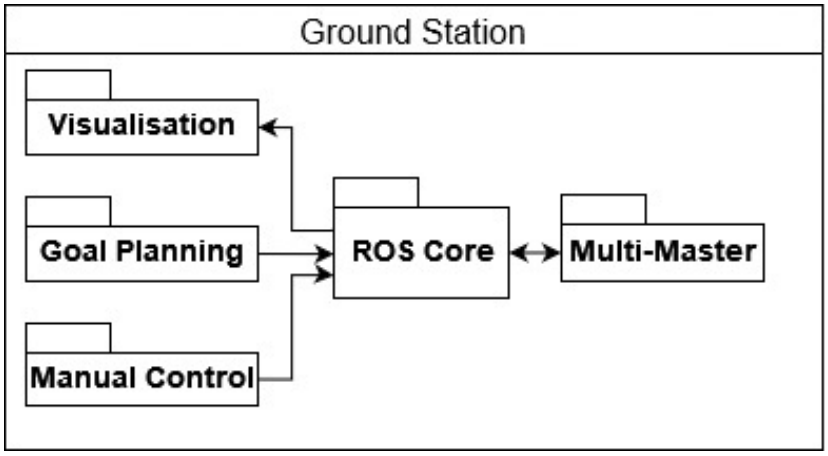


Figure 3.4: REALMS ground station architecture diagram showing commands sent to the robot and visualisation based on data received by the rover

Unit (IMU) data after filtering it. The 3D and 2D map generated by the vSLAM algorithm are used for autonomous navigation.

Multimaster

The multimaster component focuses on overcoming potential issues with the communication delay and loss, as well as increasing the resilience of the entire system, hence addressing the third and fourth requirements. It allows running one ROS Master on each system element and thus ensures that the topics are only shared between a ground station and its corresponding robot. The ROS Master is a central part of the ROS ecosystem as it handles topics, services and actions, registers which nodes are publishing and subscribing, holds the parameter server and directs the data traffic to the corresponding nodes. By conventional definition, there is only one single ROS Master in a given network of robots to handle all the ROS data traffic within the system. Multiple robots can share a single ROS Master, however this leads to a centralised architecture, more prone to failure, especially when the connection to the ROS Master gets interrupted.

We integrate the FKIE multimaster [98] in REALMS to prevent communication issues between the ground station and the robots by connecting multiple ROS Master instances

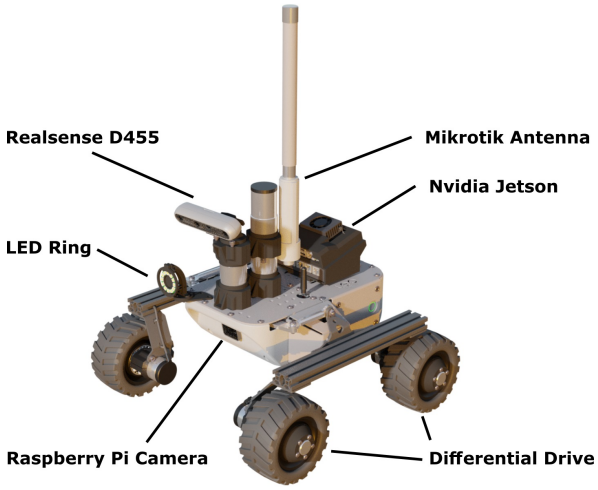


Figure 3.5: Overview of the REALMS Leo Rover hardware

and sharing topics between them. It comprises two main components, *discovery* and *sync*. *Discovery* can show all the ROS Master instances available on a network. *Sync* is used to get the topics and messages from the desired ROS Master.

The two aforementioned components are set up to allow sharing only the correct rover's topic with the desired ground station. This is done by using the option *sync_hosts* filled with the IP address of the robot and the ground station.

3.2.3 System Analysis

Each requirement in subsection 3.2.1 is analysed and the system designed to meet them accordingly. Table 3.1 shows how each component addresses each requirement. A component can serve as a key component (K) or supportive component (S). A key component is responsible to meet one of the requirements, while a supportive component contributes partially to meet a requirement in a non-essential way. In this project, the components and requirements with a grey background were handled by other team members, the content won't be detailed here but each component got verified.

Table 3.1: Components addressing the system requirements, with a white background are the ones developped in this work

Components	**Requirements** (K: Key Component, S: Supportive Component)			
	Mapping	***Movement***	***Delay***	***Resilience***
Lights	S	S		
Motors	S	K		
Camera	K	S		
IMU	S			
vSLAM	K			
Planner	K			
Multimaster			K	S
Visualisation	S	S		
Multi robot	S	S		K
Dual control mode	S	S	K	K

Environment constraints

Maximum slopes

In the permanently shadowed regions of the Moon, a robot needs to handle slopes of up to $22.1°$ [99]. We measure the maximum inclination angles the REALMS rovers can mount. They traverse a ramp as shown in Fig. 3.6 multiple times using three different surface materials while gradually increasing the inclination angle. In this way, we discover the values of the maximum inclination angle the rovers can climb according to these materials. The maximum angle is $30°$ for loose basalt, $22.5°$ for a solid wooden surface and $26.6°$ for an aluminium surface. The friction on basalt is higher than on aluminium, which causes the wheels to slip on aluminium.

Delay invariance

Standard software has a timeout function implemented. This function stops the program if no data is received in a certain amount of time. The timeout function prevents communication when there is a total communication delay of 5 s as is the case in Earth-Moon-Earth communication. The visualisation software RViz needs to connect to a ROS Master as otherwise, it returns an error after a timeout of 1 s. For terrestrial applications, it is common to run a

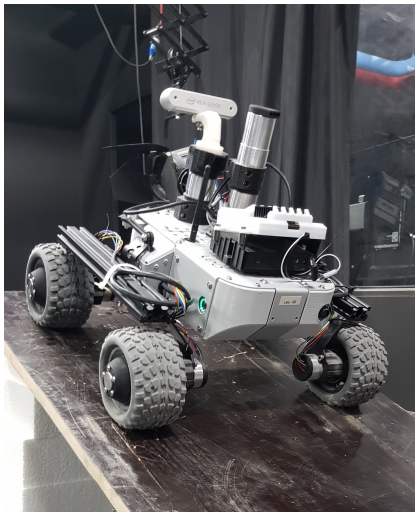


Figure 3.6: Experiment to determine the maximum slope inclination the rovers can traverse

single ROS Master in the robotic network where one robot contains the ROS Master, and the ground station is a slave connecting to the ROS Master of the robot. The communication delay does not allow this connection due to the timeout. REALMS overcomes this issue by running a ROS Master on each device involved. This way, RViz and similar software always receive inputs from their local ROS Master. The FKIE multimaster software is bridging the communication between the individual ROS Master instances, making the system delay invariant as it does not implement a timeout for the communication between the ROS Master.

Communication blackouts

The challenge contains periods with communication blackouts to represent scenarios where the communication antennas temporarily have no clear line of sight to transmit data. During a communication blackout, the rovers can move autonomously until they reach their goal and then wait for a new goal. While the communication is cut, the rovers keep waiting for new instructions, while the ground station sends the next commands to the rovers as soon as the connection is re-established. Due to the FKIE multimaster, communication with the ROS Master is ensured for the robots and the ground station. The timeout functions of the ROS nodes are not triggered since the communication to the local ROS Master is still intact. The

re-establishment of the communication between the rovers and the ground station is handled by the FKIE multimaster and its discovery function that allows connecting to an existing ROS Master.

Any packages lost during the communication loss are not retrieved. The rovers process the sensor data onboard and only send mapping data, odometry information and a low-resolution greyscale video stream back to the ground station. The only data sent from the ground station are either position goals for autonomous navigation, direct teleoperation commands or signals to trigger minor actions such as turning on or off the lights or partially restarting the system. In case of lost data packages, the commanded action must be repeated.

Resilience

The resilience of a system is its ability to recover after a partial failure. In the case of this challenge, it is important to see if all the previous requirements can be matched even with a faulty component. REALMS consists of a defined number of rover-ground station pairs. The bandwidth limits the maximum number of pairs. The ROS Master running on each machine make the system more robust as each robot and its corresponding ground station are not interdependent. If one of the two members is faulty, it can still be used to operate another member.

The REALMS used for the challenge is composed of two rover-ground station pairs, reducing the risk of failure by adding redundancy in the system architecture.Having more than one pair assures resilience and higher tolerance to potential blackouts. Additionally, the use of the multimaster setup makes the system resilient towards communication delay. The maps created by each robot are saved locally. Each map can be retrieved by the ground stations and merged on the ground stations, allowing to use an incomplete map to enhance the global map. At this point, the REALMS rovers are ready to face the lunar surface like environmental conditions expected in the challenge.

3.2.4 Experiments and Results

Testing REALMS in the LunaLab

To evaluate REALMS' multi-robot mapping capabilities, two rovers are deployed in separate locations within the LunaLab (Fig. 3.7).

Figure 3.7: LunaLab, University of Luxembourg. Equipped with an illumination system simulating lunar south pole lighting conditions.

Two scenarios are tested:

Scenario A (Fig. 3.8 (A)) demonstrates successful collaborative mapping. Rover 1 maps the top (light blue), and Rover 2 the bottom (pink) of the lab, with the overlapping region shown in purple . The mapping is completed in $6\,\mathrm{min}\,48\,\mathrm{s}$.

Scenario B simulates a failure of Rover 2 after $1\,\mathrm{min}\,30\,\mathrm{s}$, becoming unresponsive due to potential issues (e.g., low battery, wheel entrapment, odometry loss). The failure point is marked with a red circle in Fig. 3.8 (B). Rover 1 completes the mapping with reduced overlap. Total duration: $9\,\mathrm{min}\,2\,\mathrm{s}$.

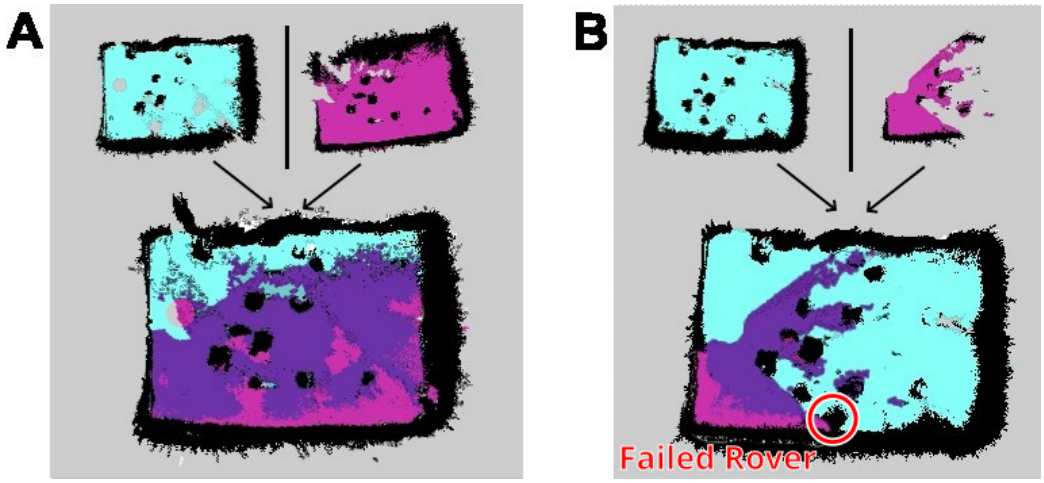


Figure 3.8: Mapping of the LunaLab done by two REALMS rovers in two different cases. **(A)** Two rovers successfully map a shared environment and merge their maps. **(B)** Two rovers mapping a shared environment with one rover failing in the process and the other taking over the area.

Realms at the first trial of the ESA-ESRIC Space Resources Challenge

The first trial of the ESA-ESRIC Space Resources Challenge consists of 6 hours of preparation and 2.5 h to realise the mission. The mission being an exploration task of a 34×47 m^2 area, including a ROI made of small rocks of $3 - 5$ cm diameter. The robots are placed in a starting area and must explore as much surface as possible, targeting mostly the ROI. The control room does not allow any point of view to the challenge area, the only insight being hand-drawn maps handed out by the organising team (Fig.3.9). Communication between the control room and REALMS is delayed as detailed in 3.2.2 but also subject to random blackouts planned by the organising team.

The mission was a success despite several communication blackouts. One robot managed to reach the ROI in the expected 2.5h time. The original plan was to have one robot directly heading to the ROI while the second one explores the area. However, the network's limited capacity did not allow the robots to operate simultaneously, given the amount of data each robot sent. After the first blackout, we lost contact with the second rover. As visible in Fig.3.9, the first rover took a direct path to the ROI, taking advantage of its small size to pass between

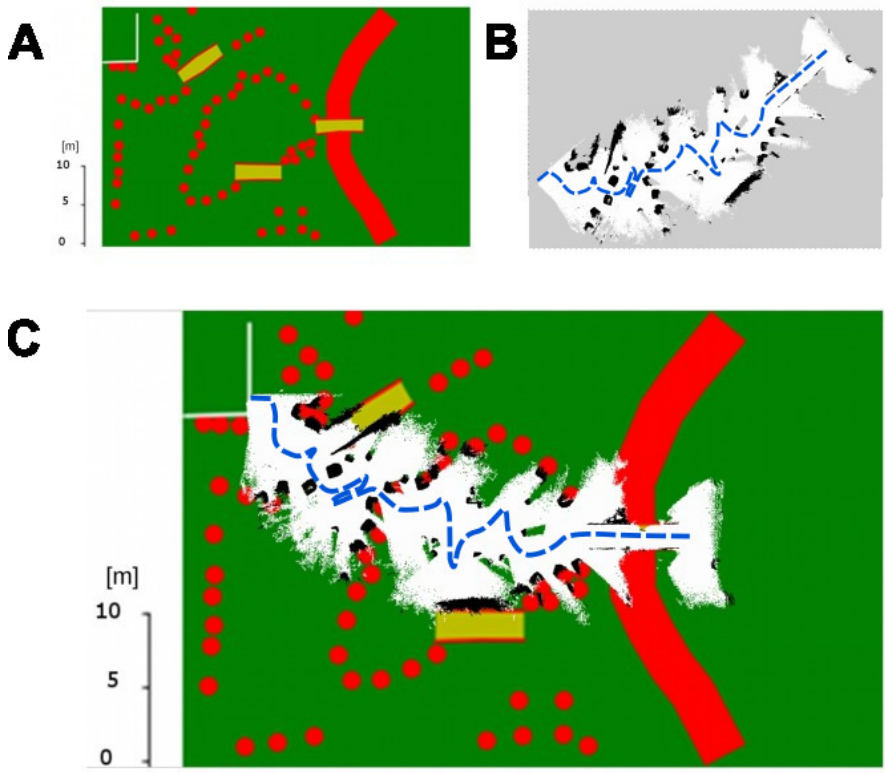


Figure 3.9: **(A)** Map provided by ESA at the beginning of the mission. **(B)** Map created by one rover during the mission. **(C)** Map of the lunar environment of the challenge overlaid with the map generated by the REALMS rover. The two maps are matching, showing the solution is accurate.

the rocks, avoiding a way longer path for larger robots. As a final result, REALMS managed to cover $310\,\mathrm{m}^2$, which corresponds of **19.4%** of the challenge area which is an interesting result.

3.2.5 Discussion on REALMS and challenges remaining to be adressed

REALMS presents a system to increase resilience and coverage for robotic mapping tasks. This is achieved by using multiple small rovers that can work in parallel to overcome challenges like partial system failures and lead the mission to success. The possibility to grow the fleet size with additional rovers allows to increase the mapping capability and system resilience.

The system shows its ability to perform during the Space Resources Challenge.

However, the first trial revealed several challenges that need to be addressed:

1. The communication architecture based on ROS 1 using the FKIE multimaster package does not provide the necessary stability to reliably connect to the robots.
2. The inter-robot communication is entirely depending on the provided access point during the challenge. This approach is less reliable and can increase network latency.
3. The resilience of the system is a major contribution to finishing the mission to this extent, given that one robot loses the connection to the ground station, the other rover can still operate.
4. The user interface easily scales on a system level, but not on a user experience level. Managing multiple robots on multiple operator computers is not feasible for large scale systems.
5. The bandwidth is limited, which causes communication losses when high data traffic is engaged, hindering the transfer of data to the ground station.

The qualification for the second trial of the SRC led to the creation of an improved version, assessing the revealed challenges: REALMS2.

3.3 REALMS2

3.3.1 The final trial of the ESA-ESRIC Space Resources Challenge

The five best teams from the first trial of the challenge advanced to a second and final field test [100]. This new phase introduced different parameters, placing a stronger emphasis on exploration within a lunar-analogue environment, as shown in Fig. 3.10. The primary objective of this trial was to identify various resources over an extended duration, within conditions that closely resemble those of a real lunar mission.

The conditions of this trial differed significantly from the first. As seen in Fig. 3.10, the terrain was composed of a simplistic lunar regolith simulant, offering a much higher fidelity to

Figure 3.10: Second field test arena for the ESA-ESRIC Space Resources Challenge

lunar conditions compared to the concrete floor used previously. Unlike the first trial, which focused primarily on reaching a single ROI, this second phase highlighted the importance of MRS by requiring exploration and mapping of several regions of interest, along with the identification of their associated resources.

Additionally, the duration of the challenge was extended to four hours, plus an additional hour dedicated to data processing—doubling the time of the previous trial. This introduced new challenges, such as ensuring the system's autonomy and energy sufficiency over a prolonged mission. Finally, the trial emphasised system resilience by simulating the failure of one randomly selected agent during the mission, requiring the system to adapt and continue its operations despite the loss.

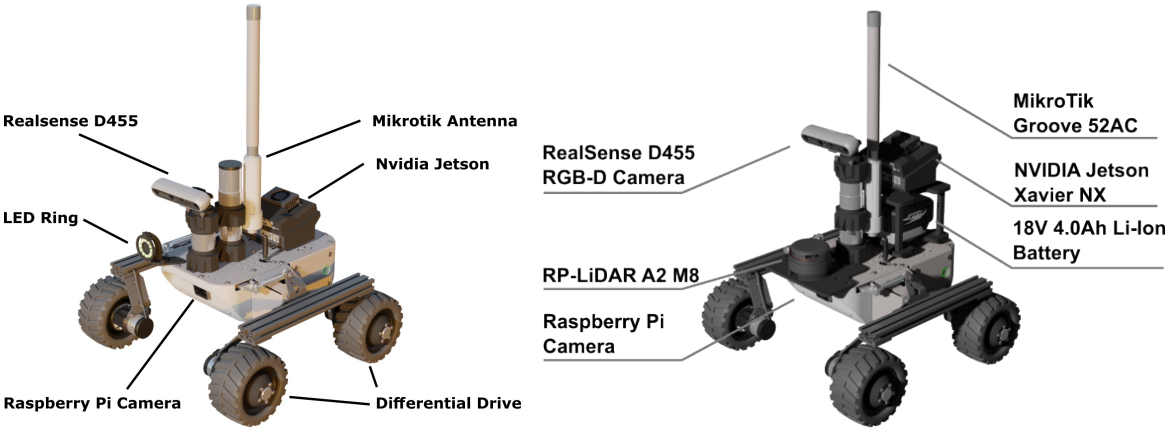
3.3.2 Proposed Solution

For the second trial, we suggest REALMS2 as a scalable MRS for space exploration. It features three main subsystems: the rovers, the lander, and the ground station for the

operators. Each operator can access the interface to send goals to each of the robots to explore the surroundings. The system is designed to address the challenges of a lunar mission, such as communication delays and the environment.

Compared to REALMS, the rovers have a more robust mapping system, more robust headlight controls, more processing power, and extended battery life. The user interface allows monitoring and controlling each robot through a single interface, making the control system more scalable and easier for a single operator to use. The upgrade to ROS 2 allows easier integration of additional robots and simplifies the communication protocol for MRS. The introduction of a lunar lander adds a central interface between the rovers and the ground station, as would be the case for a real lunar mission. The lander also acts as an additional resource for offloading high-computational processing and serves as a host for the sensors that overview the close environment and the rovers' departure.

Hardware comparison



(a) Leo rovers from the first challenge (b) Leo Rovers in REALMS2

Figure 3.11: Hardware configurations of the REALMS and REALMS2 rovers used in the ESA-ESRIC Space Resources Challenge.

As visible in Fig.3.11, REALMS2 relies on the same robotic base as the previous version by using a Leo Rover. However, it also features major improvements. To improve the mapping capabilities, we added a RPLIDAR A2M8 to complement the RGB-D Camera, giving sharper

details for 3D reconstruction.

As the Simultaneous Localisation And Mapping (SLAM) features and additional sensors require a lot of processing power, we added a new embedded computer, the NVIDIA Jetson Xavier NX, to process this amount of data. This computing board features a Graphical Processing Unit (GPU), allowing for a definite increase in the SLAM efficiency.

Additionally, powering this additional computing board and the sensors led to potential failure, so we added rechargeable drill batteries to power the newly added devices. This would create two separate power systems, leaving the internal battery only to power the control board and the motor, increasing autonomy drastically.

The biggest improvement to the software stack is the upgrade to ROS 2. ROS 2 is the successor to ROS, addressing some of the limitations we faced during the challenge, such as the struggle to set a multi-robot architecture. However, ROS 2 still faces some issues, such as a larger computational overhead.

3.3.3 Subsystems qualification

In order to qualify and validate the REALMS2 architecture, we conducted experiments in the two main additions to REALMS2: The network architecture and the map merging capabilities. As my work was mostly related to the network side, this section will only contain experiments related to the first.

Network Testing and Evaluation

REALMS2 presents one of the first applications of a mesh network architecture for a MRS that can be used in space. Due to this novel approach, the network capacities needed to be evaluated in order to develop the proper operational protocol for the missions. The three interesting points to quantify are:

- The maximum connectivity range between two robots.
- The influence of a relay robot between the operator and a teleoperated robot
- The impact of the mesh network on the ROS 2 communication architecture

Maximum Communication Range To evaluate the maximum range of the communication system using mesh routers, a robot was moved further away from the operator. The experimental setup for this is shown in the upper half of Fig. 3.12. The robot increases the distance to the operator until it loses connection. The distance measurement between the robot and the operator results in an estimated maximum range of $220\,\mathrm{m}$. The chosen environment featured a direct LOS scenario, with no direct obstruction, such as buildings, rocks or dense forests. A space exploration mission is not expected to have a high density of obstacles. For a short-range mission, there is no LOS that could be introduced by dunes or craters. According to [101], a crater would need to have a diameter exceeding $5\,\mathrm{km}$, and sometimes up to $20\,\mathrm{km}$, to be deep enough for its rim to act as an obstacle. It is worth noticing that the experimental site corresponds to an urban environment with some trees and moving obstacles. Therefore, the interferences can be considered more impactful than expected in a lunar scenario. As a result, the estimated value is a good baseline for the maximum range since the actual range would be higher in a real setup.

Mesh Network Routing Capability To verify the mesh network's capabilities for relaying communication, the robot has been placed at the edge of the communication reach, approximately $220\,\mathrm{m}$, to evaluate the network capacity. At this distance, a ping command or teleoperation is still feasible, but the communication is too limited for visualisation. A second rover is placed at $130\,\mathrm{m}$ away of the operator and $100\,\mathrm{m}$ of the teleoperated rover, as visualised in the lower half of Fig. 3.12. In this situation, it is possible to receive images from the teleoperated rover, and the evaluated bandwidth capacity would increase between 2 to 10 times.

This additional knowledge provides strategies for operating the robots during the mission. To ensure proper coverage, one of the robots should remain available to act as a relay and should not be operated on for more complicated tasks.

Integration test of ROS 2 in a mesh network The mesh network needs to support ROS 2 messages through relay nodes. The experiments highlight the differences in terms of network use between ROS 2 and its predecessor, ROS. Setting ROS for multi-robot usage in

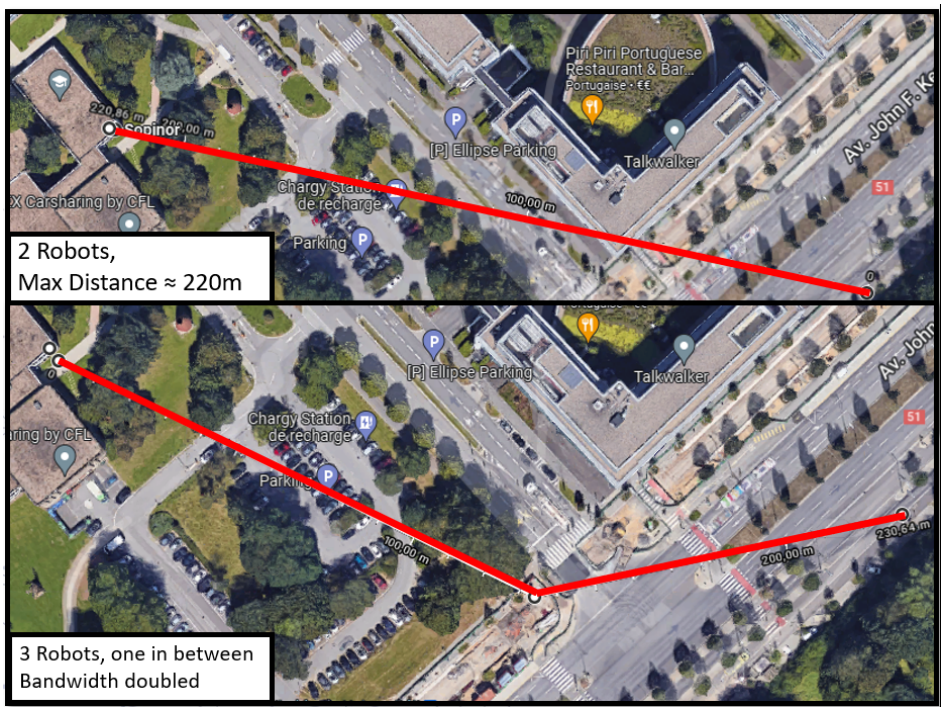


Figure 3.12: Two graphs displaying the positions of the robots during the network evaluation. On top, only two robots are used to measure the range. On the bottom one, a relay is placed in the middle, ensuring a better bandwidth

a decentralised way required some workaround relying on installing a non-official package. ROS 2 offers a more straightforward approach, providing native support for MRS. However, ROS 2 also comes with trade-offs. The experiments revealed that the ROS 2 network architecture introduces additional overhead on the message sent, leading to more bandwidth usage by each process. When testing the communication through the mesh network while simulating a communication delay, the messages were successfully transmitted using ROS 2. In REALMS, the communication delay prevented the rovers from connecting to the ROS master on the ground station. Consequently, the system was modified to provide a ROS master for each robot and ground station computer. This multi-master approach would rely on a non-standard approach, adding potential points of failure and reducing the scalability of the system. In ROS 2, this additional layer is not necessary to handle the delay.

3.3.4 Results at the ESA-ESRIC Challenge and discussion

The control room was connected to the lander through a network delay simulator as visible in Fig. 3.13. The goal was to explore the area and identify valuable resources during the 4-hour run.

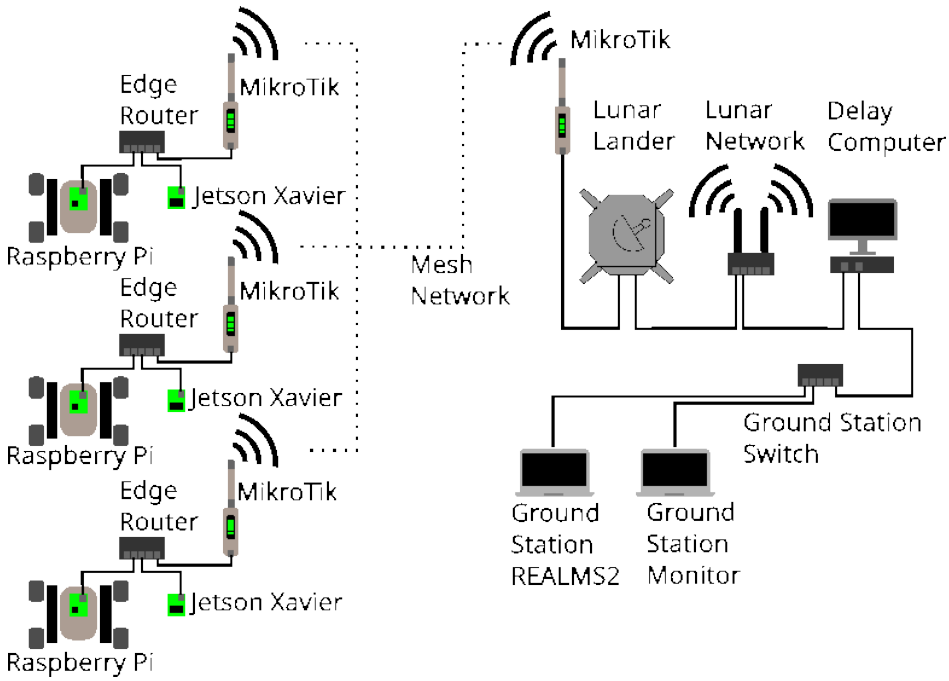


Figure 3.13: Overview of the REALMS2 Network Architecture

While the REALMS2 rovers were responsible for the mapping of the environment, the robot from Space Applications Services (SAS) analysed the rocks using spectroscopy. The Leo Rovers were used to look for potential resources and provide an overview of the terrain. When starting the mission, the autonomous navigation system of REALMS2 failed. To continue the mission, two operators manually controlled two of the rovers through teleoperation. During the challenge, the organisers requested that one rover be shut down to simulate a system failure so that they could verify resilience to unexpected events. After this, the third REALMS2 rover was used to replace the rover with the simulated failure. The lander represented the communication gateway between the rovers and the ground station. During simulated communication blackouts, teleoperation was not possible. Therefore, each rover sent data to

the lander to make efficient use of the bandwidth.

During the ESA-ESRIC Space Resource Challenge, REALMS2 was capable of mapping around 60% of the total surface of $1800\ m^2$. The three scouting robots encountered communication delays, blackouts, and a planned partial system failure to simulate the conditions of a real lunar mission. Fig. 3.14 shows the mapped area that highlights the contribution of each rover in blue, green, and red. Together, the rovers covered about 60% of the area. A minor scale issue can be observed in the blue part of the map in the bottom right corner. A possible explanation is the scale estimation drift introduced by the slightly higher movement speed of this rover when traversing the terrain. The map clearly shows two small craters on the left, a large crater on the bottom right, and several large rocks. Due to the sparsity of the map features, the map merging algorithm was unable to calculate the relative transformation. The rovers were controlled using teleoperation as the autonomous navigation stack failed to plan paths to the target locations of the rovers.

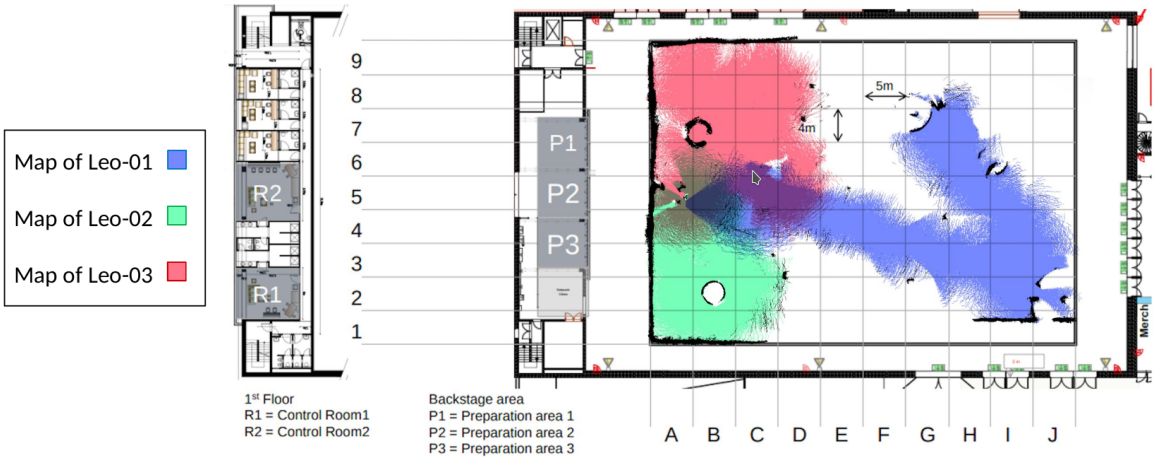


Figure 3.14: Area Mapped by REALMS2 during the ESA-ESRIC Space Resources Challenge

Discussion

The REALMS2 system shows some limitations during laboratory experiments and field tests. First, the map merging system only works in feature-rich environments, as it uses geometrical features in the map to find correspondences. A system based on rich image

data or scene semantics could reduce the dependence on geometric map features and reduce the overlap required to merge the maps. Second, odometry loss is not detected automatically and requires intervention from the operator. An automatic detection system followed by a predefined recovery behaviour could reduce the impact of temporary odometry loss. Third, the autonomous navigation stack needs to be more robust. If all three rovers were autonomously driving during the entire challenge, even more coverage could have been achieved.

Finally, we successfully established a reliable and resilient MRS that will be used for future works. However, as this system remains teleoperated, the following work will focus on higher autonomy scenarios.

3.4 Conclusion

The ESA-ESRIC Space Resources Challenge provided a unique opportunity to design a MRS dedicated to ISRU missions. Across both phases of the challenge, our systems were tested in increasingly complex and realistic lunar-analogue environments, leading us to develop a resilient approach and to highlight potential problems with future space MRS related works.

Despite the challenged faced, REALMS2, achieved promising results. It also demonstrated the use of the mesh network technology and a modular architecture as promising for future applications.

The experience gained from the challenge led to the confirmation of the three research questions previously stated.

- **RQ-1: How do robots collaborate autonomously with trust in a coopetitive system?**

  In the case of the challenge, the coordination was managed by a single operator on Earth. However, the system also highlighted the impact of the communication delay on operation, and in a setup where various competing actors would have to collaborate, it highlights the need of establishing autonomous coopetitive task allocation frameworks.

- **RQ-2: How efficiently can heterogeneous MRS communicate for space missions?**

During the second trial of the challenge, we had the opportunity of demonstrating a MRS using ROS 2 over a mesh network. However, we faced various network-related issues, mostly linked to the use of the standard ros middleware: FastDDS. A proper ROS 2 configuration for space mesh networks remains a question to be studied in the **RQ-2**.

- **RQ-3: How can the robots optimise the network topology?** As the challenge only happened indoors, the explorable distance was limited. However, questions are raised for the applicability in a real lunar setup, where the dimensions are scaled up, and one robot can't be retrieved when lost. It would require an additional layer of connectivity maintenance.

Chapter 4

Trustful multi-robot collaboration in coopetitive scenario

Related Publications

- Chovet, L.[†], Lima Baima, R.[†], Hartwich, E., Bera, A., Sedlmeir, J., Fridgen, G., Olivares-Mendez, M. A. **"Trustful Coopetitive Infrastructures for the New Space Exploration Era"**, *Proceedings of the 2024 ACM Symposium on Applied Computing (SAC24)*, 2024. `https://hdl.handle.net/10993/60481`
- Lima Baima, R., Chovet, L., Sedlmeir, J., Olivares-Mendez, M. A., Fridgen, G. **"Designing Trustful Cooperation Ecosystems is Key to the New Space Exploration Era"**, *Proceedings of the 2024 IEEE/ACM International Conference on Software Engineering (ICSE), New Ideas and Emerging Results*, In Press. `https://doi.org/10.48550/arXiv.2402.06036`
- Chovet, L., Lima Baima, R., Bera, A., Olivares-Mendez, M. A., Fridgen, G. **"Coopetitive Lunar Mapping: A Distributed Non-Proprietary Multi-Robot Coordination using Blockchain-Based Cost Optimisation"**, *IEEE Conference on Automation Science and Engineering (CASE)*, 2024. `https://hdl.handle.net/10993/62339`
- Lima Baima, R.[†], Chovet, L.[†], Olivares-Mendez, M. A., Fridgen, G. **"Towards Space Machine Economy: Assessing Non-Proprietary-Market-Based Multi-Robot Coordination"**, *Robotics and Automation Letters (RA-L)*, under submission.

[†] These authors contributed equally to this work.

"The free development of each is the condition for the free development of all."

– Karl Marx

4.1 Introduction

This chapter covers the following research question

RQ-1: *How do robots collaborate autonomously with trust in a coopetitive system?*

As highlighted in the introduction (Chapter: 1), with the growth of the space industry and the reduction of the exploitation costs, private and public institutions are targeting sending robots into space. MRS are expected on the Moon, allowing for easy scaling and offering more capabilities than single robots.

MRS involve groups of robots working together or supporting each other to accomplish specific tasks [102]. These systems can involve *homogeneous* or *heterogeneous* groups of robots and are traditionally categorised by their level of goal similarity, awareness of each other, and interaction as *collective*, *cooperative*, or *collaborative* [102]. The emergence of *coopetitive* systems, where competing agents simultaneously choose to cooperate owing to economic incentives, further expands the possibilities of MRS [103] in ISRU driven explicitly by economic incentives and shared objectives. Information-sharing protocols and market-based approaches can often improve coordination among MRS and robots' resource utilisation, cost-effectiveness, and exploration capabilities [51]. These systems can effectively mitigate adverse selection caused by information asymmetries, situations where few market participants know more about products (e.g. water or iron positions) or service quality (e.g. mapping data) than others [104], and as such, contribute to efficiency in the market [105] in line with the growing acceptance of this technology in space missions [106].

This chapter covers the implementation of a DLT-based coopetitive framework to the REALMS2 system (cf. section 3.3) in section 4.2. Then using the acquired knowledge we develop a coordination mechanism to ensure trustful coopetition in 4.3. An extended version of the previous work is detailed in 4.4, offering a task-agnostic system and improved

capabilities. The experiments are detailed in 4.5 and discussed in 4.6.

4.2 DLT for Coopetitive space MRS

Coordinating MRS in a market-based approach towards ISRU presents significant challenges due to the involvement of multiple competing entities and nations, with more than 60 countries involved in space activities [107]. Additionally, the legal [108] and technical requirements for planetary mobility systems further complicate the coordination of MRS in space exploration [72]. This work's technical and economic foundations highlight the need for robust and adaptable ISRU systems that dis-incentivise undesirable behaviour [109].

Additionally, stakeholders involved in space missions highly value the achievement of being the first to explore and acquire space and scientific data, so companies should aim to make mission outputs (e.g. videos, images, and audio) broadly accessible (e.g. through the web and media) [110]. This perspective resonates with the principles of open science, which promotes knowledge sharing, transparency, collaboration, and accessibility in research [111]. The evolving role of space ecosystems in shaping the future of space exploration [75] underscores the need for a decentralised (i.e. non-proprietary), trustworthy, and transparent digital platform. Such a platform can facilitate the seamless exchange of information and value among stakeholders [112], enabling autonomous MRS coordination for resource trading. DLT-based systems might offer a suitable solution for this specific requirement [58] and, as a consequence, have been considered the foundation of both space MRS [55] and open science platforms [113].

DLT offers several advantages for machine cooperation, including in MRS settings [114, 105]. These platforms can automate tasks such as bidding for resource usage, publicly broadcasting resource acquisition, and facilitating immediate operational cost compensation. With a DLT-based digital platform, entities can provide idle robot resources and stack intermediary profitable tasks to automatically compensate operational costs and openly recognise pioneering exploratory participants, thus increasing ISRU efficiency and enabling lower-cost space exploration. Despite the advantages, DLT in space MRS is not without challenges. For

instance, space robots face harsh conditions and limited resources, which conflict with the inefficient information processing of blockchains' intensive computation and storage replication [58, 73]. Furthermore, despite stakeholders' interest in open science [110], replicated information processing must still be aligned with the need to protect the sensitive business information of robots or organisations exposed through transactional (meta-) data [115].

Therefore, tensions exist between the opportunities and challenges of using DLT for space MRS, requiring closer investigation. This subsection aims to contribute to this understanding by designing an architecture for *coopetitive* MRS for ISRU, focusing on facilitating open science through DLT. The study investigates the technical feasibility of leveraging DLT to enable participation from universities, research institutes, and small companies in low-cost explorative scientific research.

4.2.1 Scenario

This subsection delves into our space mapping scenario that uses decentralisation and *coopetition* to enhance efficiency in mapping coordination among multiple robots. The purpose is to outline the scope and specific use case we will refer to throughout this section. We employed the earlier presented decentralised multi-robotic platform REALMS2 [116], to identify resources and analyse and map the environment [117].

To illustrate this setting more precisely, we revisit the market-based mapping scenario from [28] and illustrate the result in Fig. 4.1. This approach, proven effective in patrolling, exploration, and pick-and-delivery [51], is grounded in market-based strategies for MRS. For the sake of simplification, we are using exemplary amounts in the following scenario. The celestial surface stratification follows the existing Goldberg polyhedron approach [118] and uses the selenographic system to refer to its surface positions. An initiator stationed on Earth, the service orderer (SO), sends requests to a celestial stationed network of MRS SP, as listed in Tab. 4.1. The SO proposes to pay a maximum of $50 for mapping the target region. SP D is 5 m away, while SP C is 10 m from the target area. We consider an oversimplified abstraction of each robot's cost function, with a cost of $2 for each meter they travel. Consequently, SP D wins the contract by bidding $10 compared to $20.

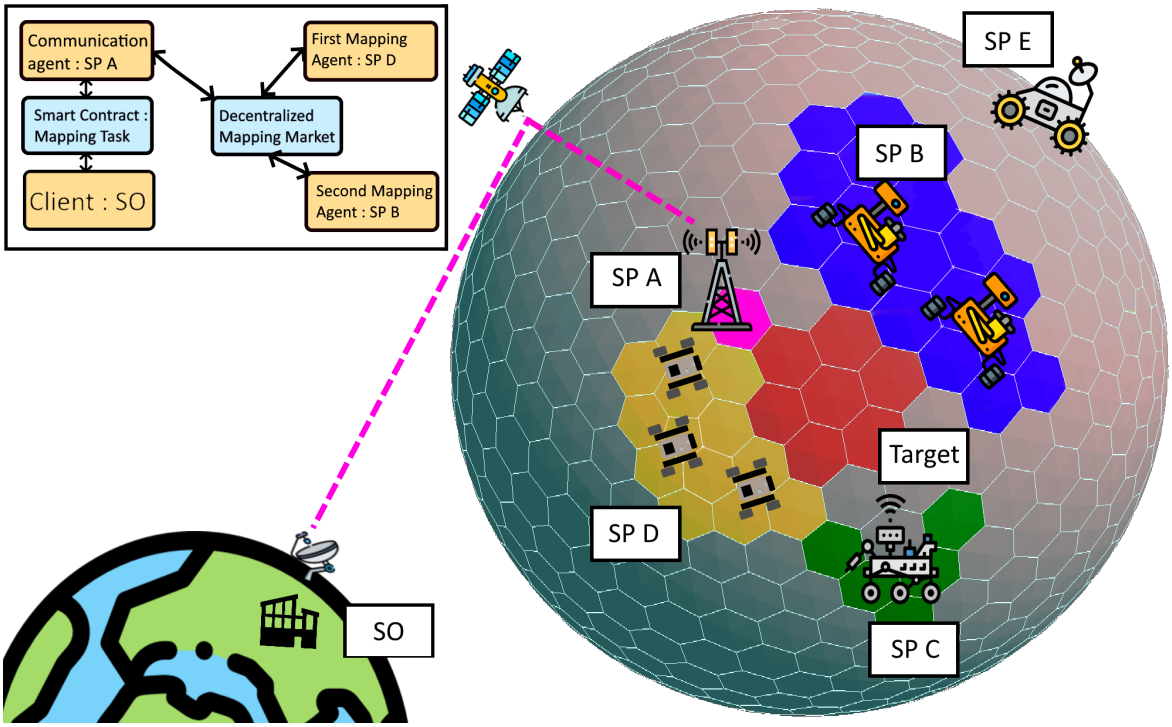


Figure 4.1: Moon's Goldberg polyhedron diagram, each zone colour represents a company's MRS operational area [118].

Table 4.1: List of service providers.

SP	Colour	Robotic Fleet	Price	Main Focus
A	Pink	Communications satellites, Multiple antennas	-	Moon-earth communications
B	Blue	Medium size offline fleet of robots	-	Mapping
C	Green	Few robots w/ embedded sensors	$20	Resource analysis
D	Yellow	Large fleet of small robots	$10	Fast mapping with less precision
E	None	A single robot	$200	Mapping

The primary need of ISRU SOs is to locate and investigate resources from celestial bodies [110]. This involves identifying areas with water or ice and mapping out the geological features of the planet's terrain. This role could often be triggered and motivated by universities and research institutions that seek to advance scientific knowledge and innovation in space. However, the scarcity of resources available to SOs due to high budget requirements for space-related research hampers advancements in this field. Incomplete mapping data is also an issue owing to the limited coverage of celestial geography [119]. Access to more precise information about celestial resources and terrain would significantly improve scientific outcomes [72].

The SPs, on the other hand, may represent start-ups [119, 120], large companies [121], space agencies [122], and organisations that already had some level of collaboration internally but not necessarily with one another. These entities are willing to work together to maximise their revenues and return on mission investment by offering resources, such as idle robots, shelter time, or energy supply. However, establishing trustworthy partnerships across various knowledge domains [61], including international agreements [108, 75] and individual initiatives [123, 124, 120], poses a challenge. Their struggle with inefficient use of exploration units, methods, and resources often results in wasted opportunities and increased costs [125]. The most significant values of these entities are in guaranteeing mission pioneering and promptly sharing exploratory data [110]. By following open science principles, they can encourage collaboration and innovation and gain access to valuable space exploration data [111]. We do not claim that these requirements are fully comprehensive or sufficient, yet we consider them necessary.

4.2.2 Solution Requirement

We conducted a case study and literature review to collect design requirements for our *coopetitive* MRS based on DLT and focused on ISRU. Although the requirement list is not final, we integrated insights to create a comprehensive list addressing the unique challenges and complexities of implementing DLT in an MRS for space. Our system architecture draws inspiration from the works of [126], [127], and requirements from [72] and [110]. It

follows the make-or-buy economic framework [53] while prioritizing transparency principles of open science [111] and the cost-efficiency characteristics of *coopetitive* settings [128]. Our technical design enables geographically distributed robots to self-coordinate [66] by using market-based strategies for service provisioning [51, 28, 52].

Functional requirements (FR):

1. Receive and process robots and SOs' map requests [52].
2. Implement a descending-price auction mechanism for robots to allocate mapping tasks [52, 51].
3. Verify the integrity and completeness of metadata associated with each job request [28].
4. Execute the payment process for completed tasks [28].
5. Provide an interface for managing and monitoring job requests during the mission [28].
6. Enable mapping and ISRU performance assessment [66, 72].
7. Ensure an adaptable robotic network compatible with the different robot types [28, 52, 66].
8. Support the seamless integration with possible existing infrastructures during space missions, such as space decentral [70], SpaceChain [129], and TruSat initiatives [130, 66].
9. Ensure compatibility with standard data formats and protocols for efficient data exchange [66].
10. Promote interoperability of robotic communication protocols to facilitate seamless coordination [66].

Non-functional requirements (NFR):

1. Conformity: Provide an interoperable economic framework interface for efficient market-based coordination [52].

2. Robustness: Protection mechanisms to safeguard against the inconsistency or loss of essential data owing to system failures or network disruptions [72, 28, 66].

3. Reliability: Enable agnostic data sharing among robots to ensure efficient coordination [28].

4. Openness: Foster a mission's public network environment, enabling relevant parties to participate on the platform [52].

5. Usability: Accessible job request statuses and pricing information for processing and decision-making [131].

6. Compatibility: Ensure metadata requirements-driven data standards to diverse mapping tasks and applications [28].

7. Maintainability: System design incentivises participants to take ownership of maintaining the network [66].

8. Portability: Keeping hardware requirements at a minimum to enable flexibility in deploying robot missions [132, 133, 134].

9. Interoperability: Ensure the system can handle traded data without significant adjustments [51, 52, 28].

4.2.3 Solution Architecture

Fig.4.2 features an overview of the architecture layers and their functions in our decentralised system architecture for job requests and contract creation. The process involves various entities, including the Client, MeshNetwork, Robot, TemplateSmartContract, JobSmartContract, and JobContract.

The lifecycle begins with the Client initiating a JobRequest by sending a message to the MeshNetwork. The MeshNetwork broadcasts the request to available Robots, and each Robot optimises and schedules its operation outside the blockchain. Once the scheduling is complete, the Robot sends the JobRequest through the TemplateSmartContract interface.

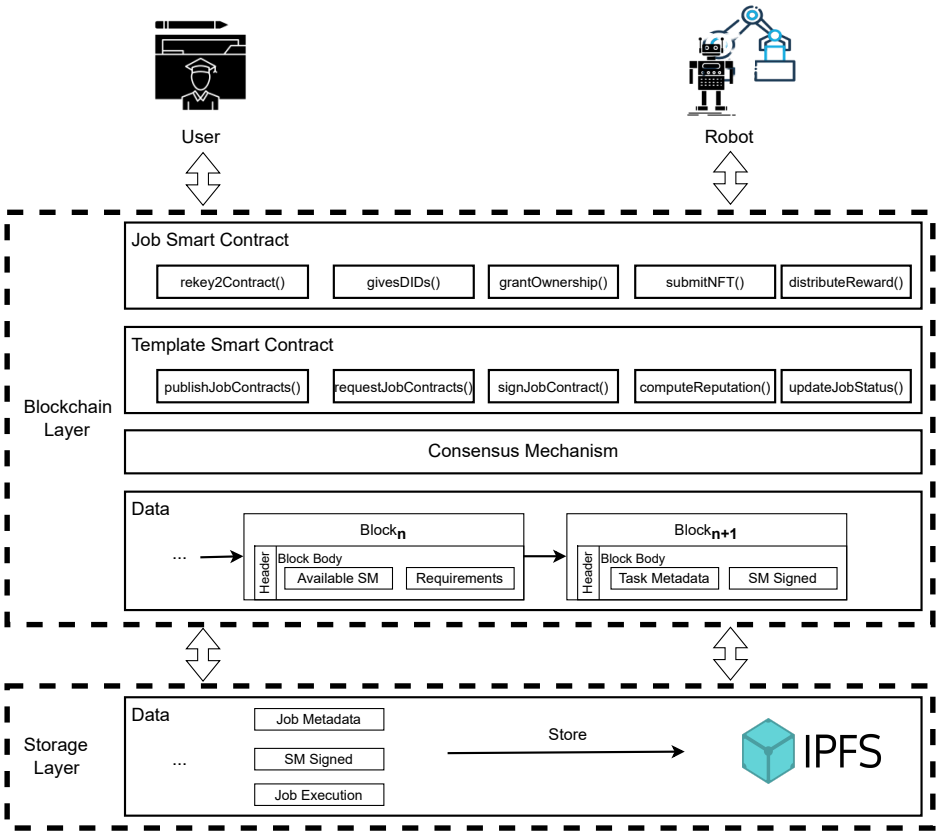


Figure 4.2: Architecture layers.

Upon receiving the JobRequest, the TemplateSmartContract creates a JobRequest's unique identifier. The TemplateSmartContract then shares the ID (transaction hash) of the JobRequest with the Robot. If at least one Robot decides to participate, a JobContract is generated and submitted to the TemplateSmartContract. The TemplateSmartContract updates the proposed job options from the JobContract with the most cost-effective proposal. The ID of the lowest-priced JobContract is returned and linked to the corresponding JobRequest, while it informs the current status and price to the Client.

Requirement validation

To ensure conformity ($NFR-1$), the economic automated decision-making is based on the make-or-buy decision framework [53]. The JobContract ($FR-4$), which records the execution

of mapping tasks and facilitates information trading, and the TemplateSmartContract acting as an interface for robots to post and monitor JobRequests ($FR-5$), facilitating efficient robot coordination and automating job assignment and compensation ($FR-1$) in a non-discriminatory environment ($FR-7$).

We initially used a mesh network for communication and control systems suitable for decentralised peer-to-peer and low-latency settings ($NFR-3$). However, we identified exploitation vulnerabilities regarding unauthorised data modification during transmission between peers. Given the high knowledge-sharing capabilities of *coopetitive* platforms [128], the system's integration allows for the secure and transparent sharing of mapping data among the participating robots, where robots add the data requirements (i.e. mapping location) to be validated by the smart contract ($NFR-4$) via the specific stratified Goldberg polyhedron registry position approach [118]. When followed, it ensures the reliability of the collected information ($FR-10$). The system publicly records who was the pioneer in being the first to explore, investigate, or discover an area or execute activities recognised as at the forefront of space exploration initiatives. As such, this verifiable record can help to build valuable partnerships and position universities as leaders in innovative interdisciplinary research. Additionally, a commission model can compensate the pioneer with every sale of the NFT associated with the map on secondary markets [63], translating into additional revenue streams.

InterPlanetary File System (IPFS) enables robust and decentralised storage of mapping data ($NFR-2$) and protects from loss or censorship ($FR-8$). It accepts standard data formats ($NFR-2$) and detects tampering through hash verification. While transactions within the blockchain are immutable, standard asset parameters can be modified. With these parameters, entities can confidently re-configure, destroy, mint, freeze, and even clawback addresses, ensuring that only authorised participants can modify the mapping metadata within the system ($NFR-7$). The history of these assets is immutably recorded in the blockchain and can be publicly verified. In our case, these systems do not depend on specific specialised hardware and enable near real-time processing in space-located facilities or, as we have successfully implemented, in a server-client structure that simplifies data

sharing ($NFR-8$). Thus, it affirms the supposed improved performance of a distributed, decentralised system consisting of multiple *coopetitive* robots compared to single individual mapping units.

During the bidding process, our robots engage in a descending-price auction. By posing a lower bid than the previous lowest bid recorded in the smart contract ($FR-2$), robots signal their willingness to execute a job. This descending-price bidding allows the robots to continuously reassess their capabilities and resources, resulting in an economically efficient solution [28]. The robots can negotiate multiple contracts simultaneously, and different robots can initiate new contracts to find new ways to execute previous contract proposals ($NFR-5$). As a result, several JobRequests may attach to multiple JobContracts, coordinated by several TemplateSmartContract instantiations ($NFR-9$). The TemplateSmartContract evaluates the proposals and ranks the options as it reach the time limit. The Client then considers the JobContract and decides whether to accept it. If accepted, an atomic payment transfer process is initiated by rekeying the contract signature to the TemplateSmartContract, potentially triggering multiple intermediate contracts. The respective bidding robot can set a maximum price threshold to ensure that auctioning outcomes align with it.

Regarding exploitation protection in this phase, to ensure the contract is not vulnerable to skipping payment after task execution, it is only considered ready for execution once the TemplateSmartContract receives the rekey power. Once granted, the contract can be signed, and the payment can be made automatically according to the contract agreement. However, if an attacker keeps bidding lower than others, there is no automated protection against spamming or the need to create new auctions. The solutions hence incorporate the bidder's reputation, which may be lower than others, and setting a time limit to restrict possible spamming attacks. These attacks may involve proposing multiple fake tasks or submitting fake lower bids, which the robot never intended to execute, or it just wanted to prevent the transaction from happening. Another potential exploitation is the failure to complete tasks upon which parties previously agreed. Even though our robots are automatically programmed to function independently, any deliberate harmful human interference can still disrupt their operations. Nevertheless, the physical limitations on communication time delays from Earth

and the celestial body and the smart contract time limit may offer some protection against such cases. Despite unintended malfunctions that may still occur, this physical protection, combined with the reputations of bidders, can improve trust in the platform.

Once the payment is received, the TemplateSmartContract triggers the participating robots to sign and execute the JobRequest and JobContract ($FR-4$). The robots carry out the specified tasks outlined in the contract: mapping or selling the map exploration rights while adding the asset description into the metadata and data ($NFR-6$). Before finalizing the contracts, the TemplateSmartContract conducts a plausibility check of the correctness and completeness of the mapping data retrieved from IPFS by assessing the requirements described on the JobRequest and the NFT metadata. This check ensures that the data adheres to the initial specifications outlined in the JobRequest without any unauthorised modifications, such as via the mutable asset characteristics. The possible mutability functions are: freezing an asset until the user meets a specific requirement, clawback by debiting a user's account for defaulting in loan payments, and revoking the ownership of assets belonging to users. Thus, the completion is successful if the robot receives the payment according to the due terms of the JobContract.

In the last stage, due to the NFT' (meta-) data, the Client is recognised as the pioneer explorer of the acquired mapping data or deliverables, signifying the process completion. The TemplateSmartContract verifies that the retrieved map metadata and IPFS data align with the original JobRequest and the acquired mapping information ($FR-3$). After the verification process, the Client can use the data for decision-making, knowing that the mapping coordination process was transparent and trustworthy. We implemented a prototype of our designed architecture to evaluate its practicality and discuss the lessons learned in Section 4.5.1.

Now that we have established a decentralised platform enabling multiple robots to coordinate and exchange mapping data in a trusted and transparent manner, the next critical step is enabling these robots to autonomously decide whether to produce such data themselves or acquire it from others. This challenge is particularly relevant in a coopetitive context where each robot, or the organisation behind it, seeks to maximise efficiency while preserving au-

tonomy. To address this, we adopt a make-or-buy decision-making framework, allowing each robot to evaluate the cost-benefit of executing a task versus outsourcing it. In the following section, We present the work published in CASE 2024, which introduces a cost-optimisation mechanism tailored for cooperative yet economically motivated MRS coordination, leveraging blockchain-based auctions and map tokenisation through NFTs.

4.3 The Make or Buy framework

To explore this decision-making process in practice, we developed a complete system that operationalises the make-or-buy framework within a blockchain-enabled multi-robot coordination architecture. This system allows each robot to autonomously evaluate whether it is more cost-effective to carry out a task or to purchase the data from another service provider. By combining robotic performance metrics with economic reasoning, our approach enhances the efficiency in ISRU missions while preserving the non-proprietary and transparent nature of the platform.

Historically, the coordination of MRS in decentralised exploration has been a research topic since the 1980s [135]. However, the market-based multi-agent coordination approaches, especially those emerging in the 2000s [28, 51], are particularly relevant for *coopetitive* space exploration scenarios. These approaches often employ auction systems, which have proven effective for space exploration [52]. Despite these market-based approaches being computationally intensive, the optimal planning problem for MRS remains *NP-Hard* [28]. This computational complexity and the latency in Earth-space communication render real-time decision-making impractical for space missions. This issue makes the potential of market-based approaches more apparent [136].

However, designing a universal cost/reward function for robotics is still one of the unresolved challenges in this context. Existing research has proposed various terms to enhance the bid valuation process [137], ranging from the most common distance-based metrics to performance indexes or computational complexity. Moreover, the communication architectures of multi-robot teams have also been improved by considering coupling constraints, energy

capacities, time windows, and possible coalitions. The computational complexity of combinatorial auctions is also recognised as highly intensive [138]. Thus, works have identified solutions, from simple single-item auctions to the more complex consensus-based bundle algorithms. However, these solutions often rely on proprietary and centralised frameworks, lacking the transparency and trust economic digital platform required in the emerging space economy [75], nor try outsourcing to other companies' MRS that may be willing to sell their data via a make-or-buy decision framework [53].

In light of these challenges and opportunities, this section aims to display the following contributions.

- We propose the make-or-buy decision framework [53] to guide robots in deciding whether to perform or outsource !apping tasks to enhance ISRU efficiency.
- We leverage blockchain technology to automate the lower complexity MRS coordination through an *coopetitive* single-item auction-based system.
- We propose a first cost function that allows each robot to offer its most competitive price for the area to be mapped, given its constraints focusing on reducing the global travelled distance. The blockchain programmatically lists the cheapest options for external buyers

Through this work, our aim is to provide insight into the adaptability of MRS in the new space economy.

4.3.1 Scenario

The scenario is the same as presented in the section 4.2.1.

We distinguish two kinds of organisations: the service orderer (SO) and the service provider (SP). SO, SP A, SP B, SP C, SP D, and SP E are involved in the system.

- Organisation SO is interested in a piece of unmapped territory on a celestial body.
- SP A provides communication services.
- SP B, SP C, SP D, and SP E are independent and potential competitors. Each operates its own MRS fleet.

- All the robots maintain a global lunar network

When an SO desires to map a specific lunar area to prospect some resources, such as in Fig.4.1, the obvious solution would be to set up and develop a whole robotic mission. However, if robots are already mapping the Moon, they might propose their mapping functionality as a service. The scenario requires a set of robots R, such as

$$R = \{r_1, r_2, r_3, \ldots, r_n\}$$

where n robots are interconnected on the same network. Each robot is owned by a company that might not share common economic interests with the other, and each has its own capabilities. The study focuses on how to minimise the global cost for the user.

To answer this scenario, we propose the following:

- The SO requests a mapping task, specifying the required information and level of detail they need.
- The SP B, SP C, and SP D MRS proposes prices meeting SO's requirements.
- The decentralised platform facilitates the bidding process, ensuring fair competition and transparency. It executes the smart contracts between SO and the organisation (SP B, C or D) that provide the best solution.
- Once a mapping task is complete, data is securely stored using NFT, ensuring its immutability and accessibility to all stakeholders.
- Other organisations and researchers can access the mapped data, enabling collaboration, validation, and further scientific studies.
- The decentralised nature of the platform ensures that all participating entities have equal access to the data and can verify the results independently.

4.3.2 Cost Function

We propose optimising the ISRU mapping process by integrating MRS with economic factors through a cost function.

System Flexibility: It is essential to highlight that this system allows every robot to include its cost function, depending on the system's. However, we propose a general cost function with adaptable parameters for each robot.

Proposed Cost Function: For a given robot r_i, there is a cost function C_i which gives the price (in the decided monetary unit) the robot asks to operate a given task. The cost function is the following:

$$C_i = E \times F_E \times LU \times P_u \tag{4.1}$$

Where:

- E: Represents the energy consumed by the robot during the mapping process with:

$$E = \Delta D \times E_d \tag{4.2}$$

  - ΔD is the distance to perform the task given by the global planner of the robot. It is a planned distance that can be less precise due to terrain irregularities.
  - E_d represents the energy consumption per unit of distance, part of the design of every robotic mission (e.g. the Scarab mission from NASA [139])

- F_E is the fee per unit of energy consumed

- LU: provides a straight-line lifespan depreciation proportion of how much of a robot is depreciated by performing a specific task, expressed as:

$$LU = \left(\frac{\Delta t}{L} + 1 \right) \tag{4.3}$$

  - Δt is the estimated time to perform the task $\Delta t = \frac{\Delta D}{V} + t_{task}$ with V the estimated velocity of the robot. t_{task} is an additional constant time in case the robot needs to

Vehicle	Planet	Lifespan
LRV	Moon	78h
Lunokhod	Moon	3 month
MFEX/Sojourner	Mars	7 sols (7.19 days)
Mars exploration rover	Mars	90 sols
MSL Curiosity	Mars	670 sols
Yutu	Moon	90 days
Exomars	Mars	218 sols
Mars 2020	Mars	836 sols
Dragonfly	Titan	2.7 years

Table 4.2: Space robotic mission lifespan planned

stay in place to perform the task. In the case of mapping, this value is null since the robot is always moving between waypoints.

- L denotes the expected lifespan of a robot in space missions with the level 1 requirement of 90 sols. As detailed in Tab. 4.2, past missions have ranged from 78 hours to 2.7 years. Notably, the last third quartile approaches two years [72].

- P_u: Embodies the navigational uncertainty as a Gaussian distribution [140].

The final formula of C_i for a given task is the following.

$$C_i = \Delta D \times E_d \times F_E \times P_u \times \left(\frac{\Delta D}{V \times L} + 1 \right) \tag{4.4}$$

The framework guides robots in deciding whether to acquire or manufacture maps. It is based on the “make or buy” decision framework [53], which consists of seven steps. We adapted these steps to account for the trigger (SO ordering) and proceed by analysing internal processes (does the robot already own the map, or can it outsource), cost estimations (acquisition and manufacturing estimations), supply chain management (verify who has the mapped area), and support systems (quality and technical supports). After going through all the behaviour steps, our system considers performance measurements closely linked to the triggers, which means the system aims to evaluate the extent to which the task targets ordered by the triggers (SO) are met. Since our trigger is cost reduction, cost-saving should be the primary performance measure while considering flexibility, resource usage, and quality

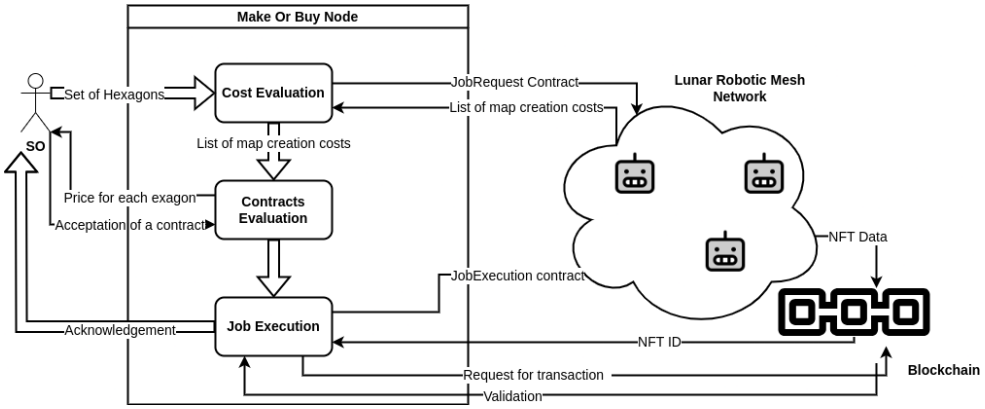


Figure 4.3: Architecture of the Make or Buy client node, featuring the SO and the Robotic network, on the Moon.

measurements such as energy and time. Furthermore, we assess the mapped area based on clients' requirements. Performance measures assess the extent to which the targets suggested by the triggers are achieved, such as cost reduction or quality. We aim to reach the lowest price and highest global ISRU efficiency.

Actors of the solution:

- Operator: A company or individual desiring to buy maps
- Client Node: A ROS node, operating like the blockchain as explained in section 4.3.3.
- Robots 0 to n: All the robots connected to the network
- Algorand blockchain: The implemented blockchain

The process is encapsulated in three primary stages as visible in Fig. 4.3:

1. **Cost Evaluation**:
    - The operator requests a set of maps represented by hexagons as detailed in 4.3.3
    - The client node generates a "Job Request" contract, which is disseminated to all robots for cost estimation.
    - Each robot evaluates its map creation costs locally, which are relayed to the client node.

2. **Contract Bidding Creation**:

    - Based on the collected cost evaluations, the client node ranks the job contracts and proposes them to the operator. It periodically recomputes until the operator accepts the job for the selected hexagon.
    - Post acceptance, the map-making initiative is triggered, creating a “Job contract” with the robot.

3. **Execution of Job Contract**:

    - If needed, the designated robot makes the map and uploads it as an NFT.
    - This robot then conveys the NFT and job metadata to the Client Node for validation.
    - Upon successful validation, the client Node oversees the payment procedure, and the map’s ownership NFT is transferred to the operator.

In summary, our framework introduces a streamlined “Make or Buy” decision-making methodology [53], matching robot capabilities with blockchain’s transparent notoriety.

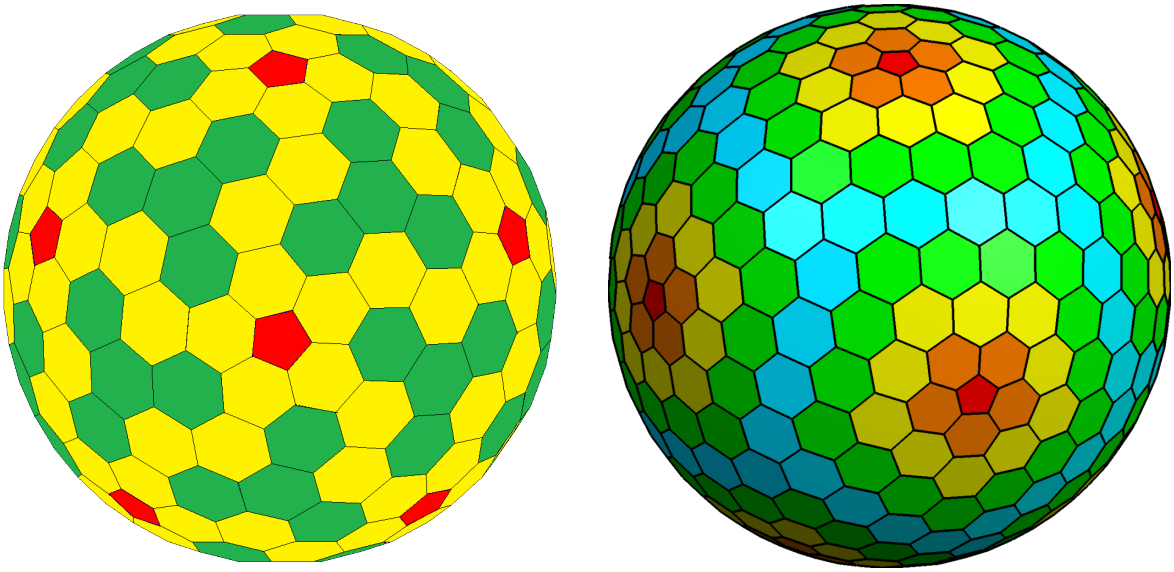
4.3.3 NFT and Goldberg sphere for Moon

The use of the blockchain covers two main functionalities: ensuring a reliable auctioning system between competing actors and authenticating maps as NFT. The reliable auctioning protocol should rely on the smart contract technology. However, this current implementation offers a ROS node that simulates the concept of smart contracts as detailed in [136], called the **client node**.

To save maps as NFT, we propose a novel approach inspired by [118] visible in Fig. 4.1 and Fig 4.4.

A Goldberg polyhedron is a solution to subdivide a sphere into H hexagons and a fixed number of $P = 12$ pentagons [141]. This solution especially fits our approach due to the following properties:

- **Uniformity**: Each cell has a fixed size, making it easier to make defined NFT collection-sharing properties



(a) Goldberg Polyhedron with the parameters $\{l = 1;\ m = 4\}$.

(b) Goldberg Polyhedron with the parameters $\{l = 3;\ m = 5\}$.

Figure 4.4: Examples of Goldberg polyhedrons used in task space discretisation.

- **Geometry**: Hexagons offer the possibility to tesselate well, covering the designated surface, providing an easy set of coordinates assuring the unicity of each NFT
- **Scalability**: A Goldberg sphere is defined by two parameters l and m that can be adjusted for any astral body and cell size.

The Goldberg parameters l and m are the number of divisions along the triangular edges of an icosahedron's face. Fig. 4.4 represents two examples of Goldberg Polyhedrons with two different sets of parameters l and m. Being a widely studied topic, it is easy to calculate the parameters l and m to get a given cell size by knowing the size of the Moon [141].

The subdivision of the mapping zone as a subset of multiple hexagons allows the turn of a single **Multi-Task Robots (MT)** to a set of **Single Task Robots (ST)**. It allows us to have a subset of **ST-MR-IA** problems, described in [142] as the simpler problem.

In this implementation, the mapped hexagons are saved on IPFS using the Pinata interface and then uploaded on the Algorand test blockchain.

4.3.4 Discussion

This work, presented at the CASE'24 conference, proposed a novel "make-or-buy" decision-making framework, leveraging blockchain-based auctions and a cost function centered on distance and energy consumption. While well received by the robotics community, two key limitations were raised: (i) the system lacked task diversity and was tailored primarily for mapping tasks, and (ii) it did not account for task durations where robots remain stationary, such as imaging or computation. These limitations prompted a reevaluation and extension of the framework to address a wider array of ISRU scenarios.

4.4 The Make-or-Buy Framework Extended: Task Agnosticism and Improved Cost Function

Building upon the results and feedback received from the initial framework presented in the CASE'24 paper, we extend the make-or-buy approach to better reflect the operational realities and complexities of in-situ resource utilisation (ISRU) missions. While the original framework demonstrated the viability of decentralised market-based coordination in a cooperative-competitive (coopetitive) multi-robot setting—achieving a 17.5% reduction in travel distance—it was primarily focused on mapping tasks and relied on a cost model driven by mobility-related factors. This design limited its applicability to a broader range of robotic services and under-represented scenarios where task execution requires little or no movement, such as long-duration measurements or onboard data processing.

In response, we present an expanded version of the make-or-buy framework that introduces two major contributions: task agnosticism and an improved cost function. First, the updated architecture supports heterogeneous tasks with varying execution profiles (e.g. energy-intensive, time-consuming, stationary). Second, the cost function is extended to incorporate multiple operational dimensions—such as energy usage, task duration, hardware depreciation, uncertainty, and contract overhead—enabling robots to autonomously evaluate whether to execute or outsource a task based on economically-grounded decision-making.

Table 4.3: Robotic SP and Main Capabilities

SP	Colour	Robotic Fleet	Main Focus
SP0	None/Lander	Antennas & satellites	Moon-Earth communication
SP1	Red	Specialised robot	Resource analysis & mapping
SP2	Yellow	Medium fleet	Rapid low-precision mapping
SP3	Green	Medium offline fleet	High-precision mapping
SP4	Blue	Large agile fleet	Fast low-resolution mapping

This multi-parameter pricing strategy enables a more flexible and realistic coordination model, reflecting not only spatial but also temporal and strategic trade-offs.

This extension moves the system closer to the vision of autonomous machine economies, where robots act as self-governed agents negotiating task ownership in real time, without relying on centralised orchestration. Inspired by the make-or-buy principle in economics and adapted for decentralised robotics, our approach allows MRS to dynamically self-organise under resource constraints and incomplete information, while maintaining transparency and trust through distributed ledger technologies.

The updated framework has been validated through a combination of high-fidelity virtual lunar simulations and physical lunar-analogue laboratory experiments. These experiments assess the scalability, robustness, and economic rationality of the coordination strategy across diverse robotic agents and ISRU-relevant task types. The results demonstrate that our system can approach the performance of centralised solvers while maintaining the benefits of decentralisation—namely resilience to failures, avoidance of monopolistic bottlenecks, and adaptability to emerging conditions.

In the following sections, we detail the improved scenario, architectural upgrades and the improved cost function.

4.4.1 Updated scenario

To validate and extend the applicability of the make-or-buy framework beyond mapping tasks, we define an updated scenario that generalises the previous use case introduced in Sec-

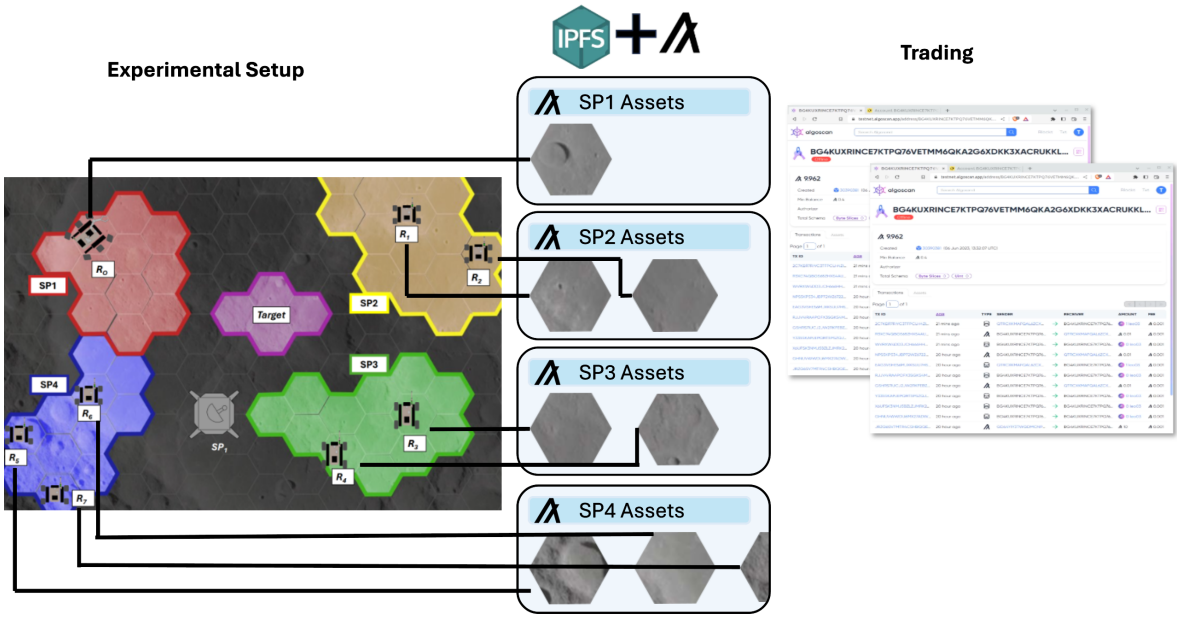


Figure 4.5: Scenario use case. The Moon is cartographically mapped via a Goldberg polyhedron diagram, each colour represents a SP MRS operational area

tions 4.2.1 and 4.3.1. While the initial scenario focused exclusively on decentralised allocation of mapping requests among competing service providers (SPs), the revised scenario expands the system to support a diverse set of ISRU-relevant robotic services, including imaging, data processing, and other static or long-duration tasks.

As illustrated in Fig.4.5, the scenario features a single service orderer (SO) and five robotic service providers (SP0 through SP4), each operating its own MRS fleet with varying capabilities and specialisations as described in Tab.4.3. These SPs represent different organisations—public or private, large or small—already deployed on a planetary surface and available for task execution or data trading. Compared to the prior setting, which assumed homogeneous task types and comparable robotic behaviours, this updated configuration introduces heterogeneity both in task demands and in robotic capacities (e.g. sensing quality, computational power, mobility constraints).

Each robotic service provider may autonomously decide whether to execute a given task request internally (“make”) or outsource it to another agent (“buy”), based on a refined cost-benefit analysis encompassing energy usage, task duration, hardware depreciation, and

execution uncertainty. These decisions are facilitated by a decentralised auction mechanism, where agents continuously evaluate their suitability for upcoming jobs using the improved cost function (Section 4.4.2).

This new scenario more realistically captures the operational diversity expected in future ISRU missions and better aligns with the notion of a non-proprietary machine economy. In particular, it supports:

- **Task diversity**: Jobs are no longer limited to mapping but may also include static observation tasks, processing of previously gathered data, or composite workflows.
- **Capability-aware bidding**: Robots assess each task based on their specific strengths (e.g. high-resolution imaging vs. fast mobility) rather than treating all tasks equally.
- **Autonomous market participation**: Each SP makes decentralised economic decisions based on its own internal model and resource availability, fostering a dynamic, agent-driven task market.

The scenario also highlights the transition from simple coordination to economic agency, where autonomous systems act not only as executors of predefined roles but as participants in a dynamic task economy.

4.4.2 Improved cost function

A robust and accurate cost function is essential for effectively coordinating tasks and establishing hierarchies in decentralised, market-based MRS. While the initial version presented in the CASE'24 framework prioritised simplicity by focusing on energy consumption and distance travelled, it lacked sensitivity to task diversity and time-based economic constraints. Specifically, the original model assumed that all tasks involved motion (e.g. mapping), and therefore excluded task durations or stationary operations from the cost calculation. This limited its applicability to broader ISRU scenarios where robots may need to stay in place for prolonged sensing, computation, or data transmission.

To address these limitations, we introduce an improved, multi-parameter cost function designed to handle heterogeneous tasks and better reflect real-world operational trade-offs.

Unlike state-of-the-art solutions that rely on single-dimensional metrics such as distance or battery usage [143], our model integrates energy, time, depreciation, uncertainty, and operational overheads into a single coherent economic expression. Each robot autonomously computes its cost using internal datasheet parameters—such as energy per distance, maximum velocity, and expected lifetime—and recalculates bids in real-time based on its current resource availability and the nature of the task.

Improved Cost Function: Each robot r_i bids a price (in chosen monetary unit) corresponding to executing a task calculated by the following function:

$$C_i = (E \times F_E + \Delta t \times F_T) \times LU \times P_u + \Delta\$ \tag{4.5}$$

This formula introduces several new components compared to the initial function:

- The **energy term** $E \times F_E$ retains the original energy-related cost, but now complements it with a **time-based term** $\Delta t \times F_T$ to account for cases where robots spend time without moving—critical for tasks such as imaging, data processing, or long-range communications.
- The **lifetime usage factor** $LU = \frac{\Delta t}{L}$ explicitly quantifies hardware wear over time, scaling both energy and time cost contributions based on the expected mission lifespan L (typically 90 sols).
- The **uncertainty factor** P_u models navigational risks as a probabilistic term, capturing the likelihood of environmental or estimation errors affecting execution cost.
- Finally, $\Delta\$$ encapsulates **operational overheads**, such as blockchain transaction fees, network coordination latency, or contract negotiation complexity.

The cost parameters are detailed in Tab. 4.4. Notably, this formulation supports greater flexibility and expressiveness by decoupling execution cost from specific task types. Robots can now accurately price both mobile and stationary tasks, encouraging specialisation and fairer competition in heterogeneous MRS environments.

Table 4.4: Cost Function Parameters

Symbol	Description	Calculation or Source
E	Total energy spent	$(D_{task} + D_{exec}) \times E_d$
D_{task}	Distance to task location	Planner output
D_{exec}	Travel distance during task	Based on task type
E_d	Energy per distance unit	Robot datasheet
Δt	Total task duration	$\frac{D_{task}+D_{exec}}{V} + t_{task}$
V	Average Robot velocity	Robot Datasheet
t_{task}	Stationary task time	Zero for mapping; predefined for imaging or processing (datasheet)
F_E	Energy cost rate	Externally economically supplied
F_T	Time cost rate	Economic model dependent
LU	Lifetime usage factor	$\frac{\Delta t}{L}$ (typical $L = 90$ sols)
P_u	Navigation uncertainty	Gaussian uncertainty model
$\Delta\$$	Operational fees	Contract transactions overhead

This comprehensive model allows the framework to scale toward more general applications in space robotics, such as on-orbit servicing or surface science payloads, where time and resource heterogeneity play a decisive role in task distribution.

4.4.3 Improved Make Or Buy Implementation

Our decentralized MRS uses automated auction-based task allocation for efficient economic coordination of heterogeneous ISRU planetary exploration tasks. The adapted *make-or-buy* decision-making framework [144], adhering to the space industry's standards [145], is summarized in eight steps in Tab. 4.5.

Robots and the network use this logic to negotiate task execution through descending-price auctions and continuously re-evaluate bids based on task urgency, current workload, resource constraints, and comparative advantages [146].

The flow diagram (Fig. 4.6) illustrates the message-level passing of a typical algorithm execution. The steps are more clearly described as follows:

1. **Job Initiation**:

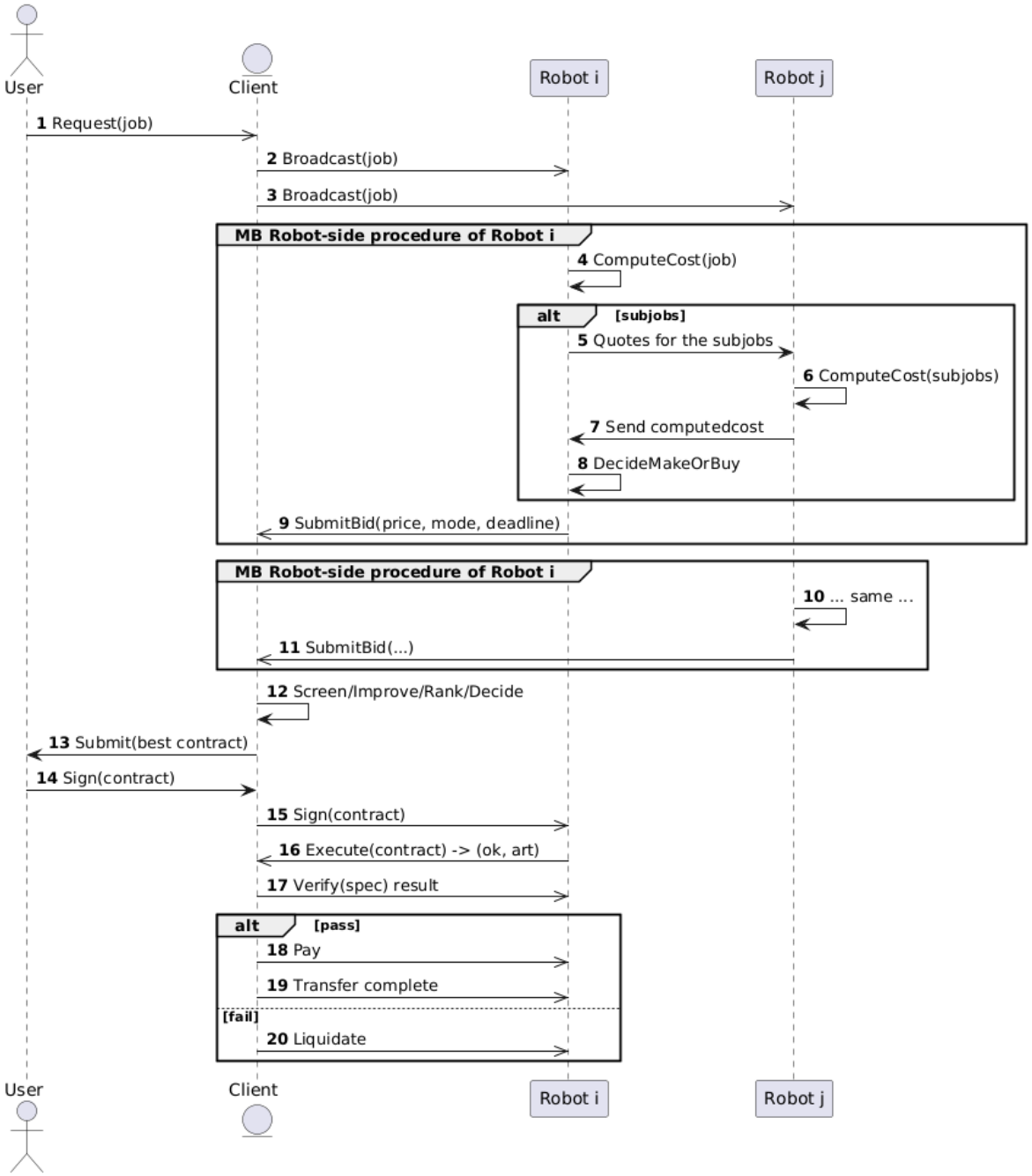


Figure 4.6: System Execution Flow Diagram

Table 4.5: Economic *Make-or-Buy* Automated Behavior

Step	Framework Stage	Automated Robot Decision-Making
1	Trigger Identification	Detect and evaluate new JobRequest
2	Capability Check	Assess owned assets (partial or all) and capability
3	Cost Estimation (Make)	Estimate internal cost (distance, time, energy)
4	Cost Evaluation (Buy)	Evaluate outsourcing cost from bids
5	Supply-Chain Check	Check registry for completed peer tasks & bids
6	Support System	Compare internal cost vs external bids
7	Decision-Making	Submit or sign lower TemplateContract bid
8	Performance Review	Verify compliance with specs (sensors, resolution)

- The Initiator, a Client (SO) or Robots, sends a job request (with task description) to the decentralized network or broadcasts to the mesh network.
- The MRS network assesses robot availability, entering if robots are available or if none are found after a deadline.

2. **Autonomous Economic Evaluation (*Make-or-Buy*):**

   - Idle Robots assesses JobDescription requests based on resource availability, collateral, execution cost, and economic efficiency (distance, energy, task duration, robot heterogeneity, and costs), calculating bids using the cost function.
   - Robots autonomously decide to opt in or out of the task, moving back to Idle or sending a Job Contract following the TemplateContract requests. Conditioned to:
     - If can only partially perform the task, initiate another subtask TemplateContract request, and add it to the initial contract's execution chain.
     - If can complete the task, submit proposals.
   - Network stars decentralized auction coordination.

3. **Decentralized Auction Coordination**:

   - TemplateContract transparently manages proposals submitted by all Robots willing to participate, creates JobContract, and makes them active for client consideration.

- Robots and Clients autonomously reassess these contracts' price, while TemplateContract rearranges in a descending-price bidding order representing resource efficiency.
- Upon auction expiration, the Initiator evaluates to accept or decline the active JobContract.

4. **Task Execution and Asset Ownership Transfer:**

   - Upon contract acceptance, Client and Robot sign the contract, allowing for automated payment and ownership transference once the job is executed, or collateral if not.
   - Robots autonomously perform assigned tasks,
   - NFT-based ownership is stored and transferred using IPFS decentralized storage, representing pioneering exploration and mapping records.
   - TemplateContract monitors JobContract execution compliance with TemplateContract tasks:
     - When it complies, Clients may confirm, optionally providing JobContract satisfaction ranking.
     - When it doesn't comply, collateral is transactioned as compensation.
   - Transaction finalized; system returns to Idle.

The robot-side bidding logic is detailed in Algorithm 1; the client-side auction and settlement are in Algorithm 2.

4.5 Experiments

After detailing the architecture and decision-making principles underpinning our coopetitive framework, we now focus on empirically validating the system's core functionalities. This section presents a series of experiments designed to evaluate both the technical feasibility and economic efficacy of the proposed approach, in simulated and real-world lunar-analogue conditions.

Algorithm 1 Decentralized Make-or-Buy — Robot-side procedure (running on each robot)

Inputs:
job $= (\text{task}, \text{spec}, \text{loc}, \text{deadline}, \text{reserve})$
energy rate F_E; time rate F_T; transaction fee $\Delta\$$

procedure MAKEORBUYBID($job,\ F_E,\ F_T,\ \Delta\$$)
  if not CAPABLE(self, job) **then** ▷ busy or not capable
    return No bid
  end if (Step 2 Capability)
  $C_{\text{make}} \leftarrow$ COSTCOMPUTATION(self, $job,\ F_E,\ F_T,\ \Delta\$$) (Step 3 Cost(Make))
  (can_split, subJobs) $\leftarrow$ SUBDIVIDE(job)
  if can_split **then**
    $C_{\text{buy}} \leftarrow$ QUOTES(subJobs)
    if $C_{\text{buy}} = \infty$ **then**
      return No bid
    end if
    $C \leftarrow C_{\text{make}} + C_{\text{buy}}$; mode $\leftarrow$ Hybrid
  else
    $C \leftarrow C_{\text{make}}$; mode $\leftarrow$ Make
  end if
  price $\leftarrow$ PRICE(C)
  SUBMITBID(self, price, mode, job.deadline) (Step 4 Submit)
end procedure

Our objective is twofold. First, we assess whether decentralised coordination via DLT can reliably support task sharing and data exchange among heterogeneous robotic agents under constrained environmental and communication conditions. Second, we evaluate the capacity of the make-or-buy decision model—under both its initial and extended formulations—to yield economically rational task allocations in various ISRU-relevant scenarios.

To this end, the experiments are structured in three phases:

- **Section 4.5.1** examines the real-world applicability of our blockchain-based coordination system using a lunar-analogue testbed, focusing on data handling, NFT exchange, and system latency.
- **Section 4.5.2** validates the initial make-or-buy cost function in a controlled simulation focused on decentralised mapping, evaluating distance minimisation and pricing consistency.

Algorithm 2 Decentralized Make-or-Buy — Client-side procedure (auction & execution)

Inputs:
job $= (\text{task}, \text{spec}, \text{loc}, \text{deadline}, \text{reserve})$
T_max ▷ auction timeout / latest bid cutoff
screening_policy, *improvement_policy*, *acceptance_rule* ▷ procedural hooks

procedure RUNPROCUREMENT(job)
  $t_0 \leftarrow$ NOW
  BROADCAST(job) (Step 1 Trigger)
  bids $\leftarrow \emptyset$
  while NOW $\leq t_0 + T_{\max}$ **and not** ALLBIDSRECEIVED **do**
    bids $\leftarrow$ bids $\cup$ COLLECTBIDS ▷ accumulate, do not overwrite
    bids $\leftarrow$ SUPPLYCHAINSCREEN(bids, screening_policy)
    bids $\leftarrow$ IMPROVEBIDS(bids, improvement_policy) (Steps 5–6 Supply & Support)
  end while
  if ISEMPTY(bids) **then**
    return No award
  end if
  RANKASCENDING(bids) ▷ e.g., by price, then risk
  if BESTPRICE(bids) $\leq$ job.reserve **or** CLIENTACCEPTS(bids, acceptance_rule) **then**
    winner $\leftarrow$ BESTVALIDBID(bids)
    c $\leftarrow$ SIGN(job, winner) (Step 7 Decision)
  else
    return No award
  end if
  $(ok, art) \leftarrow$ EXECUTE(winner, c)
  pass $\leftarrow$ VERIFY(art, job.spec) (Step 8 Review)
  if ok **and** pass **then**
    PAY(winner)
    TRANSFER(art)
    LOG(success)
  else
    LIQUIDATE(c)
    LOG(fail)
  end if
end procedure

- **Section 4.5.3** evaluates the enhanced framework supporting heterogeneous tasks and the improved cost function. This test extends the scenario beyond mapping to capture richer task profiles and economic dynamics.

Together, these experiments offer a comprehensive view of our system's robustness, scal-

ability, and alignment with the principles of machine economies and decentralised decision-making in future planetary exploration missions.

4.5.1 Real world DLT and architecture validation

This evaluation of our system aimed to test and validate its accuracy and resistance in simulated lunar conditions, explicitly focusing on assessing space-related limitations, such as latency and resource constraints. We conducted an experiment to evaluate the ability of the architectural solution to collect, coordinate, and maintain mapped area data. We presented a proof-of-concept conducted in the lunalab of the university of luxembourg. Each test had an approximate duration of 30 minutes. The robotic platform used RtabMap, a visual-SLAM software to create a map of the analogue lunar terrain facility. The Lunalab, a $80\,\mathrm{m}^2$ rectangle filled with basalt, was designed to emulate the surface of the Moon's south pole, an area where researchers expect to find valuable resources for ISRU.

The primary focus of this experiment is to assess the technical feasibility and performance of the architecture. The system's architecture, the one presented in 4.2.3 illustrated in Fig.4.7, is designed for efficient data storage, NFT management, and trading with 3 stages: NFT creation, sale, and retrieval. Using *ROS 2* for development, ensuring effective communication. To bridge the communication gap between robots (LEO2 and LEO3) and the blockchain, we integrated the PureStake connector via a *REST* service [1].

All communication flows, except for the smart contract, have been implemented, encompassing the autonomous creation, negotiation, coordination, selling, and verification of NFT and mapping requirements. Despite not taking full advantage of the smart contract automated behaviour, the prototype already leverages a blockchain network for trading NFT and certain data storage aspects, enabling real-time decision-making based on blockchain data. The architecture uses IPFS for high-throughput content-addressed block storage [147]. Entities can govern their mapping NFT metadata and settings, from transferability up to their destruction, promoting value to a market for data access and recognizing pioneering exploration. These NFT encapsulate specific map data and metadata, containing coordinates,

[1] `https://www.purestake.com/`

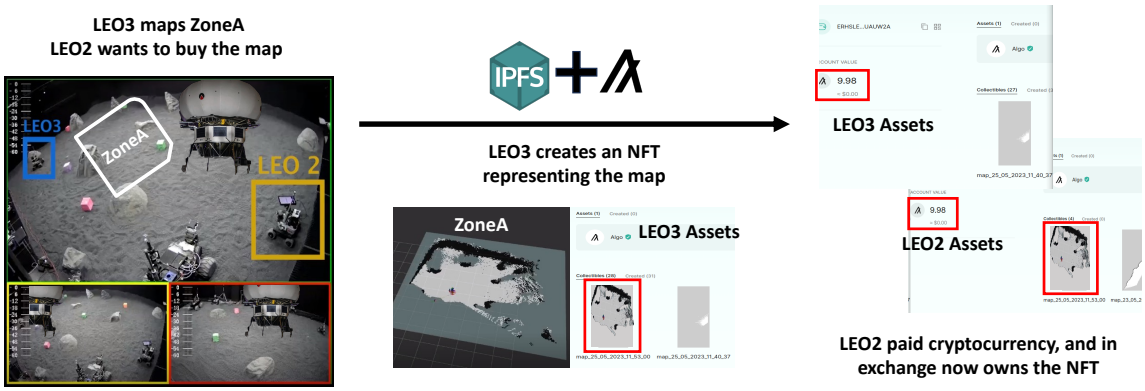


Figure 4.7: Prototype in the laboratory of simulated Moon environment.

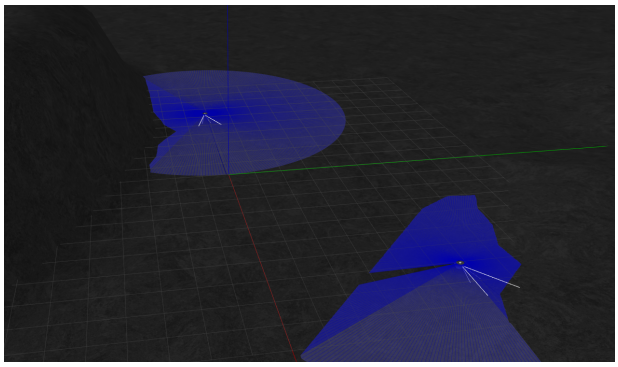
resolution, sensors, mapping algorithms, and map price, fostering trade within the network and improving ISRU's information symmetry.

The evaluation assessed the system's bandwidth usage and transaction duration in simulated lunar network conditions. Despite transactions needing to be fully optimised, the average bandwidth usage during uploads was less than **5 MB over 4 seconds, with a median of 4.7 MB.** The standard deviation was 0.5 MB, indicating a relatively consistent transaction performance. The time required for uploading maps and conducting sales transactions was approximately **10 seconds**, with a median of 8.7 seconds. The standard deviation was 2 seconds, indicating that most transactions were completed within a reasonable time frame. These durations are relatively short compared to the time robots typically take to navigate to the place and process cartographic data, which, at best, can take several minutes to hours. As a result, the upload duration, plus mapping and reaching a consensus, is negligible compared to considering the inherent latencies of bidirectional communication between terrestrial stations and robotic units (which, for the Moon case, takes a minimum of five seconds as the signal needs to pass through the different relays on Earth and satellites in space), let alone the possibility of other nations being open to start cooperating. The system displayed reasonable transaction duration usage and significantly reduced robot coordination delays compared to waiting for human coordination.

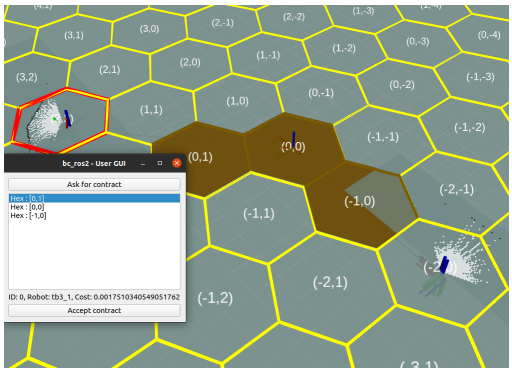
This experiment validated the use of DLT in a coopetitive space scenario, however it also highlited some of the potential limitations. Especially, battery consumption of DLT-based

Table 4.6: Non-proprietary Blockchain-Based Coordination Trade-offs in Analogue Lunar Conditions

Metric	Avg (Median)	Std. Dev.
Upload Bandwidth	∼4.7 MB	0.5 MB
Transaction Time	4 s	<1 s
Map Upload + NFT Creation	8.7 s	2 s



(a) View of the simulated lunar environment in Gazebo, featuring a lunar terrain with two robots and their laser scans.



(b) RViz2 user interface showing both robots on the hexagonal grid. Users can select hexagons (turning brown), request prices, and accept contracts.

Figure 4.8: Experimental simulation environment and user interface allowing task requests on a hexagonal mapping grid.

approach remains an open problem, as the energy consumption of these systems on earth is a well known constraint.

4.5.2 Make or buy for mapping evaluation

Experimental Setup

To evaluate our multi-robot exploration and coordination framework, we developed a simulated lunar environment using the Gazebo simulator. As illustrated in Fig. 4.8a, the environment spans an area of 100×100 meters and features elevation variations and crater-like terrain to emulate the physical characteristics of a real lunar surface. This environment is used to assess the robustness of autonomous navigation, mapping, and coordination strategies under realistic conditions.

Most of the crucial work for autonomous exploration has been covered by the work of Yamauchi [148] and related papers. The author highlights the need for three key components in any exploring robot: the mapping layer, responsible for discovering and representing the environment; the navigation layer, ensuring the robot can reach desired locations safely; and the exploration layer, which selects where to go next and sends these targets to the navigation layer.

In our experiment, we rely on a simple but reliable implementation of these three layers, as exploration is not the primary focus of this study:

Mapping layer The system uses two TurtleBot3 robots, each equipped with a 2D LiDAR sensor. The SLAM Toolbox in ROS 2 processes LiDAR data to generate a 2D occupancy grid map in real time. This allows each robot to construct and update a local representation of its surroundings, enabling decentralised, map-aware autonomy.

Navigation layer The ROS 2 Navigation Stack (Nav2) handles path planning, obstacle avoidance, and velocity control. For the purposes of the experiment, robots are configured with the standard Nav2 setup, which provides reliable autonomous movement without requiring additional customisation. The robots autonomously compute and follow collision-free paths to assigned targets based on their local SLAM maps.

Exploration layer The exploration logic is designed to ensure deterministic coverage of each hexagonal region. Each hexagon is subdivided into six triangles, and the robot's trajectory is defined by passing through the center of gravity of each triangle. This simple geometric strategy ensures that the robot explores the entire area with minimal computational overhead. While sufficient for the current simulated study, more robust strategies such as frontier-based exploration [148] are being considered for real-world deployments.

The lunar surface is discretised into hexagonal regions, inspired by a Goldberg polyhedral tiling, as described in Section 4.3.3. For simplicity and computational efficiency, the experiments focus on a limited subset of this grid. The selected area A satisfies $A \ll A_{\text{Moon}}$, allowing it to be locally approximated as a plane. Each hexagon has an edge length of

4 meters, which strikes a balance between resolution and computational load. This hexagonal tiling supports clear task partitioning and simplifies robot-to-region assignments (Fig. 4.8b).

The user interface, implemented in RViz 2, provides a visual overlay of the hexagonal grid and the robots' current positions and paths. As shown in Fig. 4.8b, the user can select any hexagon to request exploration. Selected hexagons are highlighted in brown. The system then queries each robot for estimated exploration costs, displays the responses, and allows the user to accept or reject the proposed contracts.

This simulation environment allows for reproducible evaluation of multi-robot exploration strategies, combining autonomous navigation, real-time mapping, and interactive task assignment under lunar-like conditions.

Experimental Scenario

We consider a scenario involving a single user and two autonomous robots, each owned by a different company. The objective is to compare the performance of our proposed task allocation framework against a baseline strategy where tasks are assigned randomly. This random allocation serves as a minimal benchmark, selected because existing task allocation solutions are either incompatible with our system assumptions or do not provide directly comparable results in our setting.

The evaluation is performed on a fixed set T of 7 hexagonal regions, selected to favour the baseline strategy by minimizing inter-hexagon distances. For each trial, a random subset $\tau \subset T$ is selected, and both methods (proposed and baseline) are tasked with mapping the same subset τ. The number of tasks (i.e. hexagons) in each subset is varied across trials to analyse scalability and behaviour under different workload conditions.

We assess the two approaches using the following metrics:

- **Total distance travelled** by all robots, serving as a proxy for energy consumption.
- **Total contracted cost**, computed using the system's internal cost model (Equation 4.1).
- **Actual operational cost** incurred by each robot during task execution.

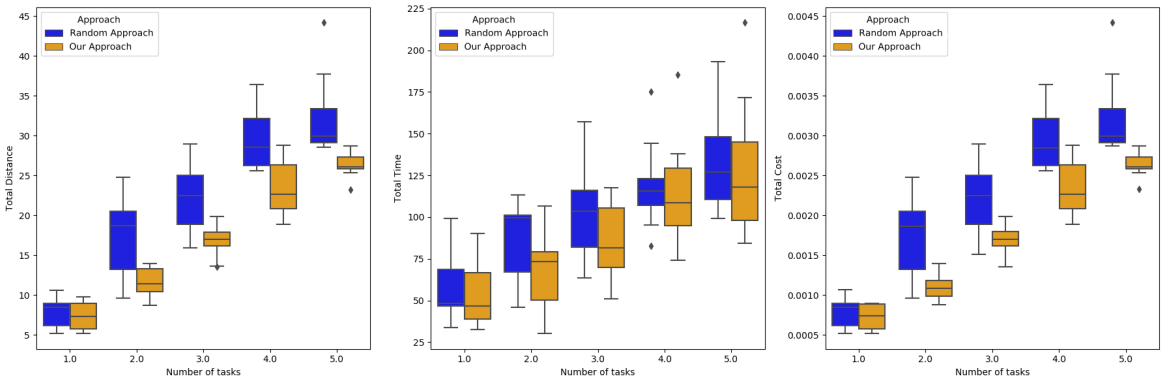


Figure 4.9: Comparison of key metrics across varying numbers of assigned tasks. The left plot shows the total distance travelled by the robots. The right plot shows the total time required to complete mapping tasks.

For each trial and subset τ, all metrics are recorded for both the proposed and baseline approaches, allowing direct comparison under identical task conditions.

Results and Discussion

As illustrated in Fig.4.9, the experimental results demonstrate that our proposed task allocation framework consistently outperforms the random allocation baseline. Specifically, it achieves an average reduction of 17.5% in total distance travelled across all trials. Since distance is directly linked to energy consumption, this indicates improved overall efficiency.

It is important to note that the experimental set T was selected to favour the baseline—by ensuring that tasks are closely grouped spatially—thus representing a best-case scenario for random allocation. In sparser or more realistic task distributions, the performance of the baseline degrades significantly, while our method remains robust. This reinforces the conclusion that even in disadvantageous conditions, our approach delivers superior results.

In scenarios where $\Delta t \ll L$, the cost function (Equation 4.1) is dominated by the travelled distance, which reflects the fact that locomotion constitutes the primary operational cost for mobile robots. This alignment between modeled cost and physical resource expenditure supports the validity of our approach.

Additionally, we compared the contracted cost estimated by the system to the actual operational cost measured during execution. On average, the contracted cost was found to

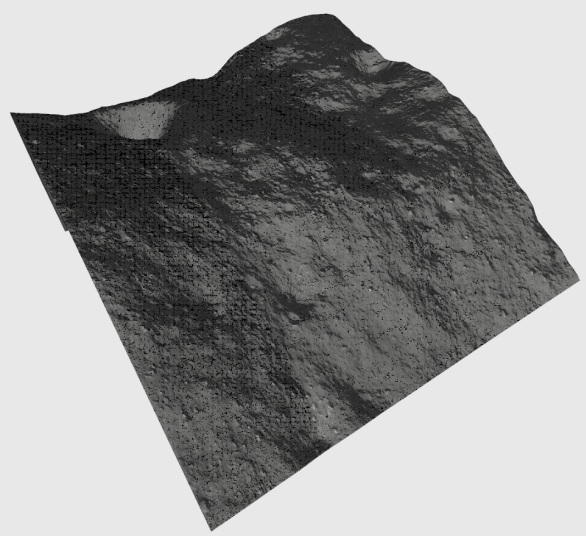


Figure 4.10: 3d model based on the site 1 from the NASA lunar digital elevation model database

be 2.18 times higher than the robot's true cost. This indicates that the pricing model ensures profitability for the robot owners, fulfilling a key economic requirement of the proposed market-based allocation mechanism.

4.5.3 Task-agnostic Make or buy framework evaluation

Experimental Setup

The experiment refers to the ISRU planetary exploration scenario depicted in Sec. 4.4.1. We implemented the proposed market-based architecture. Previous experiments detailed in 4.5.1 and 4.5.2 have validated the effectiveness of our coordination architecture and the performance of the proposed cost function.

In this experiment, we employ *Gazebo* to create a highly accurate 3D virtual lunar crater environment based on NASA's lunar Digital Elevation Model (DEM) [149] as visible in 4.10, and *ROS 2* to integrate autonomous economic decision-making logic within robot nodes. The lunar terrain was divided into hexagonal segments, each approximately 4 meters per side, ensuring a locally flat mapping scenario (Fig.4.5).

The navigation and exploration approach is the same for each robot as developed in

Section 4.5.2. The main difference of this new approach is that every robot features different characteristics, as stated in the Tab. 4.7.

External clients randomly generate task requests, prompting decentralised auction-based negotiations among robots. Robot heterogeneity is introduced through distinct robot datasheets (e.g. sensor accuracy, energy consumption profiles, velocities). Each robot evaluates and negotiates tasks autonomously using our proposed cost function, dynamically adapting to workload and internal resource constraints. This tests the auction system's dynamic efficiency in nondiscriminatory task distribution across robots.

The experiments evaluate metrics, including energy consumption, task execution time, traveled distances, task completion rates, system scalability, and adaptability to robot capability and task heterogeneity. Although fully blockchain-enabled smart contract auctions are conceptually detailed in the architecture, we implemented essential blockchain functionalities (NFT-based asset storage via IPFS) integrating *PureStake* connector via a *REST* service [2] to demonstrate feasibility and evaluate trade-offs. The robot operators carry out the creation, sale, and retrieval of NFT, while we employ *Piñata* and *PureStake* to interface with IPFS and the *Algorand* testnet, respectively.

Benchmark approaches

To quantify the contribution of our decentralised coordination framework and the economic decision-making approach, we compared performance metrics with two benchmarks:

- **Lower Bound (Random Allocation)**: Randomised task assignments, representing current non-*coopetitive* market conditions with minimal coordination.
- **Upper Bound (Centralised OR-Tools Optimisation)**: Tasks optimally allocated using the SOTA Google's OR optimiser solver [150] as a centralised alternative, providing a theoretical best-case scenario with both **Euclidean distance** (ORT_{euc}) and **our proposed market-based cost function** (ORT_{mb}).

[2] `https://www.purestake.com/`

Table 4.7: Robot parameters datasheets

Name	Size	Specification	Spawn (s)	E_d (W/m)	F_E ($/Ws)	Life (d)	P_u	Speed (m/s)	C_T ($)	HyperSpectral	$E_{picture}$ (W)	$t_{picture}$ (s)	$E_{processing}$ (W)	$t_{processing}$ (s)
a200_0001	Large	Producing energy is cheaper for it, but consumes more	0	2.4	0.000211	92	1.5	0.3	0.21	Yes	50	30	50	20
j100_0002	Medium	Fast, Expensive, Camera	0	1.2	0.0003	92	1.5	0.5	0.35	Yes	50	60	20	40
j100_0003	Medium	Fast, Cheap Light, process expensive	20	0.6	0.0001	92	1.5	0.5	0.35	No	0	0	30	60
j100_0004	Medium	Slow, cheap to move and process, no camera	40	0.1	0.0001	92	1.5	0.1	0.07	No	0	0	5	80
j100_0005	Medium	Slightly faster and more expensive, especially energy	60	1.2	0.0004	92	1.5	0.6	0.42	Yes	50	60	25	40
j100_0006	Medium	Slow and cheap low power robot	80	0.6	0.0001	92	1.5	0.2	0.14	Yes	55	45	25	120
j100_0007	Medium	Slow and cheap low power robot	80	0.6	0.0001	92	1.5	0.2	0.14	Yes	55	45	25	120
j100_0008	Medium	Slow and cheap low power robot	80	0.6	0.0001	92	1.5	0.2	0.14	Yes	55	45	25	120
j100_0009	Medium	Slow and cheap low power robot	80	0.6	0.0001	92	1.5	0.2	0.14	Yes	55	45	25	120

Having ORT_{euc} and ORT_{mb} allows us to compare the efficiency of our cost function in a SOTA approach compared to the baseline approach.

Experimental Protocol

While in the experiment in Section 4.5.2, a fixed set of 7 hexagonal map segments (T) was defined, and each experimental run randomly selected subsets ($\tau \subseteq T$) for consistent comparative evaluation among two robots.

In this new simulation, 6 heterogeneous robots (representing MRS owned by different providers, as detailed in Tab. 4.7) execute tasks (Exploration, Imaging, and Imagery Processing) autonomously and concurrently, while 3 SO can order tasks in parallel. To assess the system's inclusiveness and scalability while running, two of those MRS are already present at the start of an experimental run, while the remaining four appear every twenty seconds at fixed positions.

In our setup, one *Experimental Run*, hereafter referred to as a *Scenario*, corresponds to the execution of a single task file. Each task file contains a random sequence of five tasks, selected from two categories: **Exploration** and **Imaging**. To mimic realistic decision intervals, a uniformly distributed delay between $20\,\mathrm{s}$ and $60\,\mathrm{s}$ is inserted between successive tasks. The choice of five tasks reflects the average number of tasks that can be completed within the fixed ten-minute duration of an *Experimental Run*.

Every *Scenario* is executed **ten times** for each coordination approach under study (Proposed $MB(our)$; ORT_{euc}; ORT_{mb}; Random allocation).

In total, 120 distinct *Scenarios* are generated, resulting in a high number of *Experimental Runs* across all approaches. As each *Scenario* is repeated for four approaches and ten iterations, the complete protocol results in $120 \times 4 \times 10 = 4800$ *Experimental Runs*.

Due to resource constraints, the blockchain-based coordination mechanism is partially simulated: a dedicated *ROS 2* client node emulates the smart contract's logic, following the framework introduced in [151].

Performance Metrics

To evaluate the efficiency of our approach against the baseline, we define a set of key performance metrics. These are divided into *measured metrics*, directly collected during experiments, and *derived metrics*, obtained through post-processing.

Measured Metrics The following metrics are intended to represent the measured impact on resource usage:

- **Computed Cost**: For *experimental runs* including our $MB(our)$ approach, this corresponds to the optimal cost (C_i) computed per task.
- **Execution Time**: The time taken by a robot to complete the tasks assigned.
- **Cumulative Distance**: The total distance traveled by a robot while performing its assigned tasks.
- **Tasks Assigned**: The number of tasks assigned to each robot during an *experimental run*.
- **Distance to Task**: The Euclidean distance between a robot and its assigned task at allocation time.

Derived Metrics From the above raw measurements, we compute the following indicators of overall performance:

- **Operational Cost**: Calculated with **Execution Time** and **Cumulative Distance** in Eq. 4.1, this indicates the actual resource usage cost (C_i) during task execution, which may include deviations from the planned schedule.
- **Normalized Task Allocation**: The number of tasks assigned to a robot, normalized by its operational uptime. This quantifies individual contributions and measures load balancing across the team.

- **Task-to-Distance Ratio**: The ratio of the average number of tasks assigned divided by the average **Distance to Task**. This indicates how efficiently the approach assigns tasks spatially close to the robots.
- **Processing Reallocation**: The ratio between the number of exploration tasks reassigned and the number of other tasks assigned, indicating the reallocation of computational processing to other robots. This captures the additional effort introduced by task reallocation.
- $MB(our)$ **Ratio Deviation**: Measures the deviation of each alternative approach from the MB_{ours} baseline. Quantifies how the performance ratios observed by other approaches differ from the theoretical ratios expected under MB_{ours}, serving as an indicator of relative efficiency and consistency.

All metrics are reported both per robot and per task type, aggregated using mean and variance, and displayed with 95% confidence intervals (bootstrapped) when applicable.

Result and Discussion

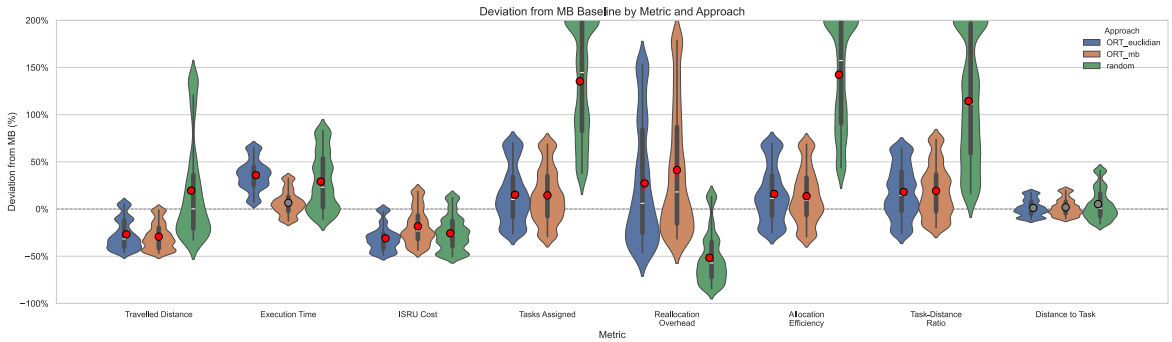


Figure 4.11: Market-Based Coordination Ratio Comparison across allocation strategies and metrics

The Tab 4.8 and Fig. 4.11 summarizes the relative performance of our decentralized improved $MB(our)$ system against other coordination strategies. Both $MB(our)$ approaches significantly reduced the total operational cost (up to 28.5%) compared to centralized and random baselines. Suggesting cheaper bidders more often secure contracts, closely reflecting

Table 4.8: Average Performance Ratios Comparing Alternative Approaches to the $MB(our)$ Baseline

Approach	ORT_{euc}		ORT_{mb}		Random	
Metric	Dev. (%)	Var. (±)	Dev. (%)	Var. (±)	Dev. (%)	Var. (±)
Operational Cost	-29.7	0.88	-16.9	1.36	-24.7	0.62
Cumulative Distance	-16.0	45.10	-29.1	0.35	189.5	8190.34
Distance to Task	1.4	0.27	1.7	0.27	9.4	0.44
Normalized Allocation	16.8	1.41	15.1	1.38	181.4	8.70
Tasks/Distance Ratio	19.8	2.02	20.2	1.94	139.4	9.53
Tasks Assigned	15.9	1.48	15.9	1.46	166.4	8.19
Execution Time	36.8	1.10	7.7	0.84	29.9	1.02
Processing Reallocation	32.4	20.87	44.9	22.53	-49.8	4.36

human behavior in agency economic decision-making. It also highlights that our custom cost function outperforms the Euclidean one in a centralized approach in terms of operational cost and cumulative distance.

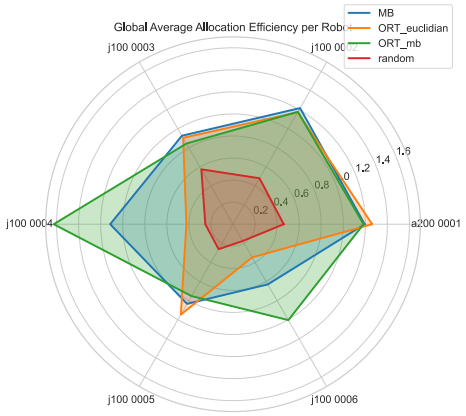
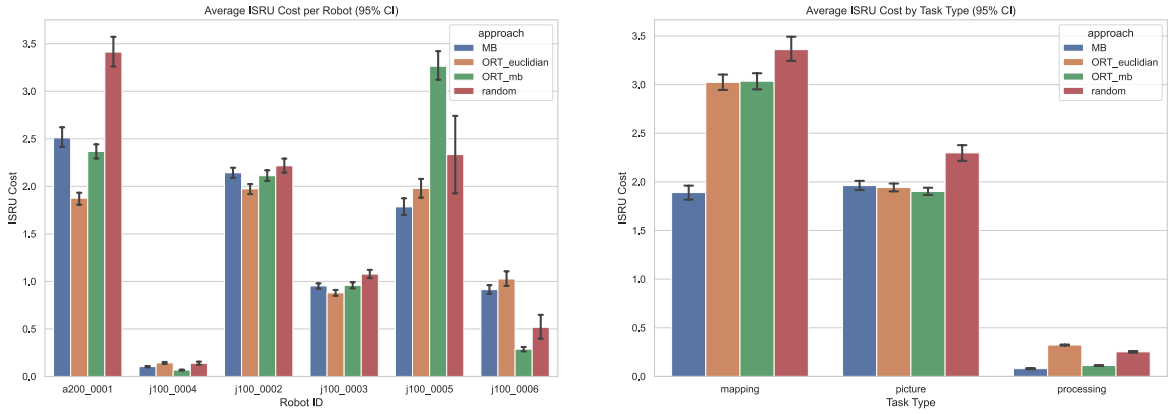


Figure 4.12: Average normalized number of tasks assigned per robot and approach

Fig. 4.12 depicts how different coordination approaches distribute tasks between robots during their available time (from timestamps), allowing fair comparisons between robots entering the simulation at different times. The results in Fig. 4.12 highlight the distinct behavior of the different coordination strategies. The random baseline consistently assigned fewer tasks compared to all other approaches. Among the centralized methods, the ORT_{mb} strategy, using our proposed cost function, achieved the highest average number of normalized task

assignments. However, both centralized strategies tended to concentrate assignments on a single robot (e.g., robot `j100_0004` under ORT_{mb} and robot `j100_0005` under $ORT_{euclidean}$). In contrast, the distributed approach $MB(our)$ yielded a more balanced allocation across the robot team. From a *coopetitive* perspective, where each agent seeks to ensure participation, such a balance is essential, as it improves fairness and benefits all robots involved.



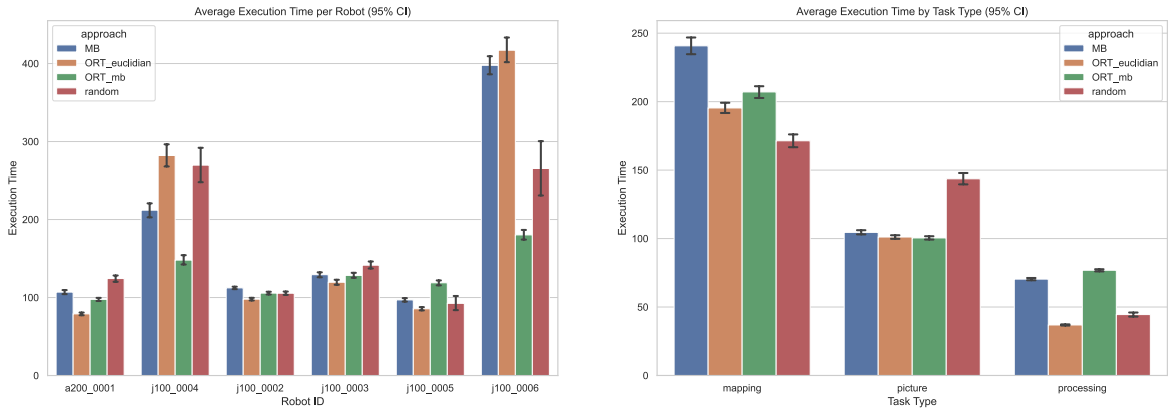
(a) Operational Cost per Robot

(b) Operational Cost per Task Type

Figure 4.13: Comparison of operational cost: **per robot** vs. **per task type**.

The approaches were also compared through the **operational cost** metrics (Fig. 4.13), derived using our improved cost function (Section 4.4.2) on the actual time and distance measured for each robot. These combined results suggest MB's improved balance between efficiency and adaptability of $\sim 28.5\%$ compared to ORT_{euc} and approximately $\sim 23.5\%$ compared to random task assignments. Robotic travel strongly influences this metric, reduced by up to 32%, confirming the significant improvement in task allocation efficiency (up to 85% more per robot) and computation processing outsourcing ($10 \sim 58\%$). This data highlights that our solution is more cost-effective than others for completing the tasks.

Fig. 4.14 points to slightly lower performance in $MB(our)$ execution time than ORT_{euc} ($+35\%$). However, MB's task specialization per distance mitigated this trade-off (up to $+75\%$), suggesting a better allocation even when the most efficient candidates were unavailable. The results attest to how $MB(our)$ approaches allowed MRS to dynamically evaluate their suitability for each task, posing continuous lower bids and attaching multiple codependent



(a) Task Execution Time per Robot

(b) Task Execution Time per Task Type

Figure 4.14: Comparison of task execution time: **per robot** vs. **per task type**.

contracts simultaneously. Furthermore, $MB(our)$ had a minimal variance in the distance to the task assigned between approaches ($\pm 0.3\%$), suggesting further potential inefficiencies or missed opportunities.

The results validate the central hypothesis that market-based decentralised coordination enables more rational, resilient, and efficient multi-robot operations in ISRU scenarios. Robots under MB coordination autonomously engage in cost-aware task selection, reflecting human beings' economic behaviour, such as outsourcing and task specialisation. This contrasts with static centralised optimisation, which leverages predefined allocations rather than run-time adaptability.

We observed the emergence of division of labour among robots without requiring predefined roles, purely driven by economic incentives embedded in the cost function. Higher-speed or lower-cost robots naturally assumed more mobile tasks, while energy-intensive computation (e.g. blockchain upload and transfers) was distributed to specialised agents so they could quickly become available. This is attributed to optional task assignments (not mandatory) and the term of the cost function LU that discourages the overuse of specific agents, preserving system health and the mission's lifespan.

Notably, although smart contracts are simulated, this prototype validates DLT's feasibility to automate robotic economies, eliminating the need for human input. Entities could manage

their mapping NFT, covering aspects from transferability to destruction, thus enhancing the value of data access markets and recognizing pioneering exploration. However, network consensus dependency could hinder coordination under node disconnection.

Despite observable benefits, market-based coordination risks short-term economic inefficiencies by overlooking future logistic planning. Additionally, a universal and transparent cost/reward function may be vulnerable to malicious manipulations (e.g. collusion or underbidding). While our collateral-taking helps mitigate risks, future studies could explore game-theoretic safeguards or reputation-based trust metrics. Although company cost functions will likely be proprietary algorithms, alternative performance-based reward mechanisms could provide additional robustness [28].

4.6 Discussion

RQ-1: *How do robots collaborate autonomously with trust in a coopetitive system?*

This chapter set out to explore how autonomous robots can collaborate in a decentralised environment while maintaining trust in each other's actions and in the coordination process. Addressing this question required designing both a mechanism for autonomous decision-making and an infrastructure that ensures transparency, accountability, and robustness against failures or misbehaviour.

To answer RQ1, we proposed a coordination architecture grounded in DLT principles and auction-based task allocation. In this system, trust is not externally imposed, but emerges from the properties of the protocol: transactions are transparent, decisions are auditable, and each agent operates based on local economic reasoning. Robots evaluate the cost of executing a task based on internal datasheet parameters, bid autonomously, and reach agreement through market-based interactions.

We evaluated this system in two main phases, each designed to progressively increase complexity and realism:

Phase 1: Homogeneous Robots and Mapping Tasks

In the initial experiments, we assessed whether decentralised negotiation alone could lead to efficient collaboration. Robots had similar capabilities, and tasks involved local mapping of lunar hexagons. Using a custom cost function based on energy consumption and travel distance, robots autonomously submitted bids for tasks and contracts were awarded accordingly. Results showed a clear improvement in system performance: the total distance travelled was reduced by over 17.5% compared to a random baseline. This validated the hypothesis that simple local reasoning, when coordinated through an open negotiation protocol, can produce globally beneficial behaviour.

Phase 2: Heterogeneous Robots, Tasks, and Extended Cost Model

To better reflect realistic exploration scenarios, we introduced heterogeneity among both robots and tasks. Each robot had a different operational profile: energy cost, speed, sensor capabilities, and spawn time varied according to datasheet parameters (Table 4.7). Tasks also became more diverse, including stationary actions (e.g. imaging, processing) with execution time and energy constraints. This required extending the cost function to include time penalties, uncertainty propagation, and wear-based depreciation.

By integrating this extended model, we observed new behaviours emerging from the system:

- **Self-specialisation:** Robots tended to favour tasks aligned with their capabilities, without requiring explicit role assignment. For example, low-speed robots opted out of distant mapping tasks, and high-sensor-power robots concentrated on imaging.
- **Load balancing:** The auction mechanism naturally spread tasks across the available agents. No robot was overloaded unless supply was constrained, and idle robots opportunistically took over newly available work.
- **Scalability:** The system maintained coordination quality with up to nine simultaneously operating agents, despite increased complexity in robot profiles and task types.

These results provide a strong affirmative answer to RQ1. Collaboration emerged not from predefined schedules or global planning, but from local, economically motivated decisions executed within a transparent negotiation framework. Trust, in this context, is realised through:

- **Verifiability:** Every transaction (bid, contract, execution) can be traced, audited, and evaluated independently by any party, reducing ambiguity and miscoordination.
- **Autonomy:** Each robot maintains control over its own decisions, accepting or rejecting tasks based on internal constraints and preferences.
- **Fairness:** The auction protocol ensures that opportunities are not monopolised, and that competition reflects actual capability and availability.

While our implementation did not include adversarial agents, failure modes, or full consensus mechanisms, it lays the foundation for systems in which autonomous trust is not assumed but systematically built into the architecture. Future work could extend this framework to include reputation tracking, economic penalties, or cryptographic proofs of work completion.

Limitations and Open Challenges

Despite these promising results, several limitations constrain the scope and applicability of this study:

- **No adversarial behaviour.** All agents were assumed to be honest and cooperative. Real coopetitive environments may involve misreporting, free-riding, or malicious denial of service.
- **Simplified consensus.** Auctions relied on direct broadcasting without modelling delays, forks, or Byzantine failures. Incorporating realistic consensus (e.g. PBFT, PoS) would affect scalability and latency.
- **Restricted heterogeneity.** Robot profiles varied along a few datasheet parameters (energy, speed, sensors). Real lunar rovers differ in mobility modes, computation, payload, and maintenance constraints.
- **No resource pricing dynamics.** Costs were fixed functions of energy/time. More advanced markets would include dynamic pricing, scarcity, or strategic bidding.

- **Limited scale.** Experiments tested up to nine robots. Industrial deployments would require coordination at higher scales, raising questions of performance and overhead.

These constraints do not undermine the value of the results but highlight where further work is required. In particular, robustness to adversarial behaviour and integration with real-world consensus protocols remain open challenges for future research.

4.7 Summary

In this chapter, we investigated how autonomous robots can collaborate with trust in decentralised planetary exploration scenarios. To address **RQ-1**, we designed a coordination framework combining blockchain-inspired protocols and a market-based task allocation mechanism grounded in the make-or-buy principle.

We developed a cost model that allows robots to evaluate and bid for tasks based on energy use, time constraints, and uncertainty. This model was validated in experiments of increasing complexity, from uniform mapping tasks to heterogeneous robot capabilities and task types.

Our findings demonstrate that:

- Economic negotiation enables efficient and fair task distribution without centralised control.
- Trust can emerge from the system design itself, through transparency, autonomy, and auditable decision processes.
- The framework scales to multiple agents and adapts dynamically to robot heterogeneity and resource constraints.

This work contributes to the vision of distributed machine economies in space, where intelligent agents self-coordinate exploration and resource use through rational, transparent, and trustworthy interactions.

Contributions:

1. **We demonstrate the feasibility of blockchain-based map exchange in a real MRS deployed in a lunar analogue environment.** This proof-of-concept validates the integration of decentralised data ownership, trading, and validation mechanisms within physical robotic platforms operating under Moon-like conditions.

2. **We propose a decentralised coordination framework for trusted multi-robot mapping in space exploration.** Our architecture enables autonomous agents to negotiate task ownership transparently using distributed ledger principles and market-based mechanisms.

3. **We extend the coordination framework to be task-agnostic and scalable to heterogeneous robots and missions.** The enhanced system accommodates diverse task types—such as mapping, imaging, and data processing—allowing robots with varied capabilities to collaborate effectively.

4. **We validate the approach in a high-fidelity simulated lunar environment with dynamic agent participation.** Experiments demonstrate the system's efficiency, adaptability, and economic rationality under realistic terrain and resource constraints, showing performance near that of centralised optimisation.

Chapter 5

Efficient communication for heterogeneous Multi-Robot systems

Related Publications

- Chovet, L.[†], Garcia, G.[†], Bera, A., Richard, A., Yoshida, K., Olivares-Mendez, M. A. **"Performance Comparison of ROS2 Middlewares for Multi-Robot Mesh Networks in Planetary Exploration"**, *Journal of Intelligent and Robotic Systems*, 2024. (Accepted/In Press]) `https://hdl.handle.net/10993/64125`

[†] These authors contributed equally to this work.

"The future is already here — it's just not very evenly distributed."

– William Gibson

5.1 Introduction

This chapter covers the following research question:

RQ-2: *How efficiently can heterogeneous Multi-Robot Systems communicate for space missions?*

In order to ensure efficient *coopetitive* systems as studied in chapter 4, the overall

MRS must establish the most efficient communication between all the agents. Connectivity maintenance is the most crucial aspect of a space system, losing the connection to a robot is not acceptable since it couldn't be recovered [92]. In a *coopetitive* system, many companies are proposing robotic platforms able to perform several tasks, however interoperability is a need. The Open Robotics foundation proposes this through the ROS 2, the newest version of ROS, focusing on aspects such as MRS and scalability [152]. ROS 2 is characterised by enhanced security features, real-time capabilities, and a flexible, dynamic architecture, allowing robots to interact in complex and dynamic environments. Most of the network features are brought by the use of the DDS technology, acting as middleware between ROS 2 and the network layer.

DDS is a communication protocol often used in robotics [153]. DDS plays the role of a messenger that allows different parts of a system, such as robots in a MRS, to talk to each other effectively through mechanisms of subscription and publication. DDS can handle large amounts of data, deal with complex communication patterns, and is designed to work well even in challenging environments. Therefore, DDS become pivotal in MRS because it ensures that all actors of a system are connected, updated, and working together smoothly [153]. However, alternatives to DDS are emerging, such as Zenoh which promises better performances[154].

Along with the OSI model, ROS 2 and the middleware rely on the network layer to work. If many network architectures exist, exploration requires some specificity, such as scalability and the ability to react to topology changes. In traditional star topology of wireless networks, each device connects to a central node. The mesh networks allow each device, or "node", to connect directly to several others. This configuration enables data to hop from node to node until it reaches its destination, enhancing the robustness of the network by offering multiple pathways for data transmission. Consequently, mesh networks are more reliable, as the failure of a single node rarely leads to a breakdown of the entire network. In addition, mesh networks are highly scalable, allowing the integration of more nodes without significant degradation in network performance [155]. This feature is critical in MRS where the number of robots may vary depending on the task at hand.

This chapter investigates the impact of using mesh network and ROS 2 alliance for MRS-driven mission in extreme environments. We first introduce the implementation of the mesh network technology in the REALMS2 system (cf. Chapter 3.3) in Section 5.2. We then highlight the current research gap in the use of the DDS technology for MRS in Section 5.3. In order to address this research gap, we establish a clear scenario and a realistic network architecture, detailed in section 5.4. We answer the research gap in two experiments in the section 5.5 and finally we conclude on the research question in Section 5.6.

5.2 Setting the appropriate network for a space mission in the Space Resource Challenge

As detailed in section 3.3.2, we had to set up a mesh network for the REALMS2 system. Although there are many approaches to mesh networking, we applied the HWMP+ protocol. As developed in the section 2.6, HWMP+ is the approach most suited to our implementation. Especially since the Mikrotik Groove A52ac router, the one equipped on the REALMS2 rovers, is designed to work optimally with this technology.

The process of setting up the mesh network represented an important part of this project, as the right settings would influence the results of the challenge.

After various trials and errors, along with documentation reading, the process to set a Mikrotik Groove antenna as a Mesh Network node can be summed up as the following lines of code:

Listing 5.1: MikroTik Groove Mesh Configuration

```
/interface mesh
add mesh-portal=yes name=mesh1
/interface wireless
set [ find default-name=wlan1 ] band=2ghz-b/g/n country=luxembourg disabled=no \
    frequency=2412 installation=outdoor mode=ap-bridge ssid=leo_mesh \
    station-roaming=enabled wds-default-bridge=mesh1 wds-mode=dynamic-mesh
/interface ethernet
set [ find default-name=ether1 ] speed=100Mbps
```

```
/interface lte apn
set [ find default=yes ] ip-type=ipv4-ipv6
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip firewall connection tracking
set udp-timeout=30s
/ip settings
set max-neighbor-entries=2048
/interface mesh port
add interface=wlan1 mesh=mesh1
add interface=ether1 mesh=mesh1
/interface ovpn-server server
set auth=sha1,md5
/ip address
add address=192.168.44.1X0/24 interface=mesh1 network=192.168.44.0
```

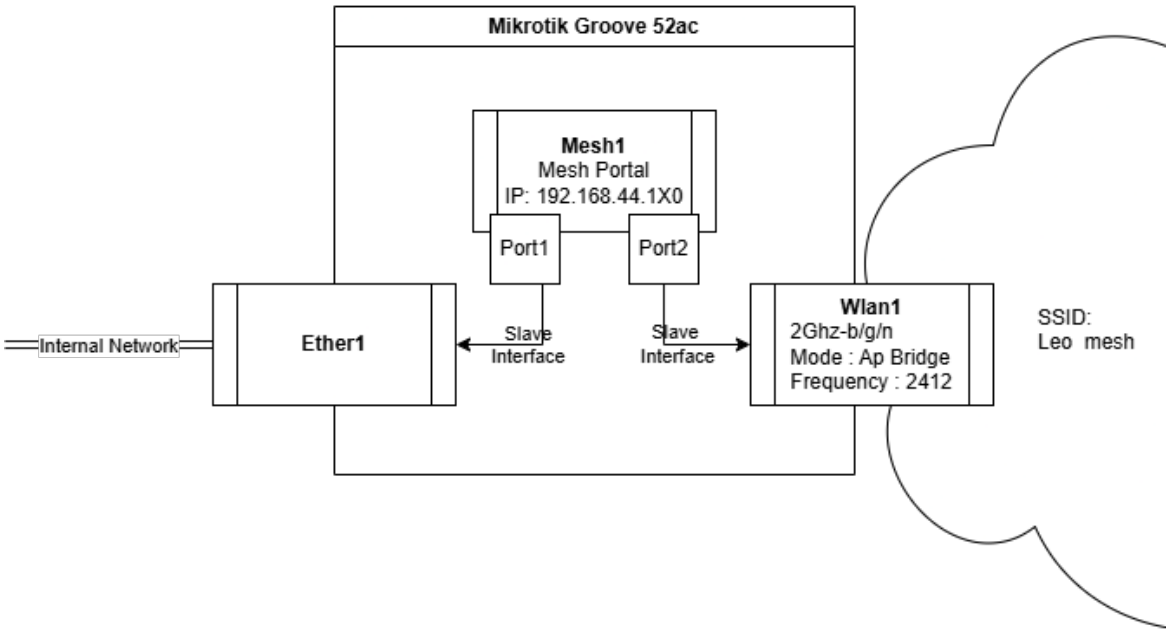


Figure 5.1: Graphical representation of the Mikrotik Groove interfaces set for the mesh network

When applied to a Mikrotik Groove router, these settings correspond to the visual representation of Fig.5.1. As visible in the configuration, the MikroTik Groove device includes **three network interfaces**: two physical interfaces (`ether1` and `wlan1`) and one virtual interface (`mesh1`). Their roles and configurations are as follows:

- **ether1** — *Ethernet interface*:
  This is the physical Ethernet port of the router. Its role is to connect the router to the rest of the robot's internal network. No additional configuration is required at this level, as the interface is fully managed by the virtual mesh interface (`mesh1`).

- **wlan1** — *Wireless interface*:
  This interface enables wireless communication. Its parameters are configured to optimise performance based on the deployment environment:

  - **Band**: The frequency band determines the communication range and throughput. In outdoor scenarios, the 2.4 GHz band is preferred for its longer range and better penetration, whereas 5 GHz may be chosen in indoor or high-density environments for greater bandwidth and reduced interference.
  - **Frequency**: The specific channel (e.g., 2412 MHz) is selected using the router's built-in frequency scan tool. This ensures the use of a less congested frequency to improve reliability and reduce interference.
  - **Mode**: The `ap-bridge` mode enables the interface to participate as an access point within the mesh. Combined with the `dynamic-mesh` WDS mode, it allows `wlan1` to be seamlessly integrated into the mesh topology managed by `mesh1`.

- **mesh1** — *Virtual mesh interface*:
  This logical interface manages the mesh network using the HWMP+ routing protocol. It aggregates both `wlan1` and `ether1` into a unified mesh node. A single IP address is assigned to `mesh1`, simplifying network configuration and routing across both physical interfaces.

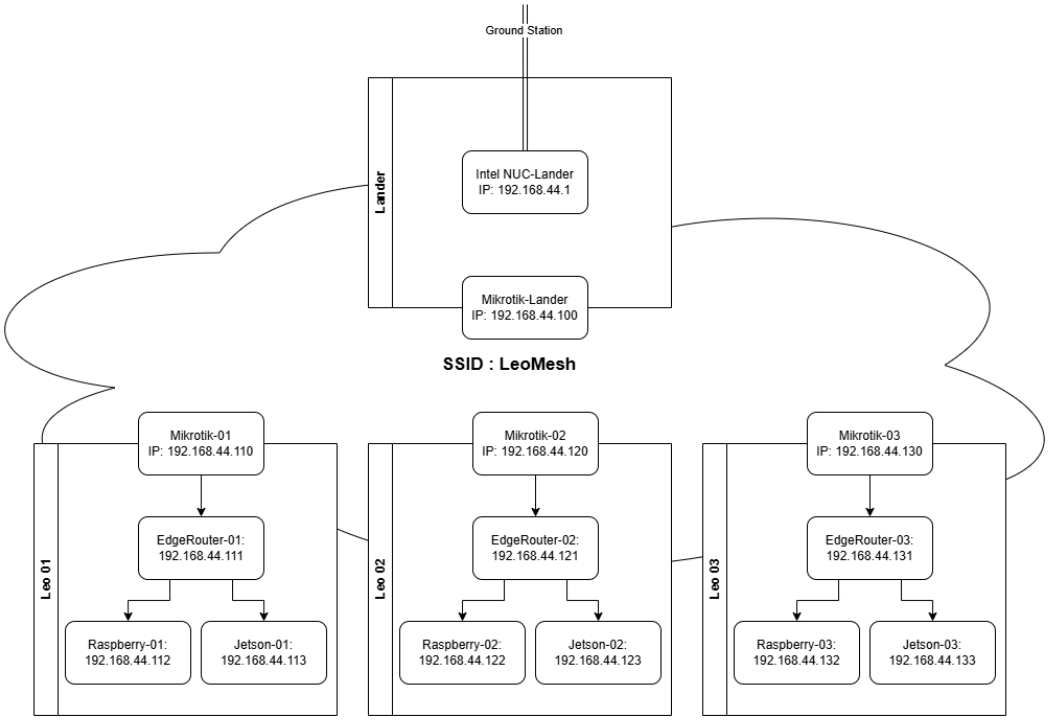


Figure 5.2: Graphical representation of the network of REALMS2

The choice of the IP addresses was a design choice decided for easier access to all the members of the system. The REALMS2 network architecture is visible in Fig. 5.2. Within the subnet `168.182.44.0/24`, we adopted a structured IP addressing scheme based on the last octet (i.e., the host identifier). The format used is `168.182.44.ABC`, where the digits `A`, `B`, and `C` encode the role, robot number, and hardware type, respectively.

- **A** – Entity type:
  - `0`: Lander-related components
  - `1`: Rover-related components
  - `2`: Debugging or temporary devices
- **B** – Unit ID:
  - `1`, `2`, `3`, etc., identifying the rover or lander index
- **C** – Device type:

- `0`: Antenna
- `1`: Edge router
- `2`: Raspberry Pi
- `3`: Jetson computer

For example, `168.182.44.120` would correspond to a rover (`A=1`), unit 2 (`B=2`), and its Raspberry Pi (`C=0`).

As detailed in 3.3.3 this specific setting allow for a range of **220 meters** between two nodes, while adding a relay increases the bandwidth capacity from 2 to 10 times.

5.2.1 Takaway from the use of the Mesh Network in the REALMS2

During the second trial of the ESA-ESRIC Space Resources Challenge (see section 3.3.4), the use of a mesh network in combination with ROS 2 proved essential for ensuring resilient communication in a lunar-analogue environment lacking infrastructure. The architecture allowed each rover to act as a dynamic relay node, supporting a decentralised and redundant communication system that automatically adapted to node failures or changing topology.

Key advantages observed:

- **Extended coverage**: By leveraging mesh networking, a single operator could maintain communication with robots beyond direct line-of-sight, achieving up to $220\,\mathrm{m}$ of range in a constrained test environment.
- **Resilience**: The system dynamically rerouted data through neighbouring robots, ensuring mission continuity despite simulated blackouts and individual node failure.
- **Modular scalability**: Additional rovers could be seamlessly integrated into the mesh without reconfiguring the network.

Issues encountered:

- **ROS 2 DDS overhead**: The DDS middleware used by ROS 2 introduced substantial data overhead. While DDS enables QOS guarantees suitable for space-grade communication, it resulted in higher bandwidth usage than anticipated, particularly under constrained mesh conditions.
- **Limited real-time performance**: DDS's reliability mechanisms and QOS layers—though beneficial for resilience—introduced communication latency and jitter when packet loss occurred, especially in multi-hop transmissions. This affected the real-time control of robots during teleoperation.
- **Increased compute load**: DDS serialisation and QOS enforcement consumed considerable computational resources on the Jetson Xavier, particularly during multi-robot operation with frequent large message publication (e.g., map data, point clouds).

These identified communication limitations, particularly those stemming from DDS middleware in ROS 2,such as excessive data overhead, inconsistent reachability in dynamic topologies, and high CPU consumption, highlighted the need for further investigation. As a direct outcome of these observations during the ESA-ESRIC Space Resources Challenge, we initiated a dedicated study to benchmark alternative ROS 2 middleware implementations for mesh networks in planetary exploration scenarios. The work presented in the following sections evaluates FastDDS, CycloneDDS, and Zenoh under realistic conditions and dynamic topologies.

5.3 Research gap

As highlighted in the state of the art, MRS is a widely studied topic due to its promise in enabling resilient, flexible, and scalable robotic systems for complex missions. While substantial work has been conducted on task allocation, control architectures, and cooperative behaviours, most of this research assumes ideal network conditions, often overlooking the role of communication infrastructure in real-world deployments.

With the adoption of ROS 2, the robotics community gained access to a more modular and distributed middleware that facilitates decentralised system architectures. However, ROS 2's reliance on the DDS standard introduces new challenges, including increased data overhead, complex QOS configurations, and limited resilience under constrained or unstable network conditions. These issues are exacerbated in dynamic or large-scale MRSs, where communication bottlenecks and inconsistent connectivity can critically affect performance, decision-making, and mission success.

Our experience with the *REALMS2* system during the ESA-ESRIC Space Resources Challenge revealed that ROS 2-based communication over a mesh network is far from trivial. Although the mesh topology enabled extended communication range and fault tolerance, several limitations were observed:

- ROS 2 exhibited significant message overhead and bandwidth usage, leading to degraded performance in high-load scenarios.
- DDS discovery mechanisms caused unnecessary traffic, particularly during node initialisation or network topology changes.
- Network jitter and delays during relay-based communication routes impacted task execution reliability and synchronisation.

Despite these challenges, the mesh network allowed the system to maintain partial operability during signal blackouts and node failures—highlighting its potential as a backbone for decentralised planetary exploration.

However, there remains a critical lack of systematic studies assessing ROS 2's behaviour over mesh networks under realistic constraints. No prior work has quantitatively compared different middleware options (e.g., FastDDS, CycloneDDS, Zenoh) in terms of reachability, delay, jitter, CPU/RAM usage, and network overhead on mesh topologies. Furthermore, the community lacks standardised guidelines for designing communication architectures for MRS operating in constrained, ad hoc, or large-scale environments.

This gap motivates the need for dedicated research that bridges middleware-level communication and network-layer behaviour, focusing on practical, scalable, and energy-efficient

solutions for ROS 2-based MRSs. The follow-up study builds directly on these findings, evaluating the interplay between ROS 2 middleware and mesh networks in planetary exploration scenarios, with the intention of informing the design of next-generation resilient robotic systems.

5.4 Scenario and System Overview

To address the research gap outlined in the previous section—namely the lack of communication architectures validated for heterogeneous, decentralised MRS in mesh-networked, bandwidth-constrained environments—we present a system design tailored for lunar ISRU missions along with a comprehensive scenario of highlighting the potential problem of this approach. This section introduces a decentralised, robot-agnostic architecture enabling autonomous decision-making via a market-based coordination strategy, fully integrated with a resilient ROS 2 and mesh network infrastructure.

5.4.1 Network Architecture

To meet the needs of scalable, resilient communication in decentralised MRS, we propose a mesh network architecture based on the HWMP+.

We consider a heterogeneous MRS deployed in an outdoor area to perform a set of tasks. A set $R = \{r_1, r_2, r_3, ..., r_n\}$ consists of n different robots connected through a mesh network M and all participating in it. For $i \in \{1, n\}$, a robot r_i is composed of $r_i = \{c_i, s_i, a_i\}$ with c_i a set of m embedded computers $c_i = \{c_{i_1}, ..., c_{i_m}\}$, a network switch s_i, and a mesh antenna a_i. The embedded computers represent each device connected to the mesh that is directly on a robot. The focus of this system is to propose high adaptability to all robotic platforms by connecting all the embedded computers to the switch, which is itself connected to the mesh router. We also consider a set of l operators and each operator is represented as $o_k = \{c_k, s_k, a_k\}$, where $k \in \{1, l\}$, c_k denotes associated computer, s_k refers the associated network switch, and a_k the respective antenna. In that case, the mesh network comprises n robots and one operator o_k is represented by set $M = \{a_1, a_2, ..., a_n, a_k\}$. Fig.5.3 presents

the proposed architecture.

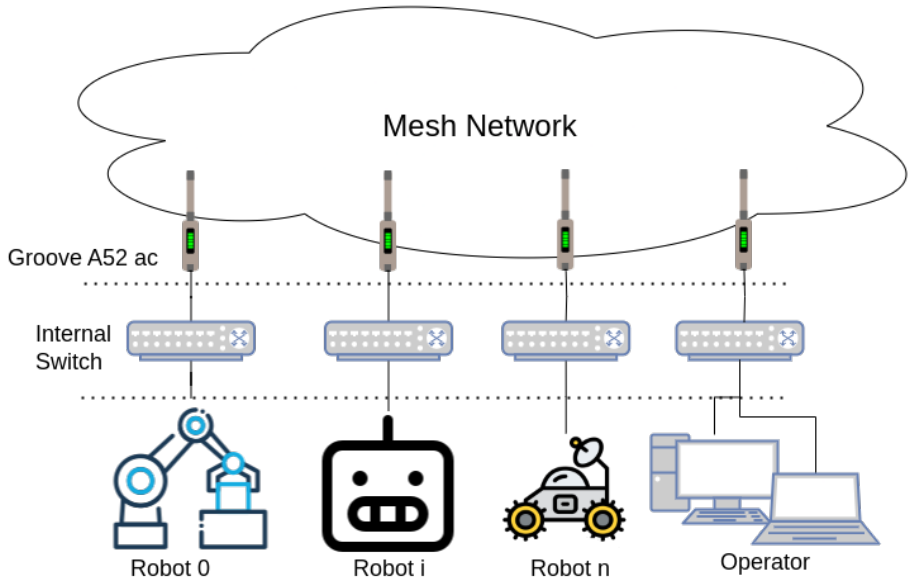


Figure 5.3: Mesh Network architecture

We consider that all robots have various capabilities, such as movement, data collection, and data processing. They also have to face real-time mechanics. As explained in the state of the art, HWMP+ is designed with large dynamic systems in mind. According to [82], HWMP+ has better metric computation, which results in less packet loss, reduced delay, and higher throughput compared to the standard HWMP. This is particularly noticeable in dynamic environments. Among the different mesh protocols, HWMP+ presents the fitting properties for the architecture proposed here, which will be the mesh network used.

5.4.2 ROS 2 Architecture

Every embedded computer c_{i_j} for $j \in \{1, m\}$ operates on a specific set of ROS 2 nodes that run the chosen RMW implementation. In its latest release, ROS 2 provides native support to two free DDS implementations, which are listed below:

- **FASTDDS**: Also called FASTRTPS, this DDS has been used with ROS 2 since its beginning, always proposed as the default version. Since the implementation described

in [37], there have been significant changes in its implementation, such as using shared memory transport, which allows the transport of larger messages [156]. The authors announce a delay below $20\mu s$ for packets up to 15kB. According to their studies, in ideal conditions, they also present a throughput in the order of magnitude of $10^4 MB/s$.

- **Cyclone DDS**: Eclipse Cyclone DDS is designed as a free and open-source middleware. It is focused on high throughput and low delay, even if no official data are provided. The developers of FASTDDS have conducted a comparison with Cyclone, as cited in [157]. For intra-process communication, FastDDS presents significantly better delay results, especially for larger payloads. In the case of inter-process communication, FastDDS claims to have a delay of around 30% lower than Cyclone. They also highlight the stability of FastDDS, represented by the low delay increase as the payload gets larger. In terms of throughput, both DDSs present the same results.

Another possibility, commonly used by MRS, is to rely on another middleware than DDS. Eclipse Zenoh is very promising in this way, being close to natively implemented in the latest version of ROS 2. It offers the first easy-to-use RMW not relying on DDS. This technology is designed to streamline the discovery process and improve efficiency. During the discovery phase, robots attempt to identify the topics of other robots, which can be a crucial network operation. Fortunately, Zenoh has been developed to significantly reduce the overhead required for this process, cutting it down from 97% to 99.9%. This technology delivers results similar to those of DDS while minimising the required resources. Discovery overhead is frequently cited as a major problem with DDS technology in wireless networks, as it can cause network downtime [158].

In the paper [159], the authors explained that Zenoh is less likely to cause network failure compared to typical ROS 2 DDS. It is specifically created to operate smoothly even in unstable network conditions. One other advantage of Zenoh is the existence of pico Zenoh, a version compatible with microcontrollers. The only DDS implementation offering microcontroller-compatible features is FastDDS; however, various robotic applications use

microcontrollers for motion. In the case of the usage of Zenoh in the architecture, all the ROS 2 nodes within the robot r_i should share the same domain id D_i, but the domain id of each robot r_i should be different, $D_i \neq D_j, \forall\{i,j\} \in \{1,n\}(i \neq j)$. We note that all the available studies and comparisons [156, 157] use Gigabit Ethernet connections, and no quantitative data are given for Wi-Fi, mesh networking or any other wireless technology such as 5g/6g.

5.4.3 Problem Scenario

In the following experiment, we only evaluate the performance of RMW. However, depending on the scenario, the ideal RMW might differ. This is why we created a fictional yet realistic scenario that will allow us to select the ideal RMW. For this fictional scenario, we consider a group of robots exploring a harsh environment, specifically a lunar environment. They extend the group capabilities by working together in a decentralised manner while also maintaining their independence. Each robot continuously generates and transmits sets of data called 'messages'. As an example, these messages could represent small chunks of a point cloud acquired by various sensors. Multiple robots are expected to share their data (i.e. point clouds) for merging or optimisation operations [160, 161]. In order to map even more efficiently, a robot can obtain the point cloud data of another robot to prevent the need to map an area already visited, thus conserving energy. The state-of-the-art provides many mapping, task allocation, and control solutions [162]. However, most applications are based on simulations or highly controlled networks, and the networking aspects are often considered out of the scope. Especially given the recent developments with ROS 2. The state-of-the-art needs more propositions of MRS-proof network architecture. Currently, many works and projects rely on access points and utilise the default settings of ROS 2. As stated in [163], evolving network topologies remain an open question. As a result, there is a need for a study about the network aspects of MRS relying on a decentralised architecture. In this context, we use ROS 2 to implement a decentralised architecture. The main concerns of this problem to be surveyed are:

- Data throughput

- Stability of a robot on the network
- Power consumption

5.5 Real-world experiments

This section presents two real-world experimental campaigns conducted to evaluate the robustness and applicability of mesh networks in decentralised MRS using ROS 2.

The first experiment, described in an earlier unpublished version of this work and in section 5.5.2 , served as an initial validation of the proposed network and software architecture. It involved deploying mobile robots equipped with mesh antennas and evaluating the end-to-end communication performance of various ROS 2 middleware implementations under realistic conditions, including both LOS and NLOS scenarios. This study highlighted critical performance trade-offs in bandwidth, jitter, and CPU usage across FastDDS, CycloneDDS, GurumDDS, and Zenoh.

Based on the insights and reviewer feedback from that initial study, a second, extended experiment was conducted and forms the core of the present contribution as detailed in 5.5.6. This new campaign improves upon the original by refining the measurement protocol, extending the test scenarios, and introducing a more detailed comparison of middleware performance across varying message sizes and network conditions. It also clarifies the behaviour of relay nodes in mesh topologies and addresses limitations in discovery and routing overhead observed previously.

By presenting these two experiments in sequence, we aim to provide a complete view of the challenges and solutions in integrating mesh networking into decentralised ROS 2-based MRS. The following subsections will detail each experimental setup, environment, and the key findings derived from them.

5.5.1 Experimental Setup

Hardware

Most of the embedded computers come with an internal antenna. The straightforward approach would be to use the embedded antenna and run the routing protocol on the embedded computer. However, this solution faces multiple issues.

- Limitations: Internal antennas offer less data throughput and a limited range due to power and space limitations.
- Resource usage: Embedded computers are optimised for specific purposes and are usually used for computationally intensive tasks such as image processing. Introducing a network protocol layer would increase the amount of computation required and make the system more susceptible to internal crashes.
- Bottlenecks: Robotic systems frequently incorporate multiple computers within the platform, all connected via Ethernet. However, relying on one computer to manage all networking tasks could result in significant data throughput limitations and reduce efficiency.

On the other hand, using an external router presents multiple advantages. The first one lies in using a dedicated internal board designed for network computation, allowing a better process of the packages and optimised routing. An external power source can provide a more significant data throughput and broader coverage. One of its key features is the ability to quickly implement any networking protocol, which is crucial when experimenting. Notably, using an external antenna can increase power consumption, which should be taken into account as a potential concern.

We use the *Groove A52 ac* router from the brand *Mikrotik* [164] as the mesh node for every agent of the MRS. This router offers more than 200 meters of coverage and is compatible with the norms 802.11ac (Wi-Fi 5). The router's design also meets the need for outdoor robotics, with a weatherproof design that easily fits all current robots. Finally, it can be powered over Ethernet, providing a more straightforward wiring and power distribution approach.

The architecture proposed in Fig.5.3 is highly scalable and adaptable to any robot. Since many robots rely on multiple embedded computers, they all need to be connected to a Switch connected to the Groove A52 ac. The router used in our setup is an *EdgeRouter X* from the brand *Ubiquity* set up as a switch [165]. However, any switch providing power over ethernet is suitable.

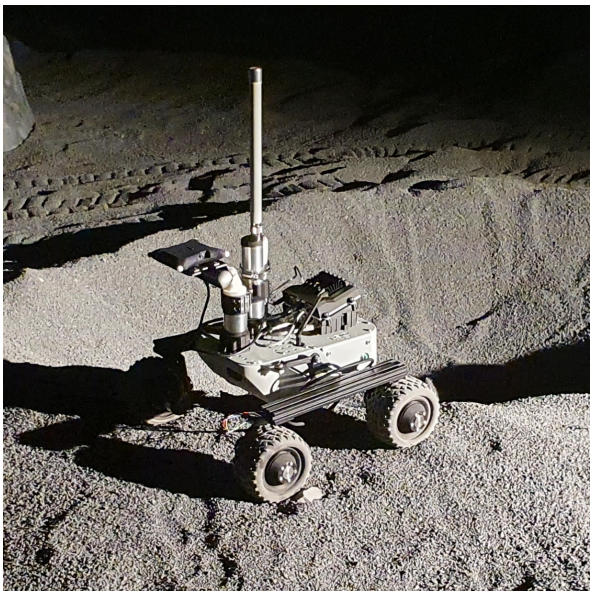


Figure 5.4: Experimental robotic platform

The robots used for the experimentations are upgraded LeoRovers [166] as in Fig.5.4, equipped with an RGBD *intel realsense_d455* camera [167], a *RPlidar* [168], an embedded computer used for the computational heavy tasks *Nvidia Jetson Xavier* [169] and the setup previously mentioned.

Software

As explained in [82], IEEE 802.11s defines HWMP as the default mesh protocol. The authors of [82] propose HWMP+ as a version more adapted to dynamic topology and optimising the shared channel resources. Most of the existing work was conducted in simulation and now requires further examination in mobile robotics.

To establish the mesh network between the robots, Mikrotik provides instructions [170],

creating a 2.4GHz Wi-Fi mesh network relying on the HWMP+ protocol. The **2.4GHz** band is chosen for its higher resistance to interferences and longer range. The authors of [82] propose HWMP+ as a version more adapted to dynamic topology and optimising the shared channel resources. Most of the existing work was conducted in simulation and now requires further investigation.

All the network communications of ROS 2 are handled by a RMW, a middleware providing a publish-subscribe model for sending and receiving data, events, and commands among the nodes. The RMW solves many issues when it comes to distributing real-time data, sending and receiving data over a network, and providing consistency in the data model. It is acting on layers 5 and 4 of the OSI network model. By definition, the mesh protocol assures the data link, so layer 2 of the OSI model. This means that there is no difference in ROS 2 between a mesh network and a regular network.

All the embedded computers are running on Linux Ubuntu 22.04.

In the case of the first experiment, each robot would run the ROS 2 humble distribution. As for the second experiment, each robot would use the ROS 2 Iron distribution. This distribution would allow a better implementation of Zenoh as a RMW.

5.5.2 First experiment

Metrics We consider multiple metrics to compare and validate the more appropriate system. The first set of metrics allows us to quantify the network in each scenario and are the following.

Bandwidth To fully understand how different middleware work with the mesh network, it's crucial to assess the bandwidth it provides.

Jitter Along a measurement, the latency measurements vary. This variation is called the jitter and indicates the stability of the network. In a system with low jitter, messages consistently take about the same time to travel from sender to receiver.

Packet Loss This metric is the percentage of packets that never reach their destination. They are an excellent input to evaluate the stability of the network. However, the experiments

showed no packet loss.

The second set of metrics applies to the whole implementation, measuring the performances of each DDS implementation:

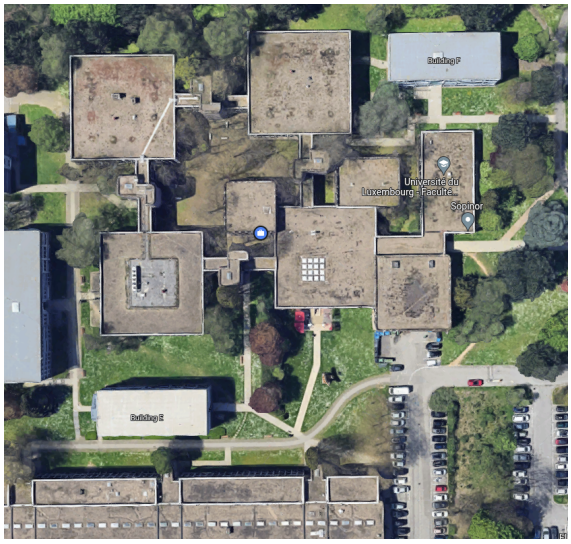
Bandwidth usage During the experiment, the robots will share messages of a fixed size over the network. This metric indicates the size of the communication depending on the size of the message to quantify the network impact of each middleware.

Latency The latency is a crucial metric for real-time systems, representing the time it takes for a message to travel between a publisher and a subscriber.

CPU Usage Most embedded computers have limited resources, so monitoring the usage of the resources by the various middleware implementations is helpful.

Experimental Scenario

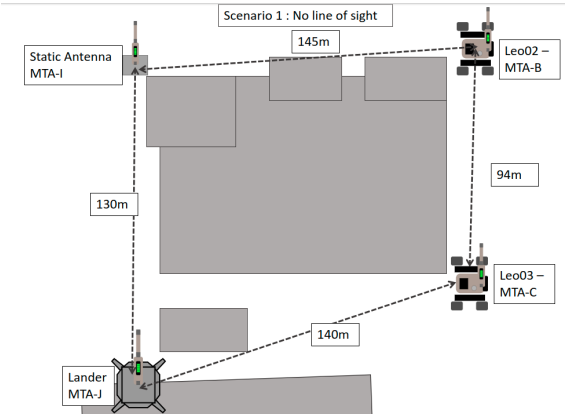
To perform the network evaluation experiments, the environment will be an urban environment (fig. 5.5a) and a crop (fig. 5.5b) field. As stated previously, ROS 2handles communication through middleware. Since all the existing studies only present results in ideal conditions, we are proposing an approach to push the middleware to their limits. As detailed in Fig.5.6, this experiment involves two robots (Leo02 and Leo03), a lander and a static antenna. The scenario involves measuring the various Key performance indicator (KPI)s of the network while a fixed-size message is communicated from the farthest robot, Leo02, to the lander. Keeping the same distances between the robots, we run the experiments in two different environments. The first one (Fig.5.6a) is a dense environment that does not allow direct LOS between Leo02 and the lander, the NLOS scenario, while the second offers (Fig.5.6b) a plain environment for a LOS scenario. The distance between the nodes of the mesh is determined to offer real conditions. We also propose to overview the mesh network topology during the experiment.



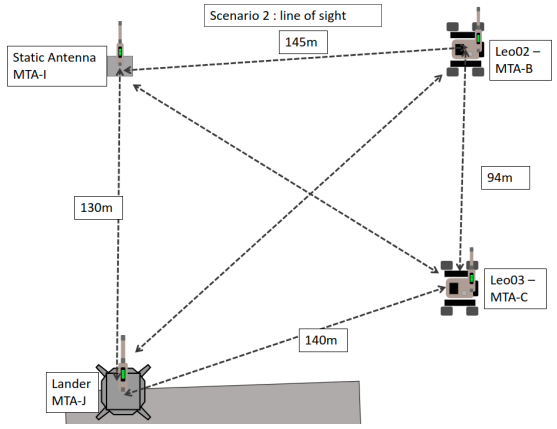
(a) Urban terrain used for the first experimental scenario

(b) Crop field used for the second experimental scenario

Figure 5.5: Experiments environments



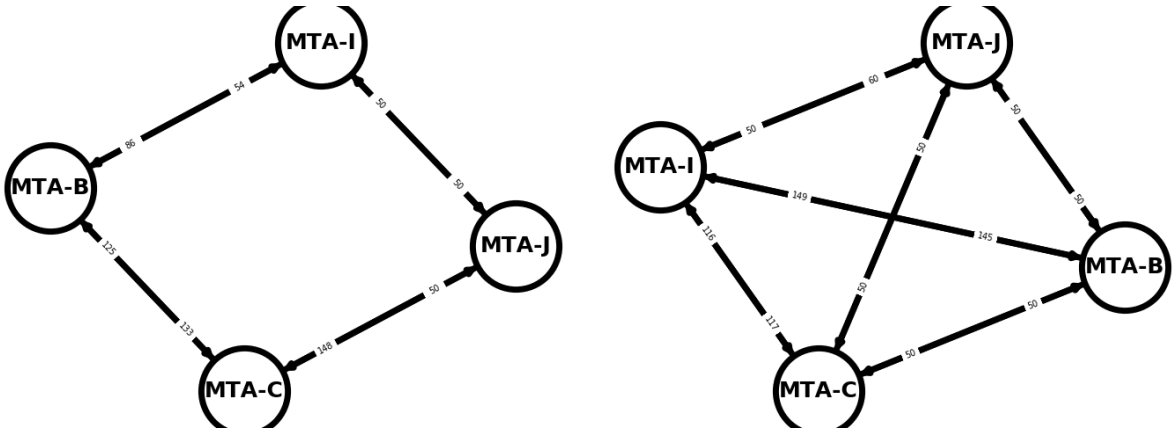
(a) First Scenario with no direct line of sight between the lander and Leo02



(b) Second Scenario with direct line of sight between the lander and Leo02

Figure 5.6: Experimental Scenarios

5.5.3 Results and discussion of the first experiment



(a) Typical network topology of the non-line-of-sight experiment

(b) Typical network topology of the line-of-sight experiment

Figure 5.7: Result of the network topology analysis during the experiment. A circle represent a node of the mesh network connected to its direct neighbour. The number closer to a node is the value of the metric to go from this node to the neighbour. MTA-B is Leo02, MTA-C is Leo03, MTA-I is the static node, and MTI-J is the lander node

A GUI tool helped us to keep an overview of the network topology during the experiments. Fig.5.7a displays the typical topology of the non-line-of-sight experiments, while Fig.5.7b displays the topology of the line-of-sight one.

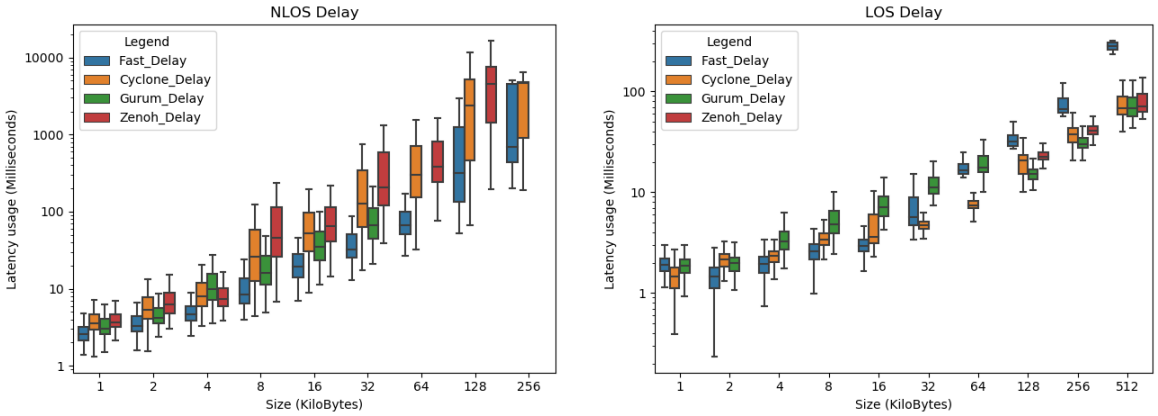
The Tab. 5.1 presents the average capabilities of the mesh network in scenarios NLOS and LOS.

Scenario	NLOS		LOS	
	lander to Leo02	Leo02 to lander	lander to Leo02	Leo02 to lander
Bandwidth	9.65 Mbits/s	8.11 Mbits/s	82 Mbits/s	49 Mbits/s
Jitter	0.205ms	0.507ms	0,168 ms	0.310ms

Table 5.1: Network Metrics over the mission without ROS 2

Since the **packet loss** and **latency** of the ping command do not vary in any scenario, they are not included in the Tab.. It is noticeable in Tab. 5.1 that using a relay like in the NLOS scenario impacts the available bandwidth by a factor of five to ten.

The following three parameters can also explain this impact:



(a) Delay of communication between Leo02 and the lander in the NLOS scenario

(b) Delay of communication between Leo02 and the lander in the LOS scenario

Figure 5.8: Box Plots representing the delays in milliseconds on a logarithmic scale relative to the message size. Each colour represents a DDS middleware implementation.

1. Data packets going through a relay adds a lot of overhead. As seen in Fig.5.7b, in the LOS scenario, the data can go directly to the lander.

2. The distance from the lander to the rover is 200 meters. The connection with the relay is 250 meters long.

3. The environment of the NLOS experiment is a dense urban environment, implying a wide variety of interferences. At the same time, the LOS has been performed in an empty field.

In the scenario where node A and node B want to communicate we learn that if a Robot C wants to offer relaying services, it should only get concerned when A and B are on the edge of the communication span.

It is also noticeable in Tab. 5.1 that the jitter is slightly impacted by the NLOS scenario but not in a significant manner. A more impactful data is the impact of the emitter and receiver. Having the lander as an emitter provides better results than having the robot. These different results can be explained by the fact that the lander was an antenna attached to a building, in a higher pose than the robots, close to the ground.

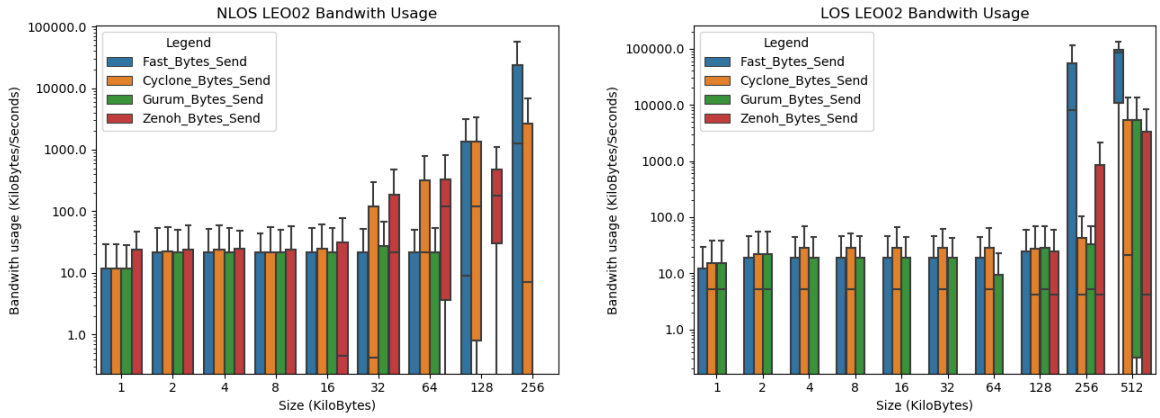
The following subsections describe the evaluations of the studied KPIs: Delay, bandwidth

usage and CPU usage, depending on the message size in both scenarios. The results are displayed in a box plot with the message size as the y-axis and the studied KPI as the x-axis. For clarity, the outliers have been removed from the figures 5.8,5.9,5.10,5.11,5.12. We then offer a comparative analysis of both scenarios with recommendations for different cases.

	NLOS	LOS
FastDDS	256KB, delay incr.	512KB, no msg
CycloneDDS	512KB, MTA_B disc.	1MB, no msg
GurumDDS	64KB, Error	1MB, no msg
Zenoh	256KB, no msg	1MB, no msg

Table 5.2: Limits and behaviours for bigger packages

The Tab. 5.2, lists the maximum message size experimented for each DDS, where it would fail and increase the latency exponentially, making the reading too difficult.

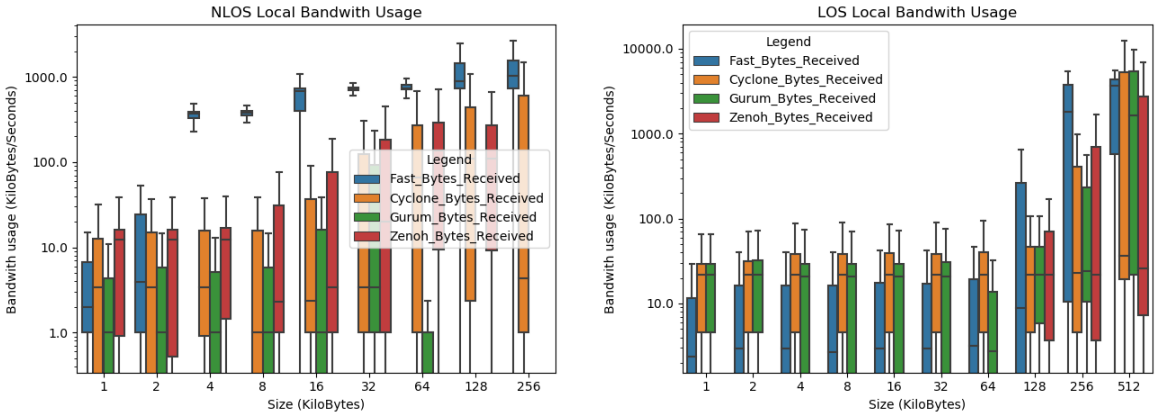


(a) Bandwidth usage by Leo02 in the NLOS scenario

(b) Bandwidth usage by Leo02 in the LOS scenario

Figure 5.9: Leo02 bandwidth usage in KB/s on a logarithmic scale for each message size and middleware implementation in both scenarios

Finally, it is worth noticing that there are missing results for Zenoh in the LOS scenario for messages smaller than 128KB, visible in fig 5.8b,5.9b,5.10b,5.11b,5.12b. Due to weather and time conditions, these experiments could not be realised, but as explained earlier, the heart of this study lies on the NLOS scenario and bigger messages.



(a) bandwidth usage by the Lander in the NLOS scenario

(b) bandwidth usage by the Lander in the LOS scenario

Figure 5.10: Lander bandwidth usage in KB/s on a logarithmic scale for each message size and middleware implementation in both scenarios

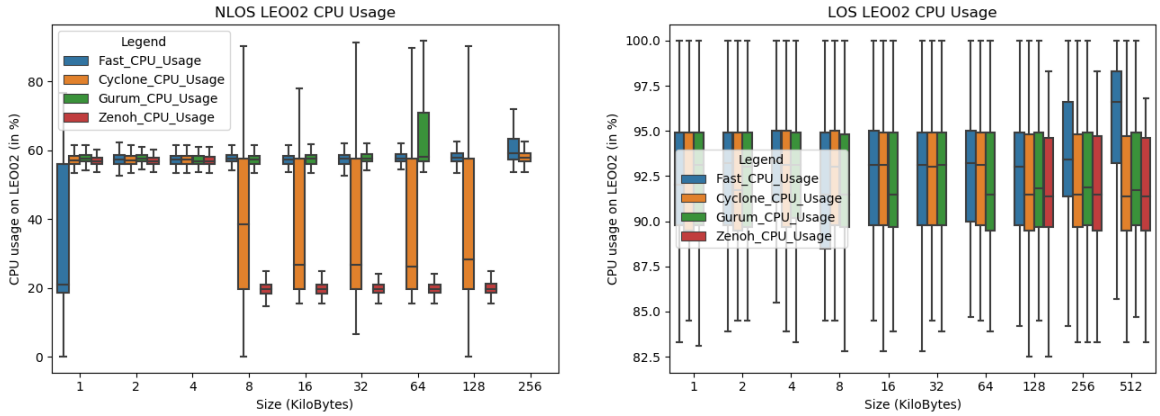
5.5.4 Delay

In Fig.5.8, the box plots reveal the latency in seconds for varying message sizes transmitted using different middleware implementations under NLOS and LOS conditions. Notably, FastDDS shows consistently lower latencies for messages up to 32KB in both scenarios, aligning with its performance claims. Interestingly, in the LOS scenario, as message size increases, other middleware outperform FastDDS, suggesting that FastDDS may be optimised for smaller messages under these conditions.

In contrast, the NLOS scenario depicts a different pattern, where FastDDS demonstrates lower latencies across all message sizes, indicating robustness in less ideal communication environments.

For this experiment, the implementation of Zenoh relies on a bridge running over CycloneDDS, and we can see that the delay for Zenoh communication might be dependent on Cyclone since they seem correlated in figure5.8.

In the context of extreme environments, the latency of communication is a less crucial parameter since a more significant delay is often implied by the conditions. For example, in the case of lunar exploration, we can expect a delay between the Earth and the Moon of 2



(a) CPU usage in percentage on Leo02 in the NLOS Scenario

(b) CPU usage in percentage on Leo02 in the LOS Scenario

Figure 5.11: Box Plot representing the CPU usage in percentage on Leo02 for each message size and middleware implementation in both scenarios

seconds.

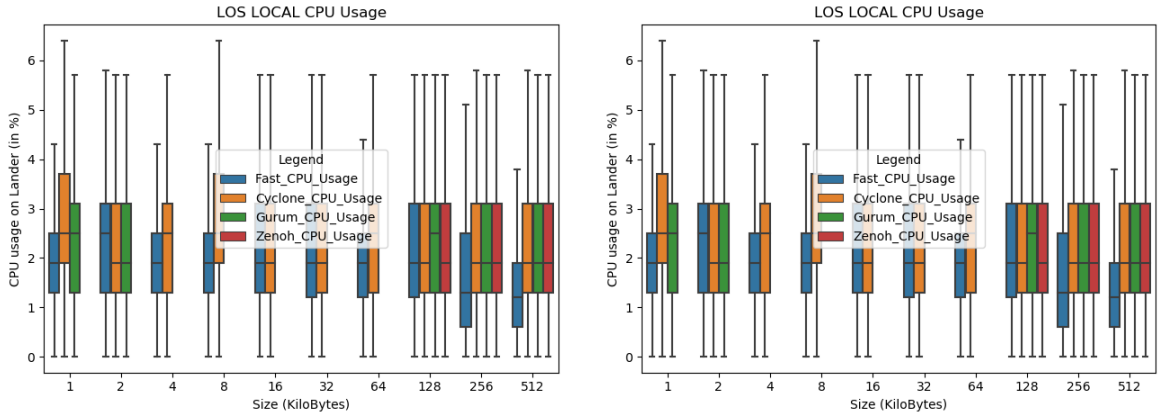
5.5.5 Bandwidth

Fig.5.9 shows the bandwidth usage on the robot Leo02 to send a fixed-size message, while Fig.5.10 shows the bandwidth usage on the lander to receive the messages.

The data reveals a marked increase in bandwidth consumption by FastDDS as message sizes grow, particularly under the NLOS scenario. This trend indicates that FastDDS bandwidth is less efficient for larger messages compared to other middleware options.

On the other side, Zenoh demonstrates improved performance when handling messages over 64KB, reducing the bandwidth usage by 400% compared to FastDDS on Leo02, suggesting that it is more suitable for transmitting larger messages of data. This becomes particularly relevant in mesh network configurations, where an overloaded node can lead to severe network consequences, such as being disconnected due to excessive bandwidth consumption.

In the context of network capacity, as referenced in 5.1, the LOS scenario bandwidth peaks at approximately 6.25MB/s. Therefore, FastDDS's average consumption of 8MB/s for 256KB messages can affect the network. This is evidenced by the observed delays and the



(a) CPU usage in percentage on the lander in the NLOS Scenario

(b) CPU usage in percentage on the lander in the LOS Scenario

Figure 5.12: Box Plot representing the CPU usage in percentage on the lander for each message size and middleware implementation in both scenarios

potential for network crashes or bottlenecks as the system tries to cope with the data load. It is crucial for network stability to manage these loads effectively, particularly when considering real-time communication requirements in a robotic mesh network.

CPU usage

Fig.5.11 illustrates the CPU usage by the robot LEO02 when sending messages of fixed sizes for both scenario, while Fig.5.12 displays the CPU usage by the lander to receive these messages.

Upon examination, in the LOS scenario, the CPU usage is relatively invariant across the different middleware solutions for both the robot and the lander. This consistency suggests that middleware choice, in terms of CPU overhead, is not a significant factor for operations within the LOS scenario.

However, in the NLOS scenario, the usage of bandwidth is significantly impacted. For the robot Leo02, the employment of Zenoh middleware results in substantially lower CPU usage—approximately one-third that of the DDS implementations like FastDDS and CycloneDDS. This reduction in CPU load is crucial in space applications where computing resources are scarce and typically reserved for critical autonomous operations.

Similarly, on the lander, FastDDS middleware exhibits CPU usage that is up to four times that of the alternative options. Such a high demand for computational resources could detract from other essential processes on the lander, particularly during complex operations where efficiency and resource management are pivotal.

Overall, in a space application of a Mesh Network for robotic exploration, robots have limited computation capabilities that should be saved for autonomous behaviours. A solution like Zenoh would offer way better performance.

Given the constraints of a space-based mesh network, where robotic explorers operate under computational limitations, middleware that minimises resource consumption without compromising communication reliability is invaluable. Zenoh emerges as a particularly attractive solution in this context, offering superior performance by significantly reducing the computational overhead on both the robot and the lander.

Summary

Table 5.3 displays an overview of the performances of each middleware in different testing conditions. This table shows the performances of each middleware in the various scenarios and can be used to choose the most appropriate depending on the conditions. Symbols display how efficient the middleware is on a specific point of interest compared to the others. A ”++” indicates that the system outperforms the others in the scenario, while a ”- -” indicates major performance issues compared to the other approaches. The intermediary symbols show: ”+” good results, ”+/-” acceptable results and ”-” less good results than the others. Finally, the ”?” indicates missing values.

In the case of extreme environment exploration, delays in communication are to be expected, as the user is operating the mission from a remote place. However, bandwidth usage and CPU usage are keys of interest. Since the Mesh Network can offer limited performances depending on the environment, it is designed to kick out any node using too much bandwidth, so the selected solution should focus on optimising the bandwidth usage. Exploration rovers must execute various mapping and navigation software that are resource-expensive. Saving computational power should also be a concern in the choice.

			Fast	Cyclone	Gurum	Zenoh
NLOS	Small Messages	Delay	++	+/-	+	-
		Bandwidth	-	+	++	+/-
		Cpu Usage	+/-	+	+	+
	Larger Messages	Delay	++	+	- -	-
		Bandwidth	- -	++	- -	++
		Cpu Usage	- -	+	+/-	++
LOS	Small Messages	Delay	++	+	+	?
		Bandwidth	++	+/-	+	?
		CPU Usage	+	-	+/-	?
	Larger Messages	Delay	- -	+	++	+
		Bandwidth	- -	+	+	++
		CPU Usage	+/-	+/-	+/-	+/-

Table 5.3: Summary table of DDS performances

Since the primary concern of this study focuses on space exploration, it is expected to have more significant messages to transfer and go through relay points. The grey part of Tab. 5.3 highlights the corresponding scenario, and two middleware stand out, Cyclone and even more Zenoh. Indeed, we show that, for large messages in a NLOS scenario, Cyclone reduces by 8,86 times bandwidth and 2,07 times the CPU usage compared to FastDDS while Zenoh reduces by 4,01 times the bandwidth and 4,03 the CPU usage compared to FastDDS. It is still important to notice that, in the NLOS scenario, Cyclone and Zenoh increase the delay compared to FastDDS by respectively 65% and 77%.

Identified limitation of the first experiment

The preliminary experiments conducted in this study provided critical insights into the behaviour of ROS 2 middleware under mesh network conditions. Our evaluation revealed

several bottlenecks, including inconsistent reachability, excessive message overhead, and unpredictable CPU and RAM usage across various middleware implementations.

Most notably, As reviewers highlighted, the experiments were unable to fully capture the effects of dynamic topologies and hardware heterogeneity—all of which are expected in real-world applications such as lunar exploration missions.

Additionally, We expected that Zenoh could provide better results by communicating natively with ROS 2 rather than through the bridge. Since Zenoh was not yet supported by ROS 2 natively, all the existing nodes must communicate through the bridge. However, in [171], the open robotic foundation offered a study and stated that they are working on a native installation of Zenoh for the upcoming version of ROS 2, Jazzy Jalisco.

5.5.6 Second Experiment

These shortcomings motivated a more rigorous investigation, which is the focus of the subsequent study. The experiment presented in this section—published as a peer-reviewed contribution in the JIRS journal—introduces the following improvements:

- A **realistic exploration scenario**, involving dynamic robot motion, relay topology changes, and message broadcasting under constrained bandwidth;
- A **hardware-diverse testbed**, including embedded and external compute nodes connected via HWMP+ mesh routers in an urban field test;
- A **more realistic application of Zenoh**, by upgrading the ROS 2 version to Iron, they robot got the ability to use Zenoh the exact same way as any other DDS, avoid its routing through CycloneDDS like in the previous experiment
- A **comparative evaluation of three RMWs** (FastDDS, CycloneDDS, and Zenoh) across multiple message sizes, using quantitative metrics such as reachability, jitter, delay, CPU/RAM usage, and message overhead;
- A **dedicated network monitoring layer** to isolate the influence of network conditions from middleware behaviour.

By addressing the limitations of our earlier setup, this extended study aims to provide actionable guidance on middleware selection for mesh-based MRS in extreme environments. The following sections presents the full methodology and results of this improved investigation.

Main differences with the first experiment

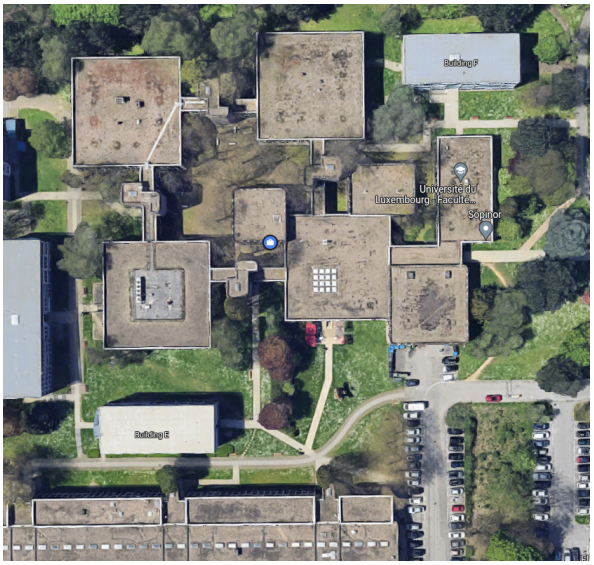
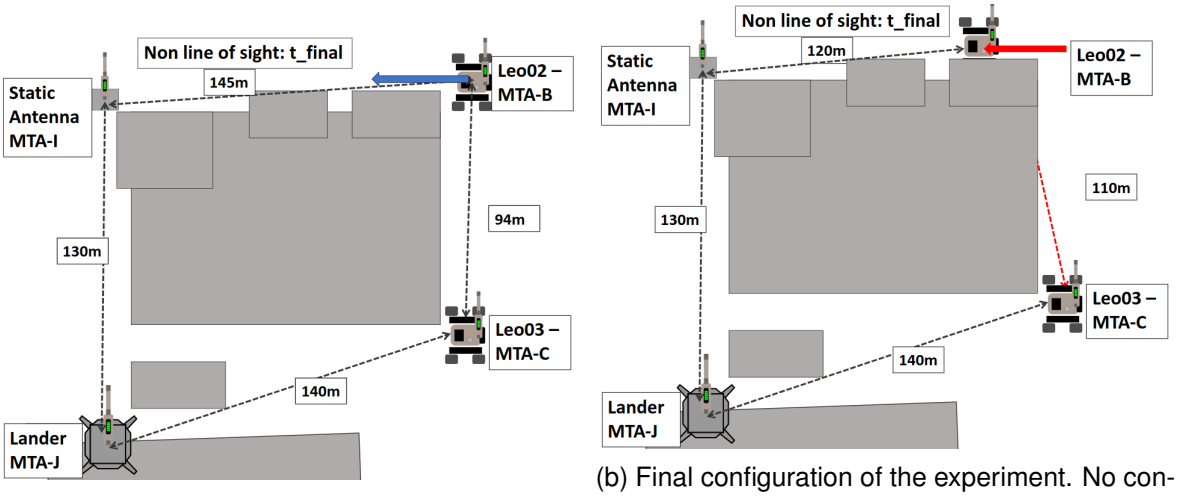


Figure 5.13: Urban terrain used for the second experimental scenario. The width of the visible environment is approximately 175 m.

The second experimental setup was designed to overcome the limitations identified in the first study and to push the evaluated middleware to more realistic and stressful operating conditions. Unlike the previous experiments—which were conducted in both LOS and NLOS conditions—the new test scenario takes place in an urban environment filled with obstacles and signal interferences (Fig. 5.13). LOS scenarios were omitted in this phase, as earlier results already provided sufficient insight into middleware behaviour under ideal conditions. The focus is now explicitly shifted to NLOS scenarios, which better reflect the unpredictable and degraded conditions encountered in real exploration missions.

All robots and routers in this test use the ROS 2 Iron distribution to ensure compatibility with the latest middleware versions, including native Zenoh support. In contrast to the earlier experiment, where all robots shared similar configurations, this setup introduces hardware



(a) Initial configuration of the experiment.

(b) Final configuration of the experiment. No connection between Leo02 and Leo03.

Figure 5.14: Experimental Scenarios

heterogeneity. As detailed in Fig.5.14, the experiment involves two robots (Leo02 and Leo03), a lander, and a static antenna. *Leo02* features two embedded devices: an internal onboard computer for low-level control (excluded from the mesh network), and an external computer responsible for all mesh communication and middleware switching. *Leo03* is repurposed solely as a relay node, helping to dynamically alter the network topology during the test.

The experiment is divided into the three evaluated RMWs that are each subdivided into seven fixed-sized messages (from a kilobyte to 64 kilobytes). This upper limit of 64 kilobytes has been found by preliminary tests, larger message sizes breaking the mesh network. To improve the quality of the result and get a proper sample, five successful runs are completed per fixed-size messages leading to 3x7x5=105 runs. A run has a duration of 120 seconds and consists of Leo02 moving toward the static antenna while sending fixed-size messages to the Lander. This trajectory, fig.5.14b, has the purpose of changing the network topology (because of the surrounding buildings) from a square shape (Fig.5.17a) to a line shape (Fig.5.17b) and investigate its impacts over each RMW. The messages sent on the network allow us to monitor the KPIs seen in section 5.5.6. The data transfer occurs in parallel: while moving, *Leo02* sends fixed-size messages to the lander, mimicking data-heavy navigation or sensing tasks in degraded conditions (see Fig. 5.14b).

Finally, while ROS 2 introduces QOS profiles that significantly influence network behaviour, especially for large messages such as point clouds—we intentionally kept all tests under the `default` QOS profile. This decision isolates the effects of the RMW itself from those of QOS tuning, focusing the analysis on middleware efficiency alone, as originally motivated by [37].

Evaluation Metrics

We consider multiple metrics to compare and validate the most appropriate system. The first set of metrics allows us to quantify the network during the two crucial steps of the scenario, which are as follows:

- **Data throughput**: Assessing the data throughput provided by the mesh network is crucial to fully understand how different middleware work together.
- **Jitter**: The delay measurements vary along a measurement. This variation is called the jitter and indicates the stability of the network. In a system with low jitter, messages consistently take about the same time to travel from sender to receiver.

The second set of metrics is measured during the whole experiment, measuring the performance of each RMW implementation:

- **Reachability**: Since the experiment consists of sending data from a robot to a lander, this metric evaluates when the robot is reachable from the lander. A device is considered reachable from another device if a connection can be established. We define the average reachability $R(t)$ for an RMW as follows: With $t \in [0, 120], \tau_i(t) \in \{0, 1\}$:

$$R(t) = \frac{1}{N} \sum_{i=1}^{N} \tau_i(t) \tag{5.1}$$

  with $\tau_i(t)$ the reachability for a given run i, N the total number of run for the RMW. The average reachability $R(t)$ is a floating-point value between 0 and 1. Higher values indicate a more stable connection.

- **Data overhead**: During the experiment, the robots will share messages of a fixed size over the network. However, each technology adds its own headers and additional content to the message. This metric studies the size of this overhead depending on the size of the message to quantify the network impact of each middleware.
- **Delay**: The delay is a crucial metric for real-time systems, representing the time it takes for a message to travel between a publisher and a subscriber.
- **CPU Usage**: Since most embedded computers have limited resources, monitoring the usage of these resources by various middleware implementations is helpful.
- **RAM Usage**: Along with the CPU Usage, the RAM usage remains crucial since the memory is shared between many of the running processes and should remain available.

Results

As explained in section 5.5.6, each RMW has been studied with seven message types over five distinct runs. Fig.5.15 demonstrates an overview of the experiment, including the robot's trajectory given each RMW. Fig.5.16 presents the network overview of each of these singular runs. Both figures emphasise the connection lost between 60 and 90 seconds, allowing the network to change its topology.

The Fig.5.15 has a zoom on the trajectories of each RMW. The missing values in the trajectories represent the loss of connection of the rover. During those 'blackout' moments, the network is either saturated or in reconfiguration. A closer look at the trajectories allows us to identify three different phases. During the first phase (I), the rover is connected to the mesh network in a full square topology (see Fig.5.17a) and therefore enhances good communication of the packages over the mesh allowing multiple routes. The second phase (II) is the transition phase where the mesh changes its topology due to the moving antenna on Leo02. During this phase, loss of connection, delayed messages and an unstable network are expected. Finally, the third phase (III) shows a reconfiguration of the network in 'line' (see Fig.5.17b).

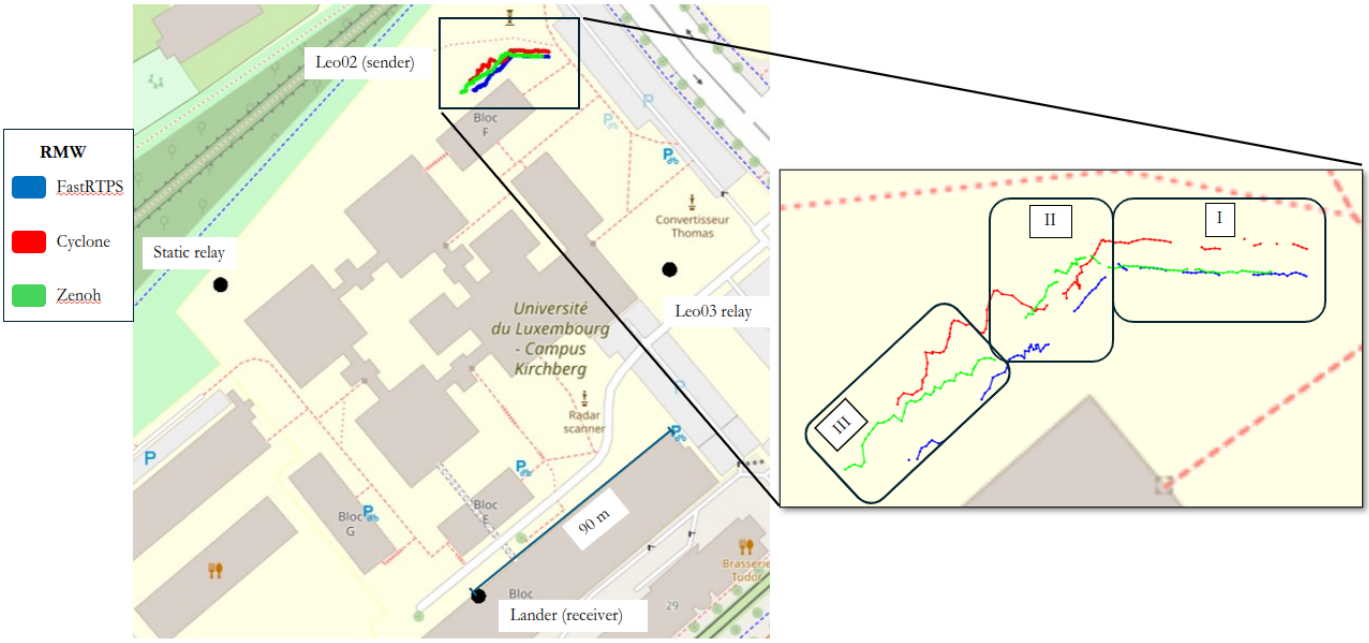
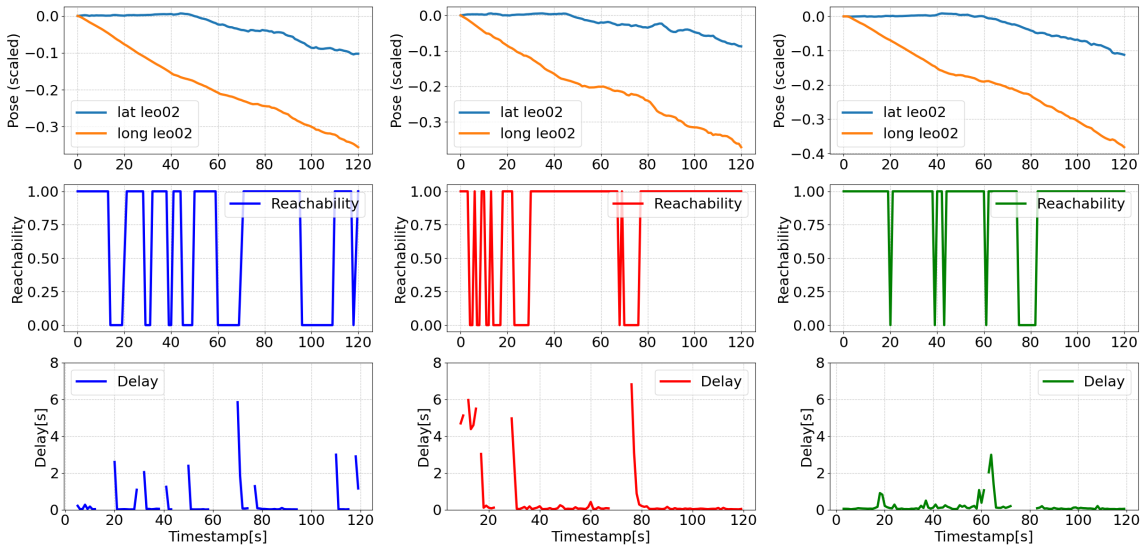


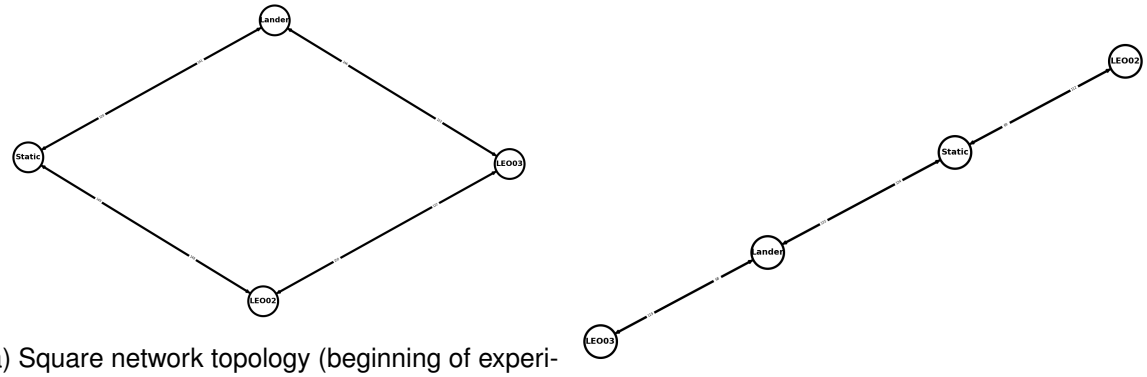
Figure 5.15: Global overview of the experiment



(a) Fifth run of Fast DDS (b) First run of Cyclone DDS (c) third run of Zenoh RMW

Figure 5.16: Overviews of a singular run for each RMW for a message size of 16 KiloBytes. Trajectory, reachability and delay are shown

In addition to the map overview, the Fig.5.16 shows how the network behaves during the experiment. The top graph represents the position of the rover related to its starting point, the middle graph is a metric called 'reachability' (if the rover is connected to the network or not), and lastly, the delay. Missing data in the delay graph is due to a connection loss between the rover and the mesh network.



(a) Square network topology (beginning of experiment)

(b) Line network topology (end of experiment)

Figure 5.17: Result of the network topology analysis during the experiment done with a GUI. A circle represent a node of the mesh network (antenna) connected to its direct neighbour.

A GUI tool helps us to keep an overview of the network topology during the experiment. Fig.5.17a displays the topology at the beginning of the experiment (t=0), while Fig.5.17b displays the topology at the end of the experiment (t=120s). This 120-second duration was decided after measuring the time for the network to change its topology while driving at a speed of 0.5m/s. After measurement, 75 seconds were enough to reorganise the network. However, all the runs have a duration of 120 seconds to fully understand the behaviour of the network before and after the change in topology. Table 5.4 presents the average capabilities of the mesh network at the beginning of the scenario (t=0) and the end of the scenario (t=120).

Scenario	t=0		t=120	
	lander to leo02	leo02 to lander	lander to leo02	leo02 to lander
Data throughput (TCP)	270 Kbits/s	164 kbits/s	1.31 Mbits/s	1.15 Mbits/s
Jitter (UDP)	0.000ms	245.385ms	0.000 ms	7.850ms

Table 5.4: Network Metrics over the mission without ROS 2

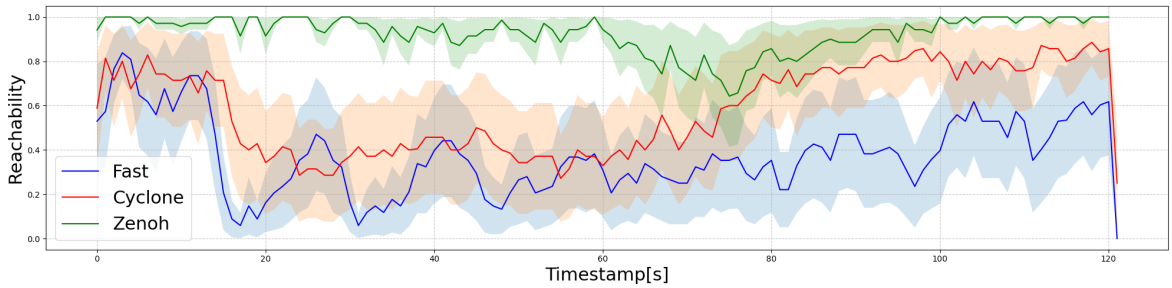


Figure 5.18: Average reachability over all the runs for each RMW with the variance.

Since the **packet loss** and **delay** of the ping command do not vary in any scenario, they are not included in the table. It is noticeable in Tab. 5.4 that the distance between the robot and the relay impacts the data throughput and that relays should be used carefully. This behaviour can be explained by the number of packet losses: the further the antennas are spread, the higher the chances of packet loss. Therefore, reduced usable bytes are received per second.

In the scenario where node A and node B want to communicate, we learn that if node C wants to offer relaying services, it should only get concerned when A and B are on the edge of the direct communication span. It is also noticeable in Tab. 5.4 that the jitter is highly impacted by the distance between each node. Also, using the lander as an emitter provides better results than the robot. Those results can be explained by the lander's antenna being attached to the first floor of a building and therefore offering better coverage.

One hundred five runs were initially expected; however, we encountered some failures during the experiment, leading us to perform 136 runs instead, giving a success rate of 77%. It is noticed that this success rate varies depending on the RMW, 73% for fast, 77% for cyclone and 83% for Zenoh.

The following subsections describe the evaluations of the studied KPIs: reachability, delay, data overhead, CPU usage, and RAM usage over time.

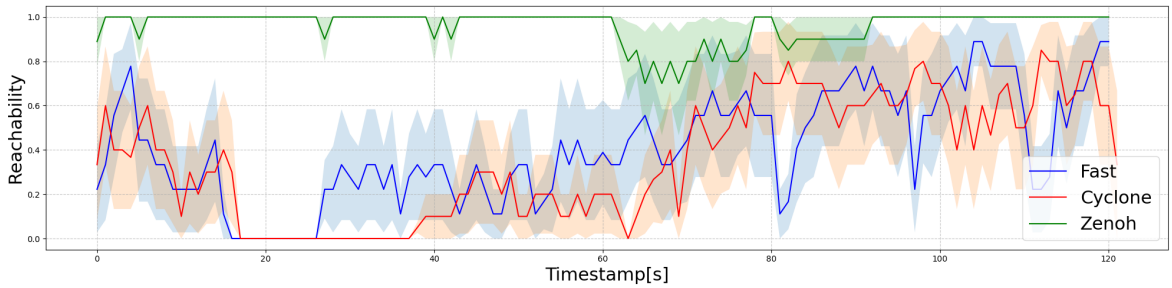


Figure 5.19: Average reachability for messages of sizes one and two kilobytes

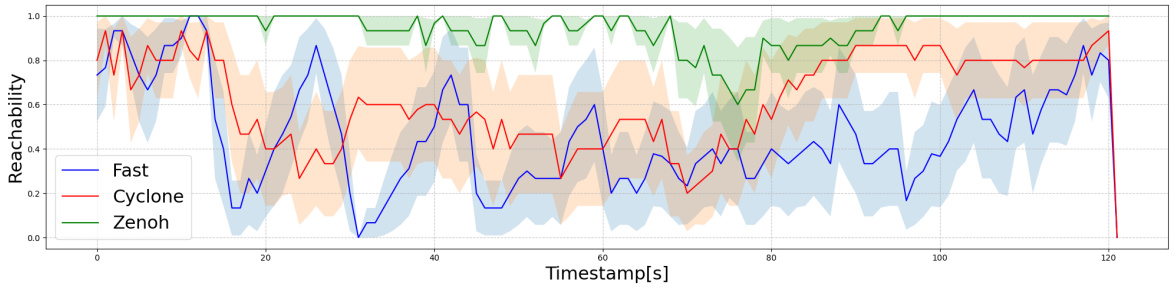


Figure 5.20: Average reachability for messages of sizes four, eight, and sixteen kilobytes

5.5.7 Reachability

In our scenario, the reachability of Leo02 from the lander is evaluated. Based on the introduced formula for the reachability, equation 5.1, N is the total amount of run per RMW (i.e 35), $\tau_i(t) \in \{0, 1\}$ represents the reachability of Leo02 at time t and $R(t)$ is the average reachability of a RMW at a time t. Fig.5.18 displays the average reachability $R(t)$ with its variance for each RMW given time.

The data reveals a similar behaviour for Fast and Cyclone, all relying on the DDS technology; however, Zenoh is showing significant improvements in terms of network stability. Figures 5.19, 5.20, and 5.21 show the average reachability for three different message groups. Fig.5.19 corresponds to the smallest messages: one and two kilobytes. Fig.5.20 highlights the medium messages (4, 8, and 16Kb), while Fig.5.21 shows the bigger messages (32 and 64Kb). Both small and medium messages as shown in Fig.5.19 and 5.20 emphasise Zenoh that outperforms the DDS (Fast and Cyclone) in terms of reachability. However, it is interesting to say that the results for bigger messages shown in Fig.5.21 mitigate the

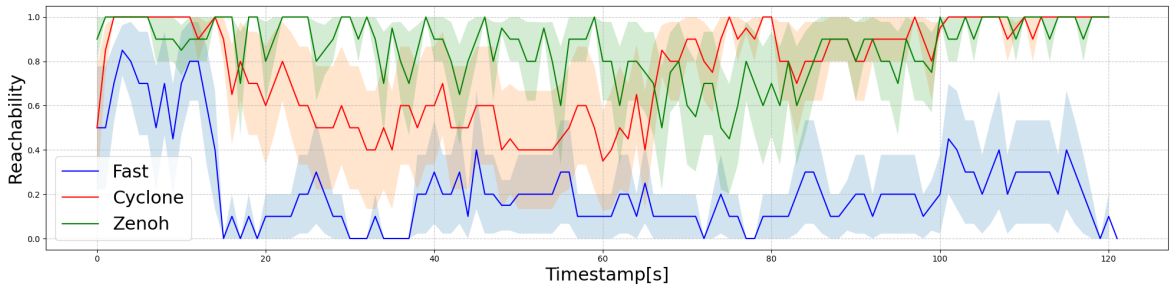


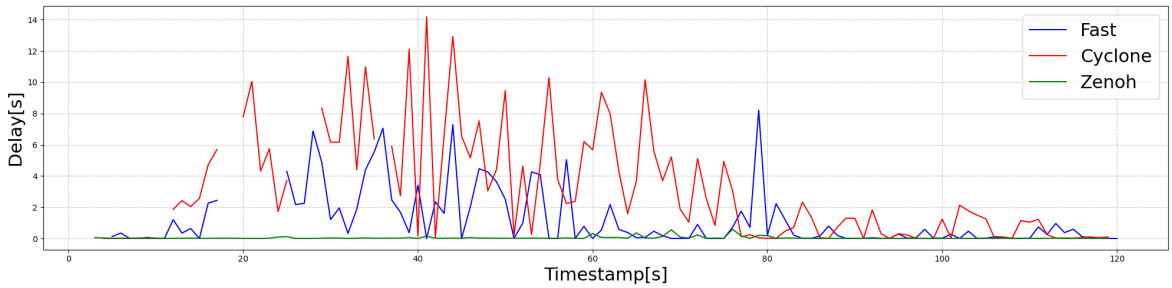
Figure 5.21: Average reachability for messages of size 32 and 64 kilobytes for each RMW with the variance.

performance of Zenoh with comparable results with Cyclone. On the other hand, Fast is underperforming compared to the other RMW.

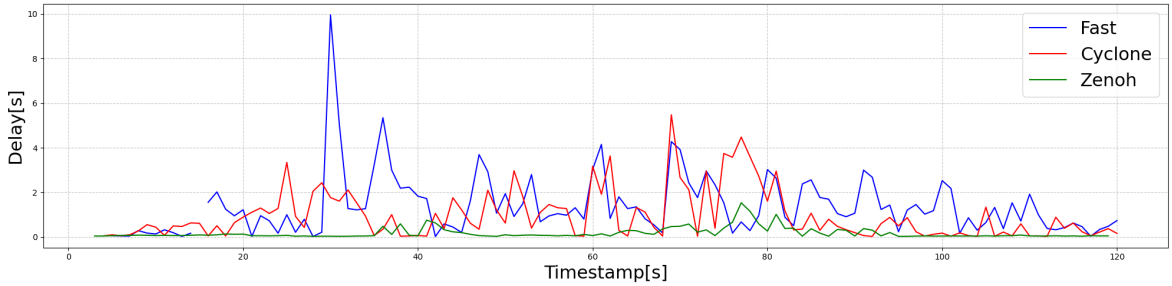
5.5.8 Delay

In this section, the delay between each message received is studied. For reference, a message is sent by Leo02 every 0.5 seconds on the network, but might be lost or corrupted during the transfer. This is why the delay in receiving the messages is studied; a higher delay means more time to update a potential point cloud or outdated information about the respective rover's residual energy.

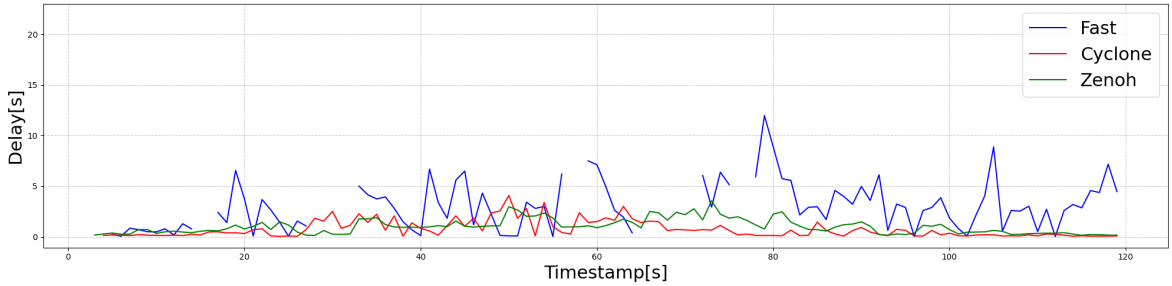
Fig.5.22, shows the average delay in seconds for various message sizes transmitted using the different middleware implementations. Compared to reachability, the variance is not displayed to enhance readability (extremums lead to the flattening of the graph). However, it is notable that FastDDS presents way more variance than Zenoh or Cyclone, reaching values up to a fifty-second delay. Fig.5.22c shows that Zenoh outperforms the DDSs by an order of scale. It is also interesting to notice that FastDDS performs better than Cyclone for very small messages, which follows FastDDS's claims. On the other hand, Zenoh has comparable results to Cyclone for large messages following the reachability tendency. It is also important to notice that Cyclone and Fast seem to be impacted by a less stable network while Zenoh remains stable all along. Around the end of the experiment, when the data throughput exceeds 1Mb/s (table.5.4), both DDSs seem more stable.



(a) Average delay for messages of sizes one and two kilobytes



(b) Average delay for messages of sizes four, eight, sixteen kilobytes



(c) Average delay for messages of sizes thirty-two and sixty-four kilobytes

Figure 5.22: Average delay in seconds over three packs of message size for each RMW without the variance.

In the context of extra-terrestrial extreme environments, the delay of communication is a less crucial parameter since any delay induced by the network would be negligible compared to the delay between the operator and the robot. For example, in the case of lunar exploration, we can expect a delay between the Earth and the Moon of 2 seconds. Such delay already removes the robot's real-time teleoperation capabilities.

5.5.9 Data overhead

The data overhead is characterised by two metrics: the bytes sent by Leo02 and the bytes received by the lander. Since for all the RMWs, the messages are of the same size, measuring the total bytes sent gives an overview of the message overhead of each RMW. Fig.5.23 shows the average bytes sent by Leo02 over the different RMW. As shown in the graph, this value does not fluctuate a lot except for Fast during a run. These peaks may be one of the reasons for the largest network instability led by Fast. It seems that Zenoh has an overall greater data overhead, this represents an issue in the data measuring. The function used to measure the data overhead took into account not only the traffic on the Wifi interface but also the Zenoh router, multiplying by two the real overhead. In that case, it appears that Zenoh shows a way smaller data overhead than the two other RMW.

Since the average bytes sent by each RMW do not fluctuate much over time, Fig.5.24 offers a Box Plot graph featuring the bytes sent by Leo02 for each message size and RMW 5.24a along with a Box Plot featuring the bytes received by the lander for each message size and RMW 5.24b. These graphs also take into account the latest remarks, dividing the bytes sent and received for Zenoh.

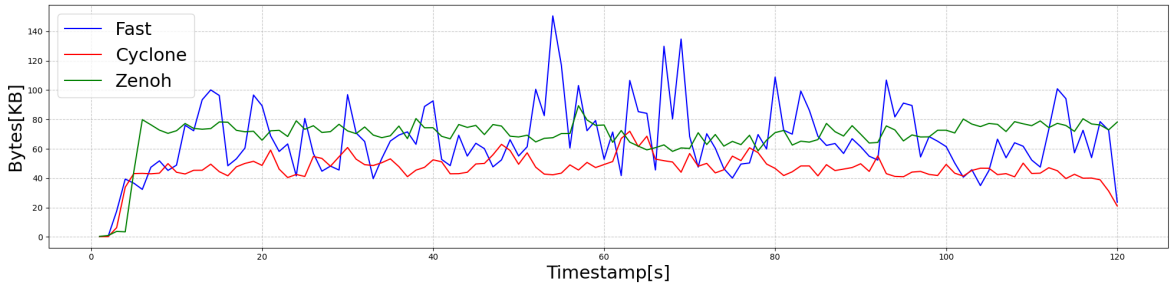
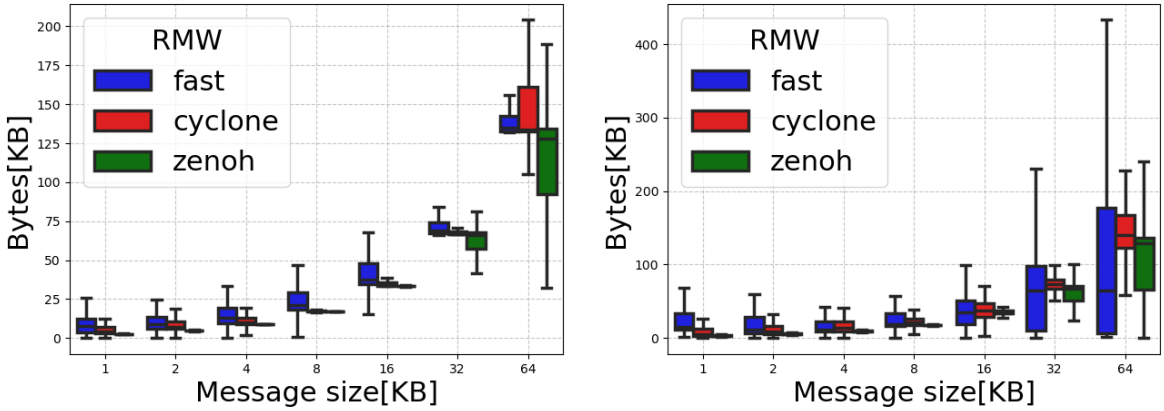


Figure 5.23: Average bytes sent by Leo02 over all the runs for each RMW without the variance.

Fig.5.24 provides a better understanding of the impact of each RMW over different message sizes. The results of the lander and Leo02 are comparable as expected, except for a slight increase of bytes on the lander side (scale differs on both graphs; Fig.5.24b has a larger scale).



(a) Data overhead of Leo02's ROS 2 related processes for every fixed-size message

(b) Data overhead of the lander ROS 2 related processes for every fixed-size message

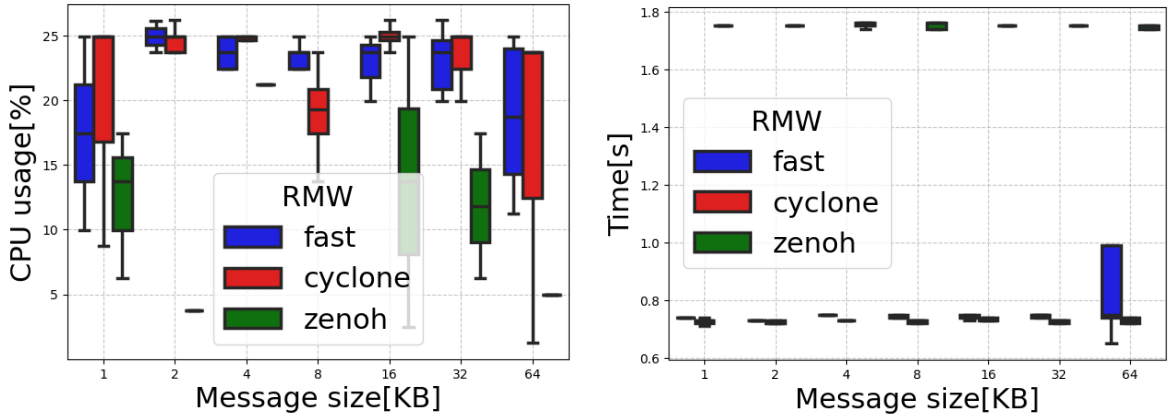
Figure 5.24: Box Plot representing the data overhead, for each RMW at every fixed size

Based on those results, Zenoh appears to add less overhead but also tends to be more predictable for all message sizes. In contrast, Fast and Cyclone tend to lead to more inconsistency and, therefore, more network instabilities.

5.5.10 CPU and RAM usage

In this section, we discuss the CPU and RAM usage of ROS 2 processes. It is important to emphasise the running processes on the rover. We only consider the packages designed for the experiment to be running (sender for Leo02 and receiver for lander), excluding even the ROS daemon. As an embedded system has a limited amount of resources, this metric allows us to quantify the impact of each RMW on the system. An RMW taking more resources leaves less room for other algorithms that might be running on the system, such as SLAM, navigation, or exploration. Furthermore, taking more resources has a direct impact on the power consumption of the embedded device and, therefore, the battery life of the robot.

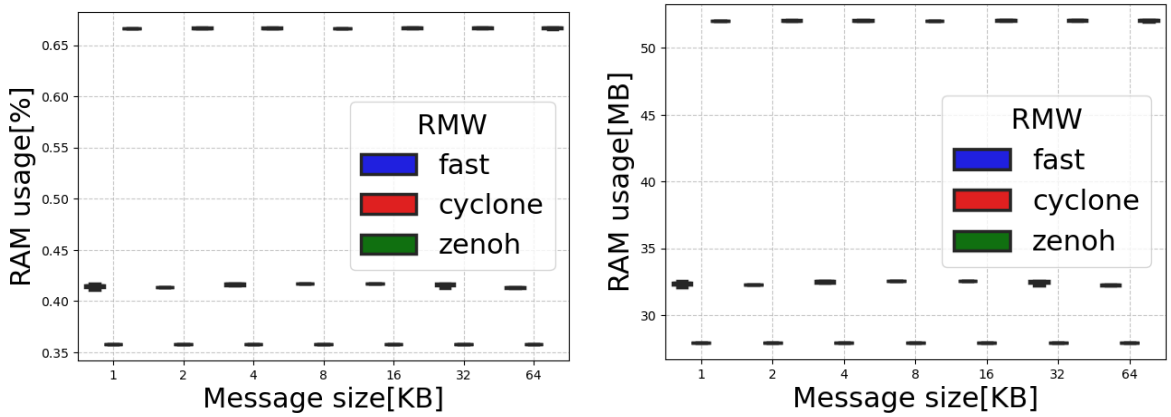
Fig.5.25a illustrates the CPU usage on the Raspberry of Leo02 when sending messages of fixed sizes for both scenarios, while Fig.5.25b displays the CPU time. Comparing both metrics shows interesting results; while the median value of Zenoh is around half the median value of FAST, it seems to take twice as long. This shows a behaviour that tends to be more lightweight for the embedded computers while being slightly slower to compute



(a) CPU usage in percentage on Leo02 (b) CPU time in second on Leo02

Figure 5.25: Box Plot representing the CPU usage in percentage

Notably, Zenoh uses a router that adds overhead to the overall resource consumption. However, even with this overhead, Zenoh depicts very good performances compared to the other RMW. While CPU consumption has a direct impact on the battery life and other algorithms' performance, RAM (Random access memory) is required to save data while computing. To this extent, Fig.5.26a displays the RAM percentage usage of ROS 2 processes (i.e. RMW consumption) while Fig.5.26b the bytes consumption of such process. Both



(a) RAM usage in percentage on Leo02 (b) RAM usage in 10^7 bytes on Leo02

Figure 5.26: Box Plot representing the RAM usage in percentage during the scenario

diagrams show the same data but use different representations (one as a percentage, the

other one in bytes) to get the RAM consumption's magnitude fully. Those diagram have very narrow box plots but depict a constant behaviour for each RMW. Starting from the left: Fast, Cyclone and Zenoh show the same amount of used memory no matter the size of the transmitted data which is really surprising for us. Cyclone and Fast use the least amount of memory with respectively 29.3MB and 34MB representing less than 0.45% of the 4GB of RAM of the Raspberry. On Zenoh's side, this number grows up to 54.6MB representing 0.66%. Zenoh takes more RAM than the other RMW, however it relies on a router to be functional. This router might add overhead to the initial 'cost' of sending a message on the network and therefore increase RAM consumption as it might have increased CPU consumption.

5.6 Discussion

Table 5.5 displays an overview of the performances of each middleware over different performance metrics. Symbols display how efficient each RMW is on a specific point of interest compared to the others. A "++" indicates that the system outperforms the others in the scenario, while a "- -" indicates major performance issues compared to the other approaches. The intermediary symbols show: "+" good results, "+/-" acceptable results and "-" less good results than the others. Finally, the "?" indicates missing values. As depicted earlier, Zenoh shows outperforming results compared to the other RMW.

For small messages, Zenoh outmatches the other RMW with smaller delays, better reachability, not too high data overhead per message, and reduced CPU usage while using slightly more RAM. However, the RAM usage is still within the acceptable margin. Medium messages still depict a good picture in favour of Zenoh with mitigated RAM usage. Large messages, also confirm the interest in using Zenoh. We recommend selecting the RMW based on the conducted experiment: if battery management and the reachability of the network are the key parameters, Zenoh might be more suited. On the other hand, if the data throughput is more important, Cyclone might be more interesting.

Table 5.6 emphasises the difference between Zenoh and the other RMW. Out of five metrics, Zenoh surpasses the other RMW in terms of delay, reachability, data overhead and

		Fast	Cyclone	Zenoh
Small Messages	Delay	-	- -	++
	Reachability	-	-	++
	Data overhead	-	+/-	+
	CPU Usage	+/-	+/-	++
	RAM Usage	+/-	+	-
Medium Messages	Delay	- -	-	++
	Reachability	-	+/-	++
	Data overhead	-	+/-	+
	CPU Usage	+/-	+/-	++
	RAM Usage	+/-	+	-
Larger Messages	Delay	- -	+	+
	Reachability	-	+	+
	Data overhead	+	+/-	+
	CPU Usage	+/-	+/-	++
	RAM Usage	+/-	+	-

Table 5.5: Summary table of RMW performances compared to each message size

CPU usage. On the other hand, RAM usage has increased, but the order of magnitude is derisory compared to embedded computer resources nowadays. On average, Fast offers comparable results, whereas Cyclone adds around 50% less data overhead to transmit a ROS 2 message over the network.

	Zenoh to Fast	Zenoh to Cyclone
Delay (reduced)	76%	69.86%
Reachability (increased)	146.93%	58.17%
Data overhead per message	-47.82%	-25.93%
CPU Usage (reduced)	41.27%	39.76%
RAM Usage (increased)	60.50%	86.03%

Table 5.6: Comparison table between Zenoh and the DDSs on global performances

A fixed communication delay cannot be avoided in the context of extra-terrestrial extreme environment exploration, forbidding any real-time operations and control. Any network-induced delay is negligible and less interesting for this scenario. However, data overhead,

reachability, and CPU usage are of high interest. Since the mesh network can offer limited performance (depending on the used protocol, hardware, and environment), it is designed to kick out any node that uses too much data throughput, so the selected solution should focus on optimising the data throughput usage and, thereby, the data overhead. Exploration rovers must execute various resource-expensive mapping and navigation software. Saving computational power should also be a concern in space-oriented scenarios. Also, computational power directly translates into electrical power, which is a critical resource in space. On the other hand, all of those metrics are pointless if the nodes in the network are unreachable. This is why we choose reachability as our key parameter regarding which RMW to choose.

In [154], the authors present similar results, where Zenoh stands out in the DDS implementation. They also highlight Message Queuing Telemetry Transport (MQTT) as the best solution, even if there is no known implementation compatible with ROS 2. It could be worth investigating an implementation with ROS. Yet, MQTT is designed for IOT and handling small messages and would probably have good performance in specific scenarios.

5.7 Summary

RQ-2: *How efficiently can heterogeneous Multi-Robot Systems communicate for space missions?*

This section presents a detailed evaluation of ROS 2 middleware implementations operating over a mesh network in a lunar analogue scenario. It explores a wide variety of scenarios, ranging from static LOS and NLOS to a dynamic topology-changing scenario. By replicating realistic conditions, including topology changes and noisy urban environments, we assessed the performance of each RMW across key metrics such as reachability, delay, data overhead, and resource usage.

The results emphasised Zenoh as a particularly promising candidate for communications over mesh networks. It demonstrated improved stability, significantly lower latency for small and medium-sized messages, and notable reductions in CPU usage, despite a modest

increase in RAM consumption. Zenoh also provided higher reachability, a crucial metric in scenarios where consistent network presence is critical. Depending on the need of a specific MRS, we provide clear guidelines on the choice of the middleware to use. These findings highlight the importance of middleware selection as a core design choice for robotic systems operating in extreme, decentralised conditions.

Contributions:

- We presented a realistic experimental setup evaluating ROS 2 middleware over a HWMP+ mesh network with dynamic topology.
- We defined and applied relevant metrics (reachability, delay, overhead, CPU/RAM usage) to benchmark RMWs in a planetary exploration context.
- We demonstrated that Zenoh consistently outperforms DDS-based solutions in scenarios with limited bandwidth and computational constraints.
- We proposed practical middleware selection guidelines for future MRS deployments relying on mesh networks.

Chapter 6

Network topology optimization

Related Publications

- Chovet, L., Kern, J. M., Bera, A., Santra, S., Olivares-Mendez, M. A., Yoshida, K. **"Robust Connectivity Maintenance for Heterogeneous Multi-Robot Systems for Planetary Exploration"**, *Under publishing process for the International conference on Robotics & Automation (ICRA)*, 2026.

† These authors contributed equally to this work.

"A system is never the sum of its parts; it's the product of their interaction."

– Russel Ackoff

6.1 Introduction

This chapter focuses on the following research questionBackPropagation

RQ-3: *How can the robots optimise the network topology?*

As seen in Chapter 4, the rise of the new space industry leads to future *Coopetitive* MRS scenarios. In *Coopetitive* scenarios, multiple robotic agents from various stakeholders such as governmental institutions and private companies— operate on the same network. These agents can cooperate for mutual benefit while simultaneously remaining in competition for limited resources or contracts. The work presented in the chapter led to the design of a **Make**

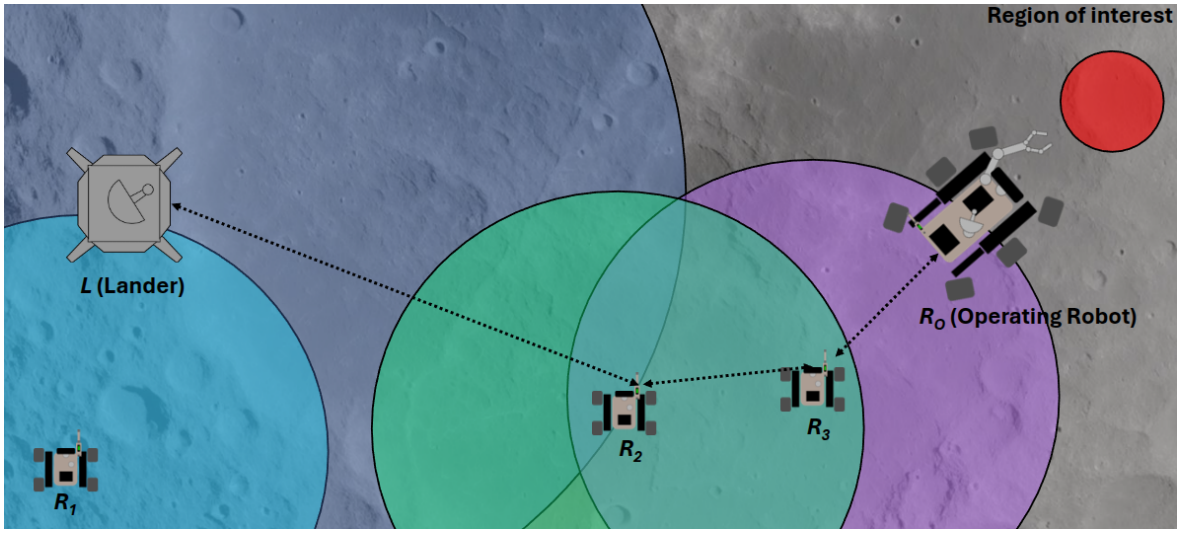


Figure 6.1: Architecture of the MRS. The red circle represent a region of interest where R_o should perform a science task. Each circle represents their communication range.

or Buy Architecture for a MRS, where any external client can request various services such as mapping, hyperspectral imaging, or in-situ processing. Robots from different organisations would compete to get the contract, leading to a more efficient and economically optimised approach for decentralised MRS coopetition.

However, such a robotic system requires the appropriate network architecture to work. In Chapter 5, we studied the use of the Mesh network topology as a solution for efficient communication. In a meshed network, each agent of the network can act as a relay point, leading to a decentralised system, resilient to single-point failure. The work presented in the chapter led to a specific implementation developed for space *Coopetitive* MRS missions. It offers a mesh network with a specific ROS 2 implementation, highlighting Zenoh as the most reliable middleware for communication over mesh networks.

In the full *Coopetitive* scenario, if a robot were sent to perform a task outside the communication range of any robot as in Fig. 6.1, it would get disconnected and lost. In order to never get lost, other robots need to adjust their position, acting as a relay and ensuring the overall system never gets disconnected.

This challenge is called **connectivity maintenance** and has been widely studied in

general robotic systems [85]. However, very few works have applied it to planetary exploration scenarios [172] where terrain, delays and mission-specific data rates add complexity.

In order to decide the optimal placement of each robot, a precise understanding of radio propagation in extraterrestrial environments is required. This behaviour is analysed through radio propagation modelling, which is a well-established field of study. Many propagation models exist [173], using the appropriate model depends on the specific environment and conditions. In the case of planetary exploration, studies such as [91] identified the deterministic two-ray model as the most accurate radio propagation model for lunar conditions.

Our work focuses on optimising the placement of a fleet of relay robots to ensure robust connectivity between a lander and an operating robot, with particular emphasis on mission-specific data rate requirements. The main contributions of this chapter are as follows:

- **Proposal of a novel approach** for strategic placement of relay robots. Ensure robust and efficient connectivity in planetary exploration missions.
- **Validation of the approach through simulations** against a state-of-the-art method, demonstrating its effectiveness in maintaining connectivity while optimising energy consumption.
- **Demonstration of our approach portability** to a physical heterogeneous MRS in a real-world scenario, highlighting its applicability to the constraints and limitations of real-world systems.

6.2 System Modelling

6.2.1 Scenario Description

We consider a lunar MRS with the goal of analysing resources in a given area of the Moon in the context of an ISRU mission. This mission requests to reach and analyse a given region of interest (ROI) for future ISRU perspectives. As detailed in Fig. 6.1, a lander is deployed at the site and relies on a team of robots to carry out the mission. The robotic team consists of a lander L, the science robot R_o, which is responsible for scientific analysis tasks and a

set of n scouting robots $R = \{R_1, R_2, R_3, ..., R_n\}$ (as shown in Fig. 6.1). We consider the set of agents as $A = \{R_o, L, R\}$. The data available on the Moon's explorability remains limited in precision, and the robots will have to adapt to the local conditions. In this scenario, we assume the availability of high-precision positioning similar to Real-Time Kinematic (RTK), enabling decimetre-level localisation for all agents. While the Moon lacks a native Global Navigation Satellite Systems (GNSS) infrastructure, such accuracy can be achieved through local positioning systems deployed from the lander or orbital assets, as discussed in [174] L is in charge of ensuring the connection to the earth, potentially through a satellite, but is also responsible for staying connected to the rovers. It is possible that R_o can travel beyond the direct communication range of L. However, it needs a permanent connection with L in order to ensure efficient telemetry along with offloading the collected data to L for further processing. To ensure a stable connection, the robots from R have the capability of extending the network through robot-to-robot communication. This is done through a meshed network using a protocol such as HWMP+ [82]. This approach operates on the OSI layer two, the data link layer. It is distinct from typical Layer 3 routing protocols, such as OLSR or AODV, which operate at the IP-level addressing. In the described scenario, the objective is to find the right set of positions for each of the relay rovers to maintain a stable connection between R_o and L. To formally represent this connection, we define the **link path** (LP) as the ordered sequence of agents that form the communication chain from R_o to L.

In the example shown in Fig. 6.1, the black arrows indicate the individual communication links between neighbouring agents. Together, these links form the link path, which in this case is represented by the set: $LP = \{R_0 \rightarrow R_3 \rightarrow R_2 \rightarrow L\}$. This sequence illustrates how the communication signal is passed through a series of intermediate relay rovers (in this case, R_3 and R_2) to maintain connectivity between R_o and L.

6.2.2 Radio Propagation modelling

At first, understanding the behaviour of radio waves is essential to determine the optimal robot placement for effective communication. Radio propagation modelling, extensively studied in literature [173], addresses this by characterising radio wave behaviour. Nevertheless,

selecting the appropriate radio wave model depends empirically on the specific scenario.

The work in [91] validated a deterministic radio propagation model for a MRS in the context of planetary surface exploration. It relied on a high-resolution terrain map and estimated the radio path loss through three propagation phenomena using Fresnel zones: free space, reflection, and diffraction. They validated the model with real-world experiments, comparing their predictions with measured data. The experiments showed promising results but highlighted that higher antenna placements caused more reflection than the model was capable of predicting. With the necessary adjustments and determining the reflection points, the model could be applied to communication-aware path planning for moving rovers on any planetary surface.

Staudinger et al. [92] introduced a space network coverage prediction framework using satellite DEMs, integrating Bayesian map fusion with path loss, data rate, path planning, and semantic annotation modules.

The radio propagation model takes the LOS component and ground reflection component into account, commonly referred to as the two-ray ground reflection model [173]. For signals above 1Ghz, refraction can be neglected in a purely hilly terrain with no buildings or sharp objects. It showed that certain distances between Tx and RX imply destructive interference and should be avoided. The receiver antenna height played a major impact on spatial interferences. In conclusion, the authors of [91] highlighted the need for a relay robot and noted that the data rate for the best relay would potentially be below the direct link scenario.

Knowing that our scenario is based on the Moon with no impacting features such as buildings or sharp objects, the study offered by [91] confirms that the two-ray ground reflection model is suited for a mobile multi-robotic network. As described in [173], the received power P_r from a transmitter at the distance d is expressed as:

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \quad \text{as long as} \quad d \gg \frac{20 h_t^2 h_r^2}{\lambda} \tag{6.1}$$

with h_r and h_t being, respectively, the height of the receiver and transmitter, λ the wavelength of the signal, d the distance between the receiver and the emitter, G_r and

G_t respectively, the gain of the receiver and transmitter, and P_t the power emitted by the transmitter.

Getting the data rate C of the communication lies on the Shannon-Hartley theorem [175]:

$$C = Blog_2(1 + SNR) \tag{6.2}$$

with B the bandwidth of the channel and SNR the Signal-to-Noise Ratio such as:

$$SNR = \frac{P_r}{P_{noise}} \quad \text{with} \quad P_{noise} = kTB \tag{6.3}$$

with k the Boltzmann's constant and T the temperature of the environment.

6.2.3 Problem Statement

The optimal placement of relay rovers to maintain connectivity between a lander L and a science rover R_o can be formulated as a **constrained network optimisation problem**. The primary objective is to ensure uninterrupted communication between R_o and L, even when R_o operates beyond direct line-of-sight.

The connectivity of the robot R_i is preserved if the received power P_{r_i} between each consecutive pair on the link path LP remains above a minimum threshold P_{th} typically [87]:

$$P_{r_i} \geq P_{th}, \quad \forall i \in LP$$

In planetary scenarios, power is tightly constrained. Each robot R_i allocates only a portion of its energy budget to communications, capped by its available transmission power P_{att_i}, leading to:

$$P_{t_i} \leq P_{att_i}$$

Since movement is one of the most energy-expensive operations, we use the total displacement from the initial position of each robot (x_i^0, y_i^0) to its assigned relay position (x_i, y_i) as a proxy for energy consumption, which yields the following optimisation problem.

$$
\begin{aligned}
\min_{(x_1,y_1),\ldots,(x_N,y_N)} \quad & \sum_{i \in LP \setminus \{L, R_o\}} D((x_i, y_i); (x_i^0, y_i^0)) \\
\text{s.t.} \quad & P_{r_i} \geq P_{th}, \quad \forall i \in LP \\
& P_{t_i} \leq P_{att_i} \\
& x_{\min} < x_i < x_{\max} \\
& y_{\min} < y_i < y_{\max}
\end{aligned}
\tag{6.4}
$$

Here, N is the number of agents in the active chain LP. If no solution satisfies the constraints, another relay must be added to LP. The optimisation is repeated until feasibility is achieved.

Since the problem aims to jointly optimise robot connectivity, energy constraints, and motion planning, it falls into a class of combinatorial optimisation problems that are NP-hard [176]; hence, the next section presents a decentralised heuristic to compute feasible solutions in real-time.

6.3 The BackPropagation approach

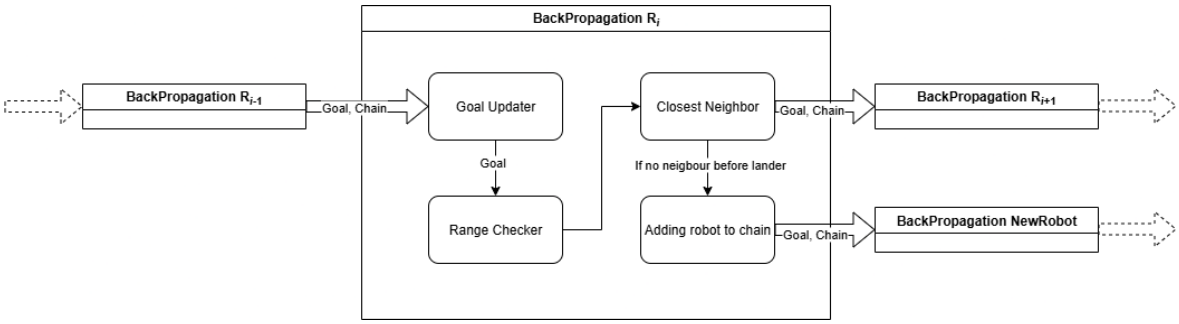


Figure 6.2: Solution flow of the BackPropagation algorithm for a relay robot R_i, R_{i-1} corresponding to the previous robot in LP, and R_{i+1} to the following one.

As detailed in our previous research in chapter 5 and other studies[92], relay is only useful when the receiving robot reaches the edge of the emitter range. It is crucial to minimise the number of relays to extend the exploring area, as too many relays could impact network capacity. When R_o receives a new goal, the entire network should adapt to ensure the

permanent connectivity of R_o with L. We propose the Algorithm 3 as a solution and call it the **BackPropagation** (BP) algorithm. This approach allies knowledge from the **Graph-theoretic, LOS and environment aware** methods. The BackPropagation relies on the link path LP as the sequence of robots between R_o and the L. This solution tackles the problem of 6.4, especially if the equation is not solvable for a given LP, it adapts and adds a robot to it.

To use the minimum number of relays, the distance between each robot of LP should be maximised before adding a new robot to LP.

Algorithm 3 BackPropagation Algorithm

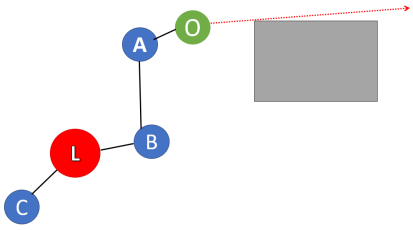
Input:
1. *goal* ▷ target position
2. *LP* ▷ list of relay robot names

Output: Optimised goal for each relay robot to ensure connectivity

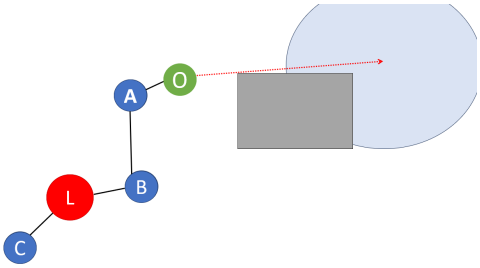
procedure BACKPROPAGATION(*goal*, *LP*)
  if $|LP| > 0$ **then**
    $goal \leftarrow$ GOALUPDATER$(goal)$
  end if
  if RANGECHECKER(goal) **then** ▷ return true if the goal is in range of a networked robot
    BackPropagation terminates; navigation to $goal$ is triggered on every robot in LP
  else
    $LP \leftarrow LP \,\|\, r$ ▷ append current robot r
    $nb \leftarrow$ CLOSESTNEIGHBOR
    if $nb = L$ **then** ▷ next up in LP is L; need to insert relay
      $n_r \leftarrow$ ADDINGROBOTTOLP
      On n_r, call BACKPROPAGATION(goal, LP)
    else
      On nb, call BACKPROPAGATION(goal, LP)
    end if
  end if
end procedure

As detailed in Fig. 6.2, for a robot $R_i \in C$ the algorithm 3 relies on four components:

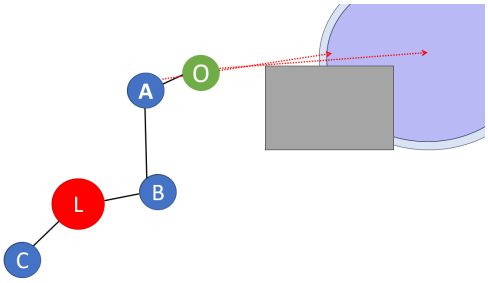
- **Goal Updater**: This function is called if the robot is not the initiator, R_o. Using Eq. (6.2), it computes a position from which R_i can connect to the previous robot R_{i-1} with sufficient bandwidth. Since valid positions lie on a circle around R_{i-1}, the algorithm selects the closest one to R_i that both (i) has line-of-sight to R_{i-1}, and (ii) is reachable based on the DEM and global map.



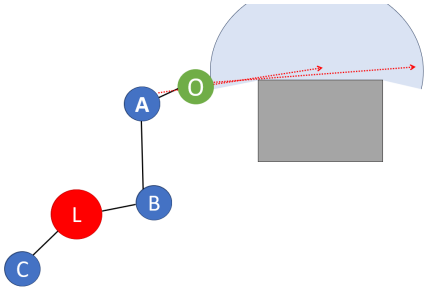
1: Initial scenario where R_o receives a goal



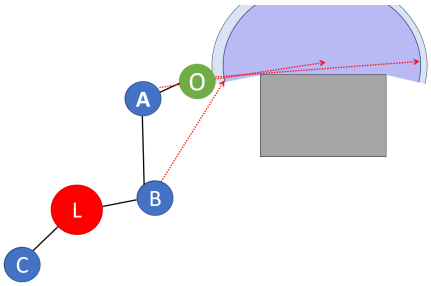
2: The goal would put R_o outside of any robot communication range



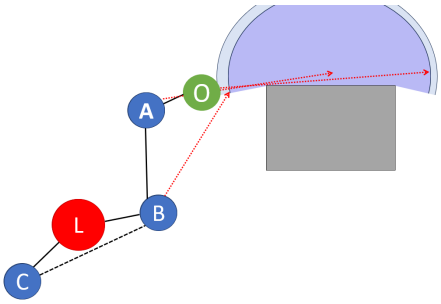
3: This information is sent to the closest robot toward L: A, which computes a new goal



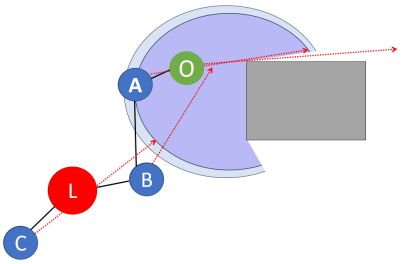
4: A triggers the BackPropagation with the newly computed goal as its goal



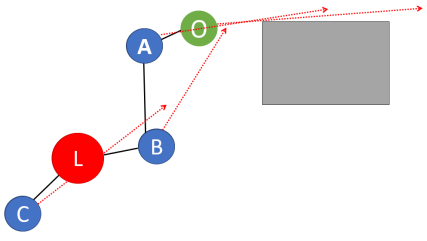
5: B is the closest, and computes a new goal to be in A's range



6: B triggers the BackPropagation with the newly computed goal as its goal, there is no other robot between B and the lander, a new link is added C



7: C computes its new pose to be in B's range



8: C's new pose is in the lander range, the BackPropagation succeeded, all robots can move toward their goals

Figure 6.3: Exemplary scenario and full run of the BackPropagation algorithm

- **Range Checker**: This function checks if moving to the goal put R_i outside of the communication range of R_{i+1} using Eq. 6.1
- **Closest Neighbour**: This function checks for the closest robot from the goal, with fewer relays to L. Mesh networks provide features to know the network architecture. If the closest relay is the lander L, the BackPropagation stops, and all the robots starts moving toward their goals. If there is a robot R_{i+1}, it triggers the BackPropagation on R_{i+1}. If there is no robot, the next function is called.
- **Adding Robot to** LP: This function searches for the closest robot that is not in LP and calls the BackPropagation on it. It also checks if any robot depends on this one for its connection. If this is the case, the dependent robots move toward L connection zone.

The future pose is sent to the closest relay, which will adjust its position. Then, if this relay should leave the connectivity zone, it asks its closest relay to adjust its position. The process continues iteratively for each relay until reaching the L. If the last relay R_l cannot connect to the lander L and the rest of the chain, it will search for the closest robot to the chain **Adding Robot to LP** function. This is done by computing the Euclidean distance from R_l to each robot. The robot with the smallest distance is then added to the chain and will move at the optimal communication distance of R_l along the shortest route. At each call of the BackPropagation algorithm, the argument passed is the goal of the previous robot and the robots in the chain where the algorithm has already been applied. The BackPropagation algorithm is run on every robot, making the system fully decentralised and scalable. It adapts to any robot joining or leaving the system. It also comes with a fault tolerance feature. If a robot faces a disconnection, it remains in control of its motion capabilities and will move back to its last known position with a connection. The time complexity of the BackPropagation algorithm is $O(n^2)$.

Each robot has its own antenna datasheet that contains the antenna's and robot's characteristics. Those characteristics are listed in Tab. 6.1. Robots will adjust their computation based on both the transmitting and receiving antenna's datasheets, following the two-ray model (see Eq. 6.1). This allows for example to acknowledge the difference between a large

operating rover, with a higher antenna, to a smaller scout rover with a lower antenna, leading to less coverage. Using the datasheets, each robot identifies the lowest maximum speed among those in the link path (LP) and limits its own speed accordingly. These parameters allow each robot to determine the slowest robot in the link path and adjust its behaviour accordingly.

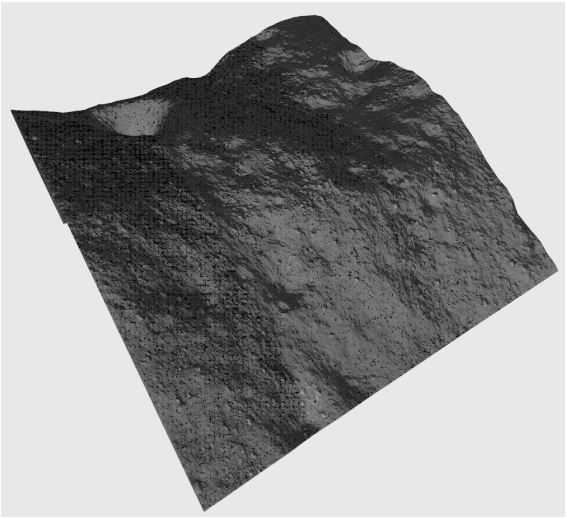
Table 6.1: Antenna Model Parameters

Parameter	**Description**	**Unit**
Power_emitted	Transmission power	W
GainTransmitter	Transmitter antenna gain (linear)	—
height_transmitter	Height of the transmitter antenna	m
desired_rate	Desired data rate	mbps
theoritical_to_real_factor	Adjustment factor from theoretical to real signal strength	—
wavelength	Signal wavelength, for 2.4Ghz network: 0.125m	m
threshold	Minimum required signal power (sensitivity threshold)	dBm
DEM_resolution	Digital Elevation Map resolution	m/pixel
B	Boltzmann constant	—
T	Temperature (used for noise calculation), in the case of lunar activity, -273.15 [1]	K
dem_path	Path to the DEM raster file	file path
chunk_size_x	Width of DEM chunk	m
chunk_size_y	Height of DEM chunk	m
offset_x	X offset applied to the DEM coordinates if the lander is not in the centre of the DEM	m
offset_y	Y offset applied to the DEM coordinates if the lander is not in the centre of the DEM	m

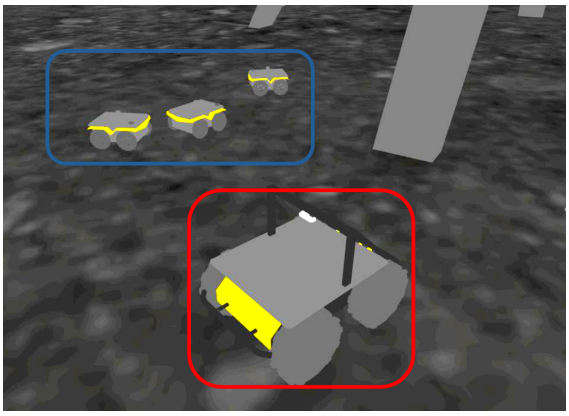
In order to have the best estimate of network connectivity, the robots rely on DEM of the landing site. This allows, by ray tracing, to see if direct line of sight communication is possible between both robots. Since the provided DEM could be not centred on the landing site, *offset_x* and *offset_y* are used to recenter the position of the lander on the 0,0 of the map. Also, as the DEMs used are at least 16km by 16km, we keep only a specific chunk of them, whose dimension is stated in the appropriate variables.

The constraints of the scenario are stored in the data, so it is directly possible to state the

[1] Lunar surface temperature reference value from https://www.diviner.ucla.edu/science



(a) 3D model simulated of the Site1 DEM.



(b) Screenshot of the simulation, featuring the operating rover R_o in red and the three scout rovers R_1, R_2, R_3 in blue.

Figure 6.4: The generated 3D environment along with a view of the robots in the simulator.

desired data rate. Finally the *theoritical_to_real_factor* variable is a float constant in $\{0, 1\}$, it is an empirically value decided to represent the difference between the computed "ideal" data rate value, compared to the real one.

6.4 Simulated Experiments

6.4.1 DEM and Virtual Environment

In order to evaluate our proposed framework in conditions representative of lunar surface operations, we created a high-fidelity virtual environment based on real lunar topography data. Specifically, our simulation leverages a DEM of the Moon, a raster dataset where each pixel encodes surface elevation, essential for realistically simulating rover mobility, line-of-sight communication, and energy consumption across uneven terrain.

Most of the existing DEM of the Moon has been realised thanks to the "Lunar Orbiter Laser Altimeter (LOLA)" [177] onboard the Lunar Reconnaissance Orbiter mission by NASA. The polar regions are of greater interest, and the authors of [178] have provided the DEM of the high-priority sites in the south pole of the Moon, offering a quality of $5\,\mathrm{meters/pixel}$.

Our study simulates a lunar environment based on **site one** [149] (see Fig. 6.4) and adds some rocks of a size below five meters, to simulate obstacles not visible on the DEM. The centre of the environment is set at X=$-11\,\mathrm{km}$ and Y=$-12\,\mathrm{km}$ in the south polar stereographic (SPS) coordinate system, in order to offer a zone flat enough for landing and multi-robot exploration. It features an elevation ranging from $-523.18\,\mathrm{m}$ to $1959.5\,\mathrm{m}$.

The parameters of the simulation are listed in Tab. 6.2, while the settings of each antenna are in Tab. 6.3. These parameters are set to reproduce a $2.4\,\mathrm{GHz}$ Wi-Fi antenna, the same as used in real-world experiments.

Table 6.2: Simulation Parameters

Site	Site 1 of [178]
Dimensions	$11\,086\,\mathrm{m}$ by $11\,086\,\mathrm{m}$
Gravity	$1.62\,\mathrm{m/s^2}$
Elevation	$-523\,\mathrm{m}$ to $1959.5\,\mathrm{m}$

Table 6.3: Antenna Network Parameters

Emitted Power	0.063W
Gain	0.519
Height	0.3m
wavelength	0.125

Table 6.4: Simulated robots parameters

Robot	R_o	R_1	R_2	R_3
Type	Husky [179]	Jackal [180]	Jackal [180]	Jackal [180]
Maximum Speed	1.0m/s	2.0m/s	2.0m/s	2.0m/s
Sensors	RGB-D	2D lidar	2D lidar	2D lidar
Function	Science Rover	Scout and Relay	Scout and Relay	Scout and Relay

A DEM such as the one at site 01 represents a zone of $11 \times 11\ Km^2$. To evaluate our approach, we subdivide the site into various subsets, taking some with enough interesting features, such as the one featured in Fig.6.4a. This subdivision allows for a realistic scenario while optimising the simulation. After normalisation, the heightmap is passed to the simulator that can generate a world such as the one in Fig.6.4a.

The simulation used for the experiment is based on a simulation framework from the Clearpath Robotics company [181]. It is modified to match our lunar environment with a heterogeneous MRS as described in Tab. 6.4.

The simulation features four robots. The first one is a Husky A200 [179], the biggest robot,

offering a larger payload size and dimensions. This robot is used to represent R_o and carries an RGB-D camera in order to map its environment and gather the more data. The three other robots are Jackals J100 [180], a smaller but faster robot, perfect for scouting of acting as relays. They are equipped with a 2D lidar for obstacle avoidance and simple mapping.

6.4.2 Benchmarking approach: FBA

The FBA is based on the approach presented in [88] together with some enhancement from [94]. In this approach, every robot is considered a node in a graph, and every link a spring. Depending on the optimal distance OR calculated by the antenna, the force f applied by a node N_A to a node N_B whose distance is d, the following:

$$f(d) = \begin{cases} \mathsf{k}_a * (\frac{1}{d^2} - \frac{1}{TR^2}) & \text{if } d > OR \\ \mathsf{k}_r * (\frac{1}{OR^2} - \frac{1}{d^2}) & \text{if } d \leq OR \end{cases} \tag{6.5}$$

with k_a an attractive constant, k_r a repulsive constant, and TR the estimated transmission reach. The two constants k_a and k_r influence the balance between the attractive and repulsive force. If $k_a > k_r$, the attractive force dominates, leading to a more robust and stable network. In the opposite case of $k_r > k_a$, the repulsive force prevails, improving the system adaptability to changes but increasing the risk of connection loss. Once all the forces applied from the neighbouring nodes are applied to a node, the resulting force is sent to the robot as a navigation goal. In order to ensure a proper comparison, the FBA approach will be run with three different sets of constants. This approach is used for a group of robots appearing among other robots, to establish a connection. While this is not exactly the same scenario, this approach is still applicable as a useful point of comparison. In order to adapt it to the scenario, we add a force coming from L and R_o on each of the rovers. If a rover gets disconnected, it keeps the same last force as before, which should lead to driving toward the ”biggest mass of robots”.

Table 6.5: FBA Parameters

Set	k_a	k_r
FBA_1	8000	8000
FBA_2	16000	8000
FBA_3	8000	16000

6.4.3 Performance Metrics

In order to evaluate our approach, we decided to compare the following three metrics:

- **Reachability**: In a single experimental run, the reachability is a Boolean value measured over time, being true if L is able to connect to R_o through the network. In the overall experiment, the reachability is the average of these values. The reachability displays how often the robot is disconnected from L. The higher the reachability, the higher the *reliability* of the system.
- **Capacity**: The capacity is the estimated network throughput between L and R_o in MB/s. The system must respect the threshold set by the user.
- **Total Distance**: The total distance travelled by every robot in the system. As movement is the most expensive task for robots, the goal of any approach is to limit the cost of the system. The smaller the total distance travelled, the more energy optimal the system is.

6.4.4 Experimental Protocol

- The fleet of robots appears at fixed positions around L in such a way that the network graph is fully connected.
- Then R_o receives a list of ten random waypoints to explore, reproducing a list of points of interest for the mission. This list of ten random is called a Scenario. Each of the waypoints is set less than a hundred meters away from the previous one, and no point is generated more than four hundred meters from L, considering the estimated maximum range of four antennas.

- The robot R_o has six hundred seconds to explore as many waypoints as possible.
- Each of the Relay robots gets attributed a connectivity maintenance algorithm depending on the runBackPropagation BackPropagation or FBA with one of the parameters sets

This represents one operational run. For each Scenario (set of 10 waypoints), ten runs are done for the BackPropagation approach but also to the FBA approach for its three sets of parameters, to ensure statistically sound results.

6.4.5 Results

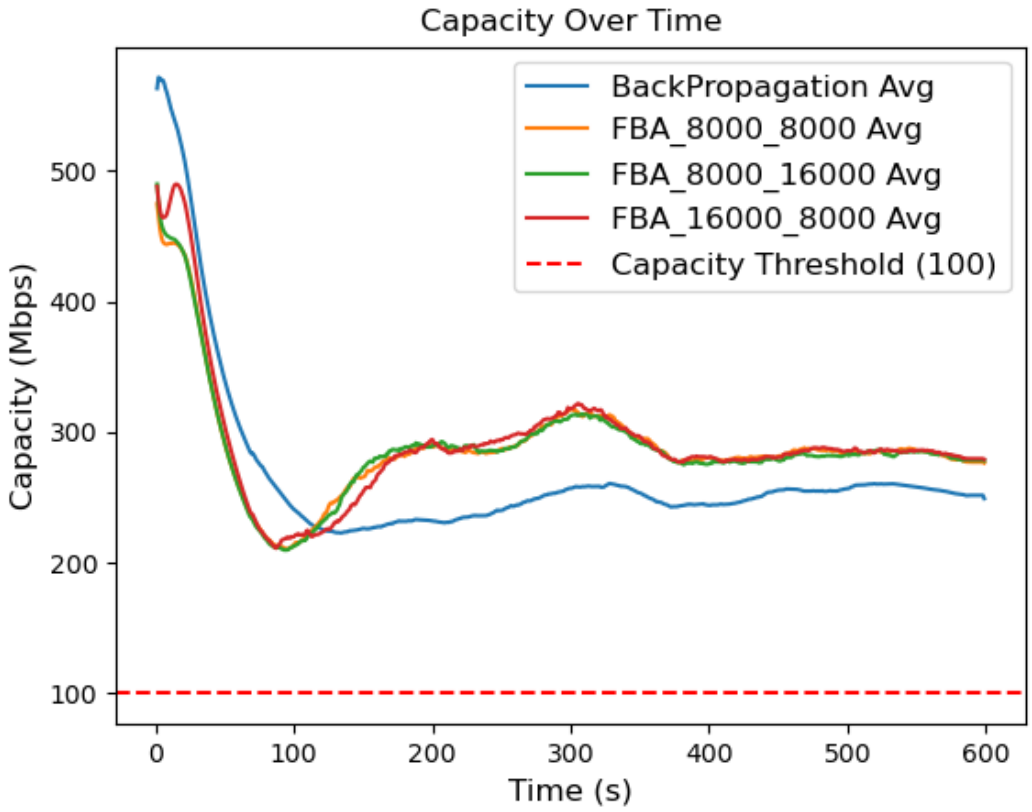


Figure 6.5: Estimated network capacity in MB/s over time for each approach between L and R_o. The red bar indicated the required threshold. All approaches never go below the expected threshold

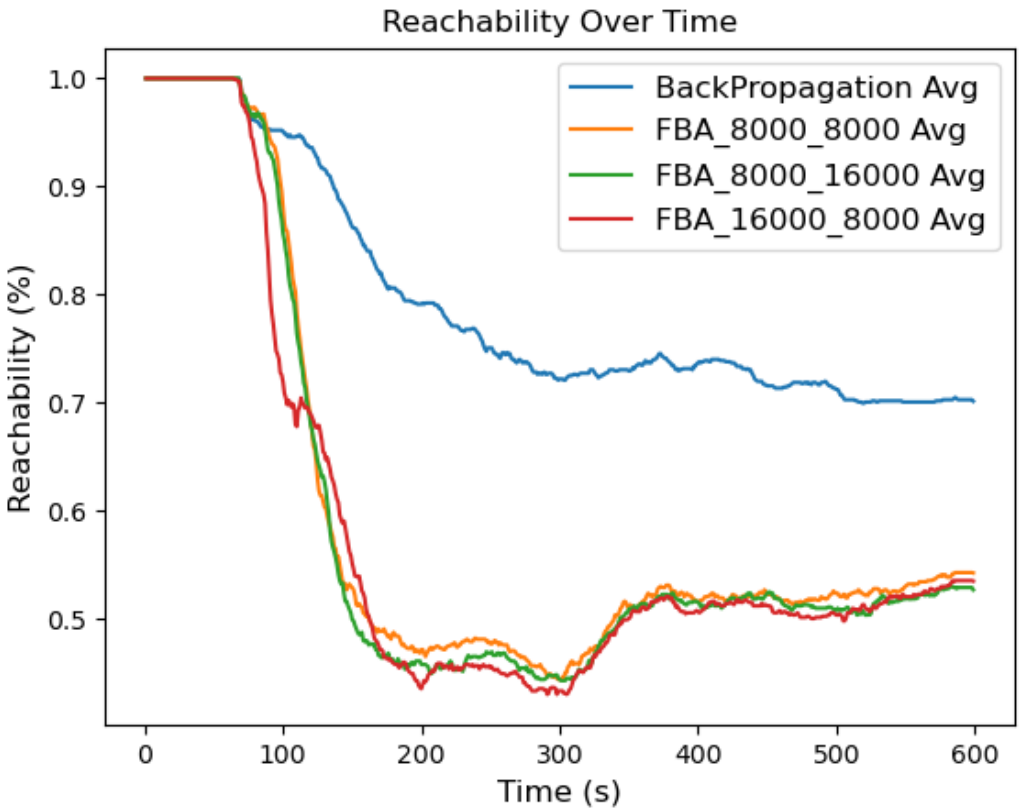


Figure 6.6: Average connectivity between 1(fully connected) and 0(no connection) over time for each approach between L and R_o. Our approach (blue) is facing 61% fewer disconnection events than the benchmarked approaches.

We conduct the experiments on 94 distinct sets of waypo.6.5, 6.6, and 6.7 show the evolution of the average results over time. Table 6.6 summarises the key performance metrics averaged over the 94 sets for each approach.

Fig. 6.5 illustrates that the BackPropagation approach consistently achieves a lower estimated bandwidth than the Fig. methods, although the values always remain well above the $100\,\mathrm{Mb/s}$ threshold (at least twice as high). This reduced bandwidth usage reflects the nature of the BackPropagation strategy, which minimises unnecessary robot movement by targeting new goals only when needed.

In terms of connectivity, as shown in Fig. 6.6, BackPropagation maintains superior

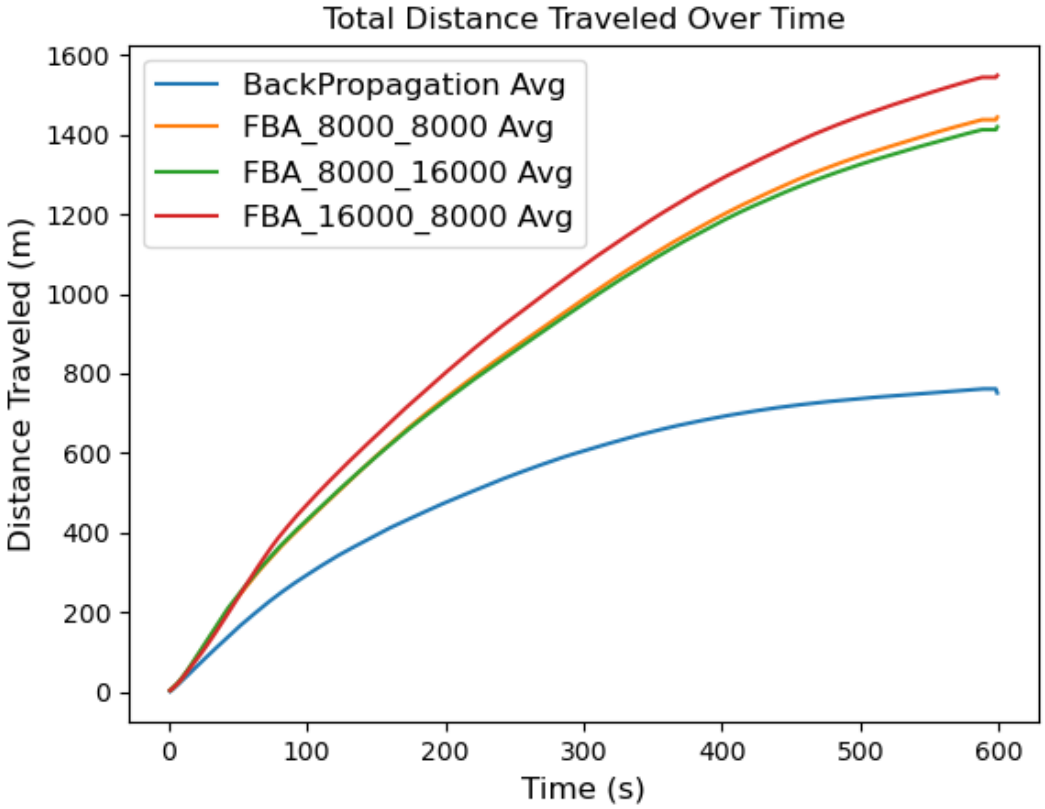


Figure 6.7: Total distance traveled by every robot in meters over time for each approach. Our approach (blue) is driving way less than the benchmarked approaches, leading to less energy consumption

performance, with R_o experiencing approximately **61% fewer disconnection events** than the average of the three FBA approaches. Specifically, BackPropagation reduces disconnection events by **57.8%** compared to FBA_1 (1.21 events on average), **65.8%** compared to FBA_2 (1.49 events), and **59.5%** compared to FBA_3 (1.26 events). This improvement comes from the fundamental difference in behaviour between the approaches. FBA is purely reactive, responding only to local conditions, while BackPropagation is proactive, predicting optimal relay positions and incorporating fail-safe mechanisms to maintain connectivity.

Regarding navigation efficiency, Fig. 6.7 reveals that BackPropagation travels roughly half the total distance of the FBA methods when performing similar tasks. This difference is

Table 6.6: Average metrics results over all the experimental runs.

Metric	BP	FBA_1	FBA_2	FBA_3
Disconnection events ↑	**0.51**	1.21	1.49	1.26
Total distance traveled ↑	**750 m**	$1443\,\mathrm{m}$	$1548\,\mathrm{m}$	$1419\,\mathrm{m}$
Tasks achieved ↓	1.86	**2.65**	**2.67**	**2.65**
Distance per task ratio ↑	**403 m**	$544\,\mathrm{m}$	$579\,\mathrm{m}$	$535\,\mathrm{m}$

partially due to BackPropagation's longer computation time per step, which results in about **30% fewer tasks completed** overall. In simulation, this computation time remained below one second, although it could not be precisely measured due to the 1 Hz logging frequency. However, this duration is negligible compared to the typical timescales involved in space missions, where planning and actuation often occur over minutes or hours. In the context of planetary missions, time is a less critical factor, but it remains important to note that this approach would be less suited for a time-critical scenario. Nevertheless, when considering the distance-per-task ratio, BackPropagation is more efficient, covering **27% less distance per task**. This improvement comes from BackPropagation's ability to strategically position relays in advance, reducing unnecessary movement. In contrast, the FBA approach requires each robot to constantly adjust its position, resulting in additional and often unnecessary movement. Surfaces in space such as lunar regolith can be very abrasive. Therefore, minimising movement can significantly reduce the risk of mechanical failures.

Since motors are typically the most energy-intensive components of a robot, these results suggest that the BackPropagation method is a more cost-effective solution in terms of energy consumption. Additionally, surfaces in space, such as lunar regolith, can be very abrasive; minimising movement can significantly reduce the risk of mechanical failures.

6.5 Real-World Experiments

The following experiment was performed during my three month research stay in the Space Robotics Lab at the University of Tohoku in Japan.

6.5.1 Multi-robot team setup

This section highlights the work performed during the three-month stay at the university of Tohoku. In order to be ready for the experiment, the following tasks were expected to be performed.

- Configure all the robots with ROS 2
- Implement the mesh network on the MRS
- Set the **BackPropagation** approach on the MRS
- Ensure portability and proper behaviour

We use a MRS consisting of robots: EX1 [182], Moonraker[183], and Clover [184], developed at the Space Robotics Lab of Tohoku University [2]. Each rover has distinct capabilities, equipped with sensors and hardware specifications suitable for a specific range of tasks as described in the Tab. 6.7. *EX1*, a high-speed exploration rover, is capable of traversing lunar terrain at velocities of up to 1 m/s. To ensure both stability and manoeuvrability, EX1 employs the Ackermann steering and a rocker mechanism. Together with an active spring-damper system, this improves the rover's tractive performance on loose soil. *Moonraker*, developed for the Google Lunar XPRIZE, is made from lightweight carbon fibre materials and high-quality hardware that allows for precise locomotion. It is a four-wheel differential drive rover. *Clover*, the third rover of the MRS, is a modular rover platform designed for autonomous navigation, SLAM and multi-rover collaboration tasks. It is a four-wheel differential drive rover.

The mesh network is set using the MikroTik Groove A52 ac routers connected to each robot, according to a related work in [185]. Each router is parametrised according the settings in Tab. 6.3 and runs the HWMP+ protocol on the data link layer.

In order to work properly, each robot was required to fulfill the following requirementsBack-Propagation

- A full ROS 2 communication stack, in order to ensure that all the robots can act as a single MRS

[2] https://astro.mech.tohoku.ac.jp/e/

Table 6.7: Real robots parameters

Robot	R_o	R_1	R_2
ROS version	ROS + ROS2	ROS2	ROS2
Speed	$1.0\,\mathrm{m/s}$	$0.2\,\mathrm{m/s}$	$0.1\,\mathrm{m/s}$
Steering Model	Ackerman	Differential Drive	Differential Drive
Onboard computers	Intel Nuc	Intel Nuc	Jetson TX2
Sensors	Intel Realsense D345	Intel Realsense D345	Intel Realsense D345
Localisation	RTK + Intel realsense T265	RTK + Intel realsense T265	Leica Total Station + Intel realsense T265
Function	Science Rover	Scout and Relay	Scout and Relay

- Localisation: In order to apply the **BackPropagation** approach, each robot is required to have its own position along with that of the other robots
- Navigation: The robot would need to be able to move to the goals generated by the BackPropagation.

The state of each robot at the beginning of the work is detailed in the Tab. 6.8 and the way to assess each requirement is described on the following subsections

The mesh network was set as described in section 5.2, using 5Ghz network in order to reduce the reach of the antenna, but also to follow the same parameters as the simulated experiment.

To optimise work and facilitate reusing previous work, each implemented functionality was done within a Docker container. A Docker container allows the deployment of a specific functionality in a fixed environment, ensuring to minimise compatibility issues. This would simplify to port the work done on any robot to another.

ROS 2 compatibility

As detailed in Tab. 6.8, Moonraker already contains a full ROS 2 stack, however the other two robots were still using a full ROS middleware stack.

[3]Even if the fully autonomous navigation was implemented on the Moonraker, it was highly limited by the embedded computer. It would require a large improvement for efficient behaviour

Table 6.8: Robots used in the real-world experiments and their capabilities at the start of research stay

Robot	Image	ROS 2	Localisation	Navigation
Moonraker		Yes	Partial [3]	Partial[3]
Clover		Fully ROS	None	None
EX1		Fully ROS	None	None

For the **EX1** and **Clover** sensors, ROS 2 drivers were installed in Docker containers, mainly based on the work carried out for the Space Resources Challenge.

The only part requiring additional work was the motor driver for both robots. As they use proprietary material, the ROS driver needed to be kept. We implemented a Docker container with a custom ROS to ROS 2 bridge, made to only bridge the necessary messages. In this way, only the twist commands would be sent from ROS 2 to the ROS motor driver, and the motor driver would send the wheel odometry back.

This work on the communication between the motor drivers and ROS 2 led to the discovery of technical issues in both drivers. In the case of EX1, it was sending the four-wheel odometry reading as soon as receiving one, leading to incorrect messages as every values would get reset after sending. For Clover, some pins were wrongly assigned, leading to failing behaviour only when the robot was moving back at a certain speed. All of these issues were fixed, ensuring proper behaviour and a fully teleoperable system.

Localisation

Despite having an implemented SLAM approach, Moonraker was unable to use it in a real-world experiment. Indeed, as soon as the approach was launched, the robot would instantly crash. The cause was the embedded computer being an Intel NUC, it would lack a GPU. The GPU is highly needed for SLAM approaches, leading us to design a newer approach.

The localisation of a robot corresponds to two main data, its position and its orientation.

Getting the orientation of a robot can be done through various means. In the case of this experience we used an Intel Realsense T265, a tracking camera whose main functionality is to track the robot movements, especially the rotations. Each robot was equipped with this camera to obtain its orientation.

In order to get their localisation, **EX1** and **Moonraker** got equipped with a GNSS RTK receiver. RTK positioning is a satellite-based technique that provides centimetre-level accuracy by using carrier-phase measurements from GNSS (Global Navigation Satellite Systems) and real-time correction data from a nearby reference station. It significantly improves positioning precision by correcting for signal delays and satellite errors. In our case, the closest public

station was too far away so we registered to a private company services, ensuring decimetre accuracy.

Using a Leica total station, we got the position of the **Clover** robot. A total station is a high-precision optical instrument used in surveying and geolocation that combines an electronic theodolite with an electronic distance meter (EDM). Devices like the Leica Total Station measure both angles and distances to a target reflector, mounted on the top of the antenna, and compute the precise 3D position through trilateration and angular triangulation.

Navigation/Teleoperation

While autonomous navigation got implemented through Docker and in simulation, showing very convincing results, the limited time and some sensor uncertainties made it too complicated to achieve for the real-world experiments.

Instead, functionalities to help multi-robot teleoperation were developed. As visible in Fig. 6.8, each robot was assigned a colour. Blue for EX1, Red for Moonraker and Green for Clover. The controller used to teleoperate the robot would take the colour of the robot, and the 3D model and goal of the robot would also be of the appropriate colour. During the experiment, to reproduce a behaviour closer to autonomous navigation, each robot was teleoperated by a different person.

Technical optimisations

To optimise the network usage within the MRS, we leverage the `ROS_DOMAIN_ID` feature to isolate each robot's ROS 2 communication. In this setup, every robot is assigned a fixed domain ID (e.g., EX1 = 10, Moonraker = 20, Clover = 120), while the lander and mission operations remain on domain ID 0. Each robot runs a dedicated Docker container that hosts a domain bridge, selectively forwarding only the necessary topics for coordination and task execution. This approach significantly reduces bandwidth consumption, as ROS 2 messages are otherwise broadcast to all nodes within the same domain.

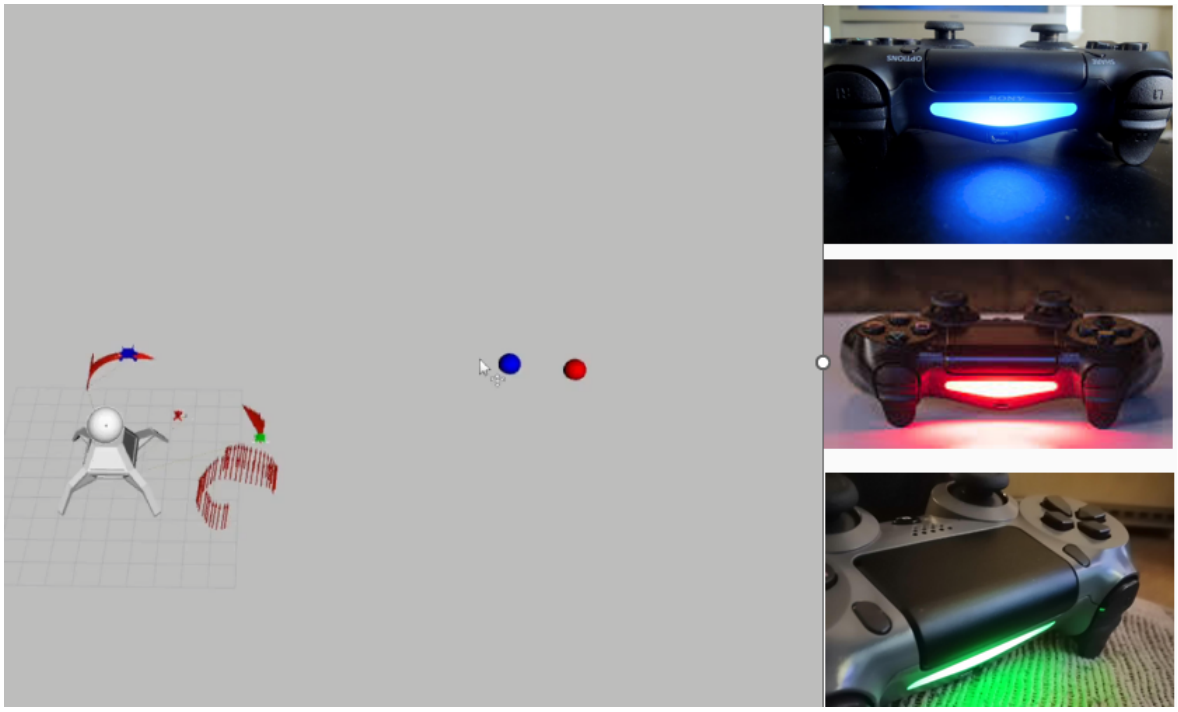


Figure 6.8: Visual of the teleoperation interface, featuring the three coloured robots along with their assigned controllers. It also features one goal for EX1 and Moonraker as blue and red spheres.

6.5.2 Experimental environment

A representative sandy field is used for performing real-world experiments, providing a $25\,\mathrm{km}^2$ outdoor experiment area (see Fig. 6.9).

6.5.3 Experimental Scenario

The experiment aim to demonstrate the applicability of the BackPropagation approach on a real space MRS. It features the robots described in Tab. 6.7.

During the experimental run, the user sends a task to the science rover R_o. Depending on the distance of the assigned task, another robot might be added to the BackPropagation chain, which then autonomously decides its ideal position.

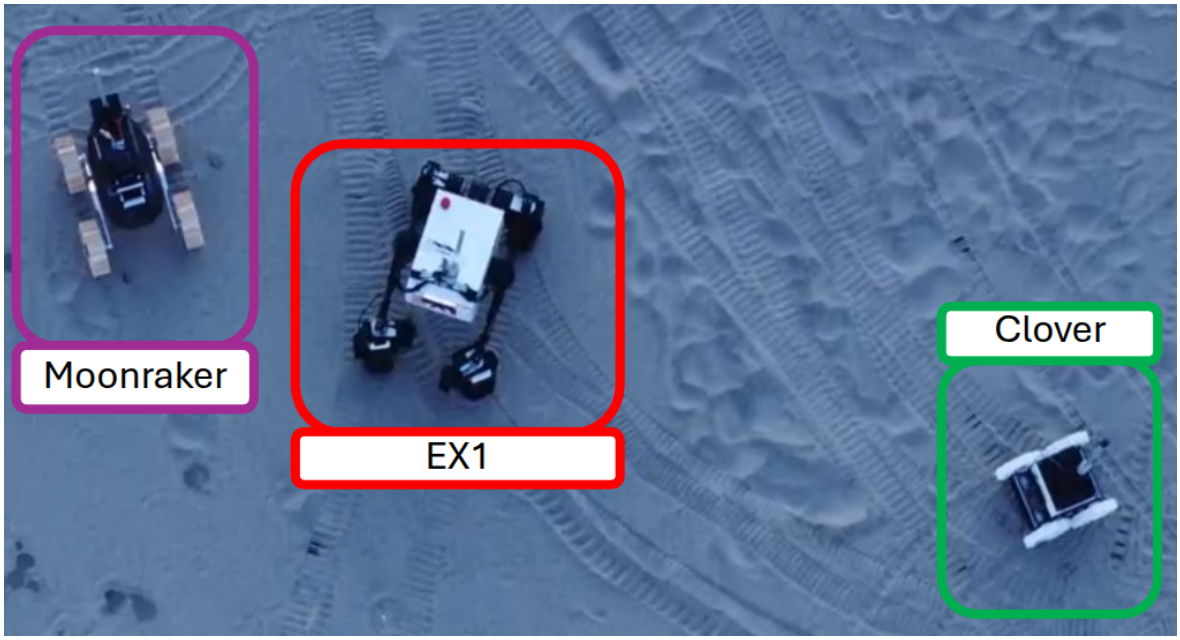


Figure 6.9: Top view of the experimental environment and MRS.

6.5.4 Results

Significant localisation challenges were observed, necessitating the use of teleoperation instead of autonomous navigation. The BackPropagation approach was successfully deployed on the robotic testbed under teleoperated conditions.

The Fig. 6.11 displays a successful sequence of operations. In Fig. 6.11a, a user requests a new position (the blue sphere) from EX1, triggering the BackPropagation. After calculation, it is determined that this position would be too far from the lander for effective communication (cf equation 6.1. In that case, the BackPropagation should be triggered on the next robot toward the lander. However, the lander is directly connected, requiring the addition of a new robot to LP. The algorithm determines that Clover is the robot that is most appropriate to add. In that case, BackPropagation is triggered in Clover in Fig. 6.11b, with the goal being EX1 goals (the blue sphere). The BackPropagation approach computes a new goal for Clover (the green sphere). Then it checks that this new goal is inside the range of the lander. Allowing the launch of movement in all robots part of the LP, Clover and EX1. As visible in Fig. 6.11c, as they started moving in the same time, Clover reaches its goal way before EX1, ensuring

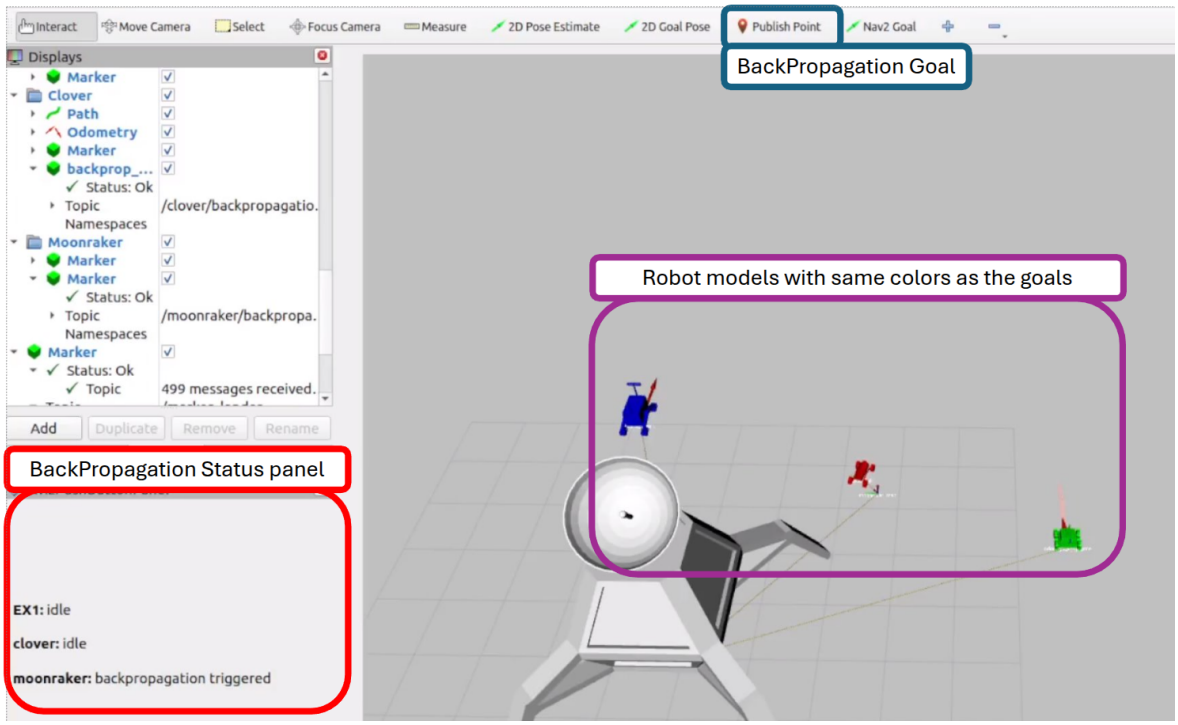


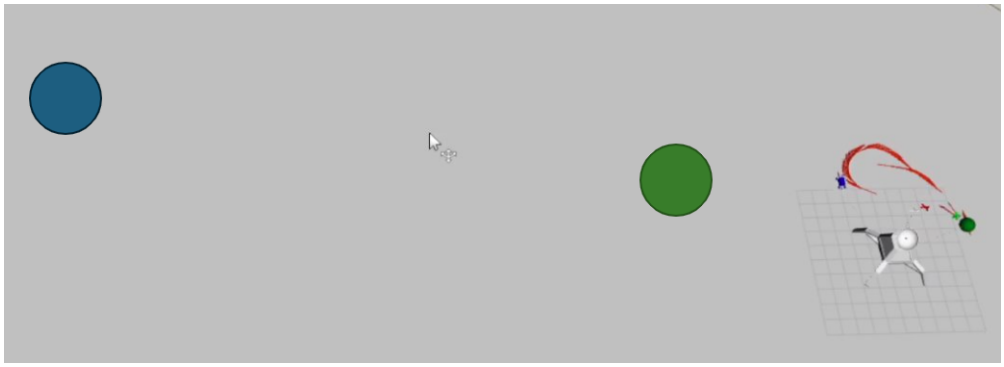
Figure 6.10: User Interface in Rviz

that EX1 stays always connected.

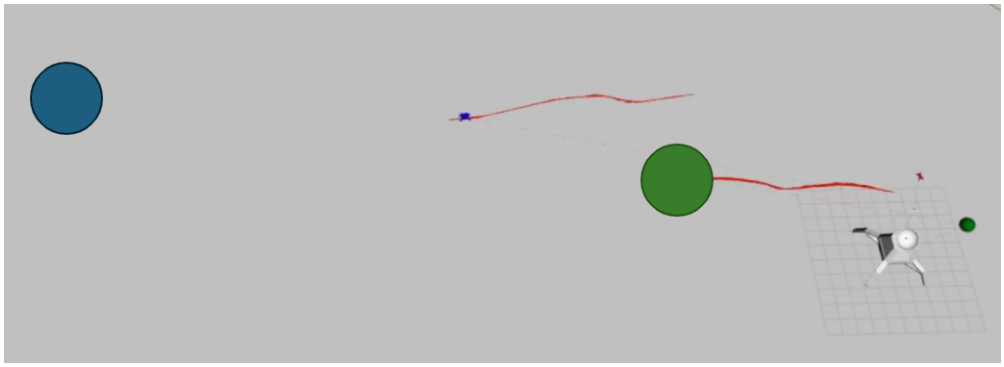
Experimental observations confirmed that the science rover R_o (EX1) consistently reached its designated target without any communication interruptions. The R_o travelled over 200m away from L – well beyond the single-hop range of $129\,\mathrm{m}$ – yet maintained connectivity by autonomously inserting a relay rover. As the science rover received a goal outside the range of L, a relay rover R_1 would compute a positional goal and reposition accordingly, preventing any loss of connectivity. R_o would compute a positional goal within approximately 11.9 seconds. This computation time is negligible compared to the typical durations required for rover traversal and positioning.

This indicates that the BackPropagation algorithm effectively mitigates connectivity disruptions in dynamic environments. Moreover, throughout the experiments, no relay rover experienced computational overload despite the different configurations. The qualitative performance observed suggests significant potential for this approach in applications where reliable communication is critical, such as in planetary exploration missions. Despite the in-

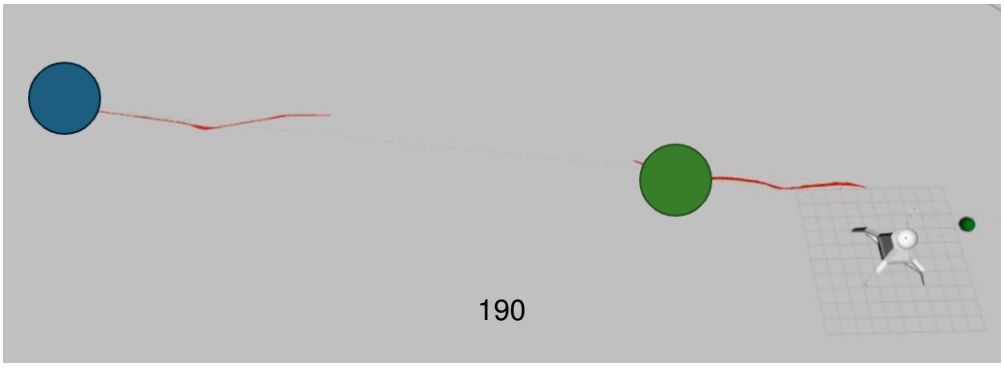
(a) Task request: The operator sends a goal (blue sphere) to R_o: EX1.



(b) BackPropagation: EX1 would be too far from the lander, BackPropagation is triggered on clover, clover receives a new goal (green sphere)



(c) Clover reached goal: Robot Clover reaches the assigned target while EX1 is moving to its goal



(d) Goal achieved: All objectives completed.

Figure 6.11: Sequence of the real-world experiment: (a) Task request, (b) BackPropagation, (c) Clover reached goal, and (d) Goal achieved.

duced limitation of real networks and robots, the BackPropagation algorithm did not overload any robot's computing resources, allowing seamless collaboration.

6.6 Discussion

RQ-3: *How can the robots optimise the network topology?*

Several challenges were encountered during the experimental validation. Most of the issues arise from reliable robot localisation. This constraint highlights the importance of robust localisation system for fully autonomous MRS

Another challenge lies in the scalability, as studies remain to be done on larger MRS. As the BackPropagation approach design is suited for large systems, the impact of keeping one communication chain of relay remains to be studied.

This work introduced the BackPropagation, a novel approach to maintain robust connectivity in heterogeneous MRS designed for planetary exploration. This decentralised approach focuses on minimising the energy consumption of the system while ensuring a required data rate, a key component for space missions. While the BackPropagation do not guarantee permanent connection, fail-safe mechanism reverts the system to the last connected state. Compared to a fully robust solution, BackPropagation requires fewer robots for the same distance, hence consuming less energy. Simulation studies, benchmarked against a state-of-the-art Force-Based Approach, indicate that our method significantly reduces disconnection events (by 61% in average) and lowers the overall distance travelled by the robots (by 27% in average). However, this approach took more time to process each task, processing on average fewer tasks for a given time than the other approaches. Real-world experiments further demonstrate the viability of the algorithm under teleoperated conditions.

These promising results highlight the potential of decentralised MRS for planetary application, offering greater coverage and better energy efficiency than traditional approaches.

6.7 Summary

This chapter introduced the BackPropagation approach, a decentralised algorithm designed to maintain robust connectivity in heterogeneous MRS operating in planetary exploration scenarios. The proposed method leverages a communication-aware placement of relay robots, guided by realistic radio propagation modelling (using a two-ray model) and mission-specific data rate requirements. By combining mesh network communication and dynamic positioning, it addresses the risk of losing connection when a robot moves out of communication range.

Through simulations and benchmarking against a state-of-the-art FBA, the BackPropagation algorithm demonstrated:

- 61% fewer disconnection events on average;
- 27% lower distance travelled per task, significantly reducing energy expenditure;
- while completing 30% fewer tasks due to higher computation time.

These results indicate that the approach prioritises resilience and energy efficiency—key factors for planetary operations.

In addition, real-world experiments conducted at Tohoku University validated the system's portability and practical feasibility. Under teleoperated conditions, the approach successfully maintained connectivity by autonomously repositioning relay robots, confirming its robustness even with heterogeneous hardware and real communication constraints.

Contributions:

- A novel, decentralised algorithm (BackPropagation) for communication-aware relay positioning in MRS using realistic radio propagation models.
- A complete simulation of the system in lunar terrain adapted from real lunar DEM, benchmarked against an existing connectivity maintenance method.
- Experimental validation in a heterogeneous real-world robot fleet, demonstrating the portability and robustness of the approach under realistic constraints.

Chapter 7

Conclusions and Future works

This thesis addresses several key challenges anticipated in the deployment of future MRS, particularly in the context of space exploration. As *coopetitive* systems emerge as a promising paradigm for space robotics, significant advancements are still required across the entire robotic stack. These range from high-level coordination mechanisms for managing coopetition to low-level networking strategies that ensure robust communication. Additionally, intermediate layers must support intelligent use of robots' relaying capabilities to maintain connectivity in dynamic environments.

7.1 Trustful multi-robot collaboration in coopetitive scenario

We introduced our **Make or Buy approach**, a task allocation mechanism allowing a heterogeneous coopetitive MRS to perform tasks in a trusted way. It relies on the DLT technology in order to ensure data ownership, but also trust to exchange currencies between robots. We also propose a cost function to evaluate the cost of each task proposed to the MRS. This framework allows any external user to request a task from the system and to get it for the cheapest price. Future work are:

- Enhancing the cost function with dynamic factors such as task priority, robot reliability, and real-time environmental conditions.

- Investigating trust and reputation models for robot agents in long-term and adversarial coopetitive scenarios.
- Deploying the Make or Buy framework in larger, real-world multi-robot settings to assess scalability and interoperability.
- Studying legal and ethical implications of autonomous economic decision-making in decentralized robotic systems.

7.2 Efficient communication for heterogeneous Multi-Robot systems

Our second contribution addressed the critical need for efficient communication in the context of space-based MRS. While mesh networks have been widely recognized as a promising topology for such scenarios, their integration with robotics middleware like ROS 2 remains underexplored. In particular, the behavior of ROS 2 communication layers over mesh networks had not been rigorously studied. Through extensive experimentation, we evaluated several ROS 2 middleware options and identified *Zenoh* -a novel one- as a significantly more stable (58% to 146.93% better) and resilient solution—especially under dynamic topology changes common in mobile robot deployments.

Future research directions in the area of mesh networks for MRS include:

- Investigating the impact of large message sizes on network performance, and developing mitigation strategies such as compression, fragmentation, or prioritization.
- Designing network resource optimization mechanisms to efficiently allocate bandwidth and minimize congestion across the MRS, given the inherent limitations of mesh architectures.

7.3 Network topology optimization

The final contribution of this thesis addresses a crucial challenge that lies at the intersection of task coordination and communication: maintaining network connectivity in dynamic, mobile MRS. While our previous contributions established the foundations for a coopetitive system operating over mesh networks, mobility introduces the risk of individual robots becoming disconnected from the network. To mitigate this, we proposed the **BackPropagation** approach, a decentralized strategy in which selected robots dynamically reposition themselves to act as mobile relays, preserving the integrity of the mesh.

This approach combines insights from graph theory with environment-aware planning to proactively maintain communication links. Our evaluation showed that while it resulted in a 61% reduction in robot disconnections, it came at the cost of a 30% reduction in overall task throughput, highlighting the inherent trade-offs between connectivity and productivity.

Future work directions include:

- Investigating adaptive balancing strategies between task execution and relay responsibilities, potentially guided by real-time utility metrics.
- Validating the approach in more complex, large-scale environments with real-world terrain and heterogeneous robot capabilities.
- Integrate the relay task as a new task part of the make or buy system.

References

[1] Shuai Li et al. “Direct evidence of surface exposed water ice in the lunar polar regions”. In: *Proceedings of the National Academy of Sciences* 115.36 (2018), pp. 8907–8912. DOI: 10.1073/pnas.1802345115.

[2] W. C. Feldman et al. “Fluxes of fast and epithermal neutrons from Lunar Prospector: Evidence for water ice at the lunar poles”. In: *Science* 281.5382 (1998), pp. 1496–1500. DOI: 10.1126/science.281.5382.1496.

[3] Anthony Colaprete et al. “Detection of Water in the LCROSS Ejecta Plume”. In: *Science* 330.6003 (2010), pp. 463–468. DOI: 10.1126/science.1186986.

[4] Gabe Mounce and Makena Young. *Reusability in Space: How Reusable Rockets Are Lowering the Cost of Access to Orbit*. Accessed July 2025. Center for Strategic and International Studies (CSIS), 2023. URL: https://www.csis.org/analysis/reusability-space-how-reusable-rockets-are-lowering-cost-access-orbit.

[5] Bohumil Doboš. “The Eagle Returned: Geopolitical Aspects of the New Lunar Race”. In: *Astropolitics* 20.2-3 (2022), pp. 121–134. DOI: 10.1080/14777622.2022.2141958. eprint: https://doi.org/10.1080/14777622.2022.2141958. URL: https://doi.org/10.1080/14777622.2022.2141958.

[6] NASA. *The Artemis Accords: Principles for Cooperation in the Civil Exploration and Use of the Moon, Mars, Comets, and Asteroids for Peaceful Purposes*. Accessed July 2025. 2020. URL: https://www.nasa.gov/specials/artemis-accords/.

[7] China National Space Administration (CNSA) and Roscosmos. *International Lunar Research Station Guide for Partnership*. Accessed July 2025. 2021. URL: https://www.cnsa.gov.cn/english/n6465652/n6465653/c6812150/content.html.

[8] M. Anand et al. “A brief review of chemical and mineralogical resources on the Moon and likely initial in situ resource utilization (ISRU) applications”. In: *Planetary and Space Science* 74.1 (2012). Scientific Preparations For Lunar Exploration, pp. 42–48. ISSN: 0032-0633. DOI: https://doi.org/10.1016/j.pss.2012.08.012. URL: https://www.sciencedirect.com/science/article/pii/S0032063312002498.

[9] S. Radl et al. “From lunar regolith to oxygen and structural materials”. In: *CEAS Space Journal* 14 (2022), pp. 663–676. DOI: 10.1007/s12567-022-00465-w.

[10] Cheng Zhou et al. “Properties and Characteristics of Regolith-Based Materials for Extraterrestrial Construction”. In: *Engineering* 37 (2024), pp. 159–181. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2023.11.019. URL: https://www.sciencedirect.com/science/article/pii/S2095809924000511.

[11] Assiya Akisheva and Yves Gourinat. “Utilisation of Moon Regolith for Radiation Protection and Thermal Insulation in Permanent Lunar Habitats”. In: *72nd International Astronautical Congress (IAC)*. Dubai, UAE, 2021. URL: https://www.researchgate.net/publication/351220810_Utilisation_of_Moon_Regolith_for_Radiation_Protection_and_Thermal_Insulation_in_Permanent_Lunar_Habitats.

[12] Jack O. Burns et al. “A space-based observational strategy for characterizing the first stars and black holes at cosmic dawn”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378.2187 (2020), p. 20190564. DOI: 10.1098/rsta.2019.0564. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2019.0564.

[13] Chunlai Li et al. “Detection of the Lunar Subsurface Layering Structures by the Lunar Penetrating Radar Onboard Chang’e-4 Mission”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. V-3-2020. 2020, pp. 595–602. DOI: 10.5194/isprs-annals-V-3-2020-595-2020. URL:

http://pmrslab.cn/publications/publications/isprs-annals-V-3-2020-595-2020.pdf.

[14] Kenneth A. Farley et al. “Mars 2020 Mission Overview”. In: *Space Science Reviews* 216.8 (2020), p. 142. DOI: 10.1007/s11214-020-00762-y.

[15] M. H. Hecht et al. “Mars Oxygen ISRU Experiment (MOXIE) — Producing oxygen on the surface of Mars”. In: *Science Advances* 8.35 (2022), eabo5219. DOI: 10.1126/sciadv.abo5219. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9432831/.

[16] NASA Science Mission Directorate. *Curiosity’s Location Map*. https://science.nasa.gov/mission/msl-curiosity/location-map/. Accessed: 2025-07-15. 2025.

[17] A. Pretto et al. “Building an Aerial-Ground Robotics System for Precision Farming: An Adaptable Solution”. In: *IEEE Robotics & Automation Magazine* 28.3 (2021). arXiv: 1911.03098, pp. 29–49. DOI: 10.1109/MRA.2020.3012492. URL: http://arxiv.org/abs/1911.03098.

[18] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. “A Survey and Analysis of Multi-Robot Coordination”. In: *International Journal of Advanced Robotic Systems* 10.12 (Dec. 2013), p. 399. DOI: 10.5772/57313. URL: http://journals.sagepub.com/doi/10.5772/57313.

[19] Lynne E. Parker. “Multiple Mobile Robot Systems”. en. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Berlin, Heidelberg: Springer, 2008, pp. 921–941. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_41. URL: https://doi.org/10.1007/978-3-540-30301-5_41 (visited on 03/15/2023).

[20] P. Caloud et al. “Indoor automation with many mobile robots”. In: *IEEE Int. Workshop on Intel. Robots and Systems, Towards a New Frontier of Applications*. 1990. DOI: 10.1109/IROS.1990.262370. URL: http://ieeexplore.ieee.org/document/262370/.

[21] R. Alur et al. “A Framework and Architecture for Multirobot Coordination”. In: *Experimental Robotics VII*. Vol. 271. Springer Berlin Heidelberg, 2001, pp. 303–312. ISBN: 9783540421047. DOI: 10.1007/3-540-45118-8_31. URL: http://link.springer.com/10.1007/3-540-45118-8_31.

[22] L.E. Parker. “ALLIANCE: an architecture for fault tolerant multirobot cooperation”. In: *IEEE Transactions on Robotics and Automation* 14.2 (Apr. 1998). Conference Name: IEEE Transactions on Robotics and Automation, pp. 220–240. ISSN: 2374-958X. DOI: 10.1109/70.681242.

[23] C.A.C. Parker and Hong Zhang. “Cooperative Decision-Making in Decentralized Multiple-Robot Systems: The Best-of-N Problem”. In: *IEEE/ASME Transactions on Mechatronics* 14.2 (2009), pp. 240–251. DOI: 10.1109/TMECH.2009.2014370. URL: http://ieeexplore.ieee.org/document/4801702/.

[24] Adam M. Brandenburger and Barry J. Nalebuff. *Co-Opetition*. en. Crown, July 2011. ISBN: 978-0-307-79054-5.

[25] Yara Rizk, Mariette Awad, and Edward W. Tunstel. “Cooperative Heterogeneous Multi-Robot Systems: A Survey”. en. In: *ACM Computing Surveys* 52.2 (Mar. 2020), pp. 1–31. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3303848. URL: https://dl.acm.org/doi/10.1145/3303848 (visited on 06/14/2023).

[26] H. Asama, A. Matsumoto, and Y. Ishida. “Design Of An Autonomous And Distributed Robot System: Actress”. In: *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems ’. (IROS ’89) ’The Autonomous Mobile Robots and Its Applications*. Sept. 1989, pp. 283–290. DOI: 10.1109/IROS.1989.637920.

[27] Carl Hewitt, Peter Bishop, and Richard Steiger. “A universal modular ACTOR formalism for artificial intelligence”. In: *Proceedings of the 3rd international joint conference on Artificial intelligence*. IJCAI’73. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1973, pp. 235–245. (Visited on 07/13/2023).

[28] M.B. Dias et al. “Market-Based Multirobot Coordination: A Survey and Analysis”. In: *Proceedings of the IEEE* 94.7 (July 2006), pp. 1257–1270. ISSN: 1558-2256. DOI: 10.1109/jproc.2006.876939.

[29] Yinong Chen, Zhihui Du, and Marcos García-Acosta. “Robot as a Service in Cloud Computing”. In: *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*. June 2010, pp. 151–158. DOI: 10.1109/SOSE.2010.44.

[30] Marco Tranzatto et al. “CERBERUS in the DARPA Subterranean Challenge”. In: *Science Robotics* 7.66 (May 2022), eabp9742. ISSN: 2470-9476.

[31] Brian Gerkey, Richard Vaughan, and Andrew Howard. “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems”. In: *Proceedings of the International Conference on Advanced Robotics* (Aug. 2003).

[32] M. Montemerlo, N. Roy, and S. Thrun. “Perspectives on standardization in mobile robot programming: the Carnegie Mellon Navigation (CARMEN) Toolkit”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 3. 2003, 2436–2441 vol.3. DOI: 10.1109/IROS.2003.1249235.

[33] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. “YARP: Yet another robot platform”. In: *International Journal of Advanced Robotic Systems* 3 (Mar. 2006). DOI: 10.5772/5761.

[34] N. Ando et al. “RT-Component Object Model in RT-Middleware - Distributed Component Middleware for RT (Robot Technology)”. In: *2005 International Symposium on Computational Intelligence in Robotics and Automation*. 2005, pp. 457–462. DOI: 10.1109/CIRA.2005.1554319.

[35] Open Robotics. *Open-RMF: The Open Robotics Middleware Framework*. https://www.open-rmf.org/. Accessed 2025-10-14. 2024.

[36] OASIS. *MQTT Version 5.0: OASIS Standard*. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html. 2019.

[37] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. “Exploring the performance of ROS2”. In: *Proceedings of the 13th International Conference on Embedded Software*. EMSOFT ’16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 1–10. ISBN: 978-1-4503-4485-2. DOI: 10.1145/2968478.2968502. URL: https://dl.acm.org/doi/10.1145/2968478.2968502 (visited on 05/25/2023).

[38] National Aeronautics and Space Administration. *Lunokhod 02*. Accessed on 2022-02-18. May 2018. URL: https://solarsystem.nasa.gov/missions/lunokhod-02/in-depth/.

[39] YouQing Ma et al. “A precise visual localisation method for the Chinese Chang’e-4 Yutu-2 rover”. In: *Photogrammetric record* 35.169 (2020), pp. 10–39. ISSN: 0031-868X.

[40] Holger Heuseler. *Die Mars Mission : Pathfinder, Sojourner und die Eroberung des roten Planeten*. München Wien Zürich, 1998.

[41] Emily Lakdawalla. *The Design and Engineering of Curiosity : How the Mars Rover Performs Its Job.* Springer Praxis Bks. ProQuest, 2018. ISBN: 9783319681467.

[42] Raymond E et. al Arvidson. “Opportunity Mars Rover mission: Overview and selected results from Purgatory ripple to traverses to Endeavour crater”. In: *Journal of Geophysical Research: Planets* 116.E7 (2011).

[43] National Aeronautics and Space Administration. *Rover Brains*. https://mars.nasa.gov/mars2020/space
Accessed on 2022-02-25. Dec. 2021.

[44] National Aeronautics and Space Administration. *Wheels and Legs*. https://mars.nasa.gov/mars2020/sp
Accessed on 2022-10-24. Dec. 2021.

[45] Lunar Outpost. *Lunar Outpost*. https://lunaroutpost.com/. Accessed on 2022-02-28. 2021.

[46] ispace. *Hakuto-R*. https://ispace-inc.com/. Accessed on 2022-02-28. 2021.

[47] Mohammad Alfraheed and Abdullah Al-Zaghameem. “Exploration and Cooperation Robotics on the Moon”. In: *Journal of Signal and Information Processing* 04.03 (Jan. 2013), pp. 253–258. DOI: 10.4236/jsip.2013.43033.

[48] Florian Cordes et al. “LUNARES: lunar crater exploration with heterogeneous multi robot systems”. In: *Intelligent Service Robotics* 4.1 (Jan. 2011), pp. 61–89. ISSN: 1861-2784. DOI: 10.1007/s11370-010-0081-4.

[49] German Research Center for Artificial Intelligence GmbH. *RIMRES - Reconfigurable Integrated Multi Robot Exploration System*. https://robotik.dfki-bremen.de/en/research/projects/rimres.html. Accessed on 2022-02-22. May 2022.

[50] Jürgen Leitner. "Multi-robot Cooperation in Space: A Survey". In: *2009 Advanced Technologies for Enhanced Quality of Life*. July 2009. DOI: 10.1109/AT-EQUAL.2009.37.

[51] Félix Quinton, Christophe Grand, and Charles Lesire. "Market Approaches to the Multi-Robot Task Allocation Problem: A Survey". In: *Journal of Intelligent & Robotic Systems* 107.2 (Feb. 2023), p. 29. ISSN: 1573-0409. DOI: 10.1007/S10846-022-01803-0. (Visited on 09/01/2023).

[52] R. Zlot et al. "Multi-Robot Exploration Controlled by a Market Economy". In: *Proc. IEEE Int. Conf. Robot. Autom.* 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). Vol. 3. Washington, USA: IEEE, May 2002, pp. 3016–3023. DOI: 10.1109/robot.2002.1013690.

[53] L.E. Cánez, K.W. Platts, and D.R. Probert. "Developing a Framework for Make-or-buy Decisions". In: *International Journal of Operations & Production Management* 20.11 (Jan. 2000), pp. 1313–1330. ISSN: 0144-3577. DOI: 10.1108/01443570010348271. (Visited on 06/07/2023).

[54] E. K. Belyaeva, D. Yu. Ivanov, and S. V. Domnina. "Contractual Arrangements Between Providers and Consumers of Digital Technologies in Space Industry". In: *Digital Economy and the New Labor Market: Jobs, Competences and Innovative HR Technologies*. Ed. by Svetlana Igorevna Ashmarina and Valentina Vyacheslavovna Mantulenko. Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2021, pp. 135–142. ISBN: 978-3-030-60926-9. DOI: 10.1007/978-3-030-60926-9_19.

[55] Primavera De Filippi and Andrea Leiter. "Blockchain in Outer Space". In: *Amer. Jour. of Int. Law* 115 (2021), pp. 413–418. ISSN: 2398-7723. DOI: 10.1017/aju.2021.63. (Visited on 10/16/2022).

[56] Davide Calvaresi et al. “Trusted Registration, Negotiation, and Service Evaluation in Multi-Agent Systems throughout the Blockchain Technology”. In: *Int. Conf. on Web Int.* Santiago, Chile: IEEE, Dec. 2018, pp. 56–63. ISBN: 978-1-5386-7325-6. DOI: 10.1109/WI.2018.0-107.

[57] Christian Catalini and Joshua S. Gans. “Some Simple Economics of the Blockchain”. In: *cacm* 63 (2020), pp. 80–90. ISSN: 0001-0782. DOI: 10.1145/3359552. (Visited on 06/14/2023).

[58] Bert-Jan Butijn, Damian A. Tamburri, and Willem-Jan van den Heuvel. “Blockchains: A Systematic Multivocal Literature Review”. In: *ACM Computing Surveys* 53.3 (July 2020), pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/3369052.

[59] Roman Beck, Christoph Müller-Bloch, and John Leslie King. “Governance in the Blockchain Economy: A Framework and Research Agenda”. In: *JAIS* 19.10 (2018), pp. 1020–1034. ISSN: 15369323. DOI: 10.17705/1jais.00518. (Visited on 12/21/2020).

[60] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: https://bitcoin.org/bitcoin.pdf.

[61] Niclas Kannengiesser et al. “Challenges and Common Solutions in Smart Contract Development”. In: *IEEE Transactions on Software Engineering* 48.11 (Dec. 2022), pp. 4291–4318. ISSN: 1939-3520. DOI: 10.1109/TSE.2021.3116808.

[62] Thomas Heinz Meitinger. “Smart contracts”. In: *Informatik Spektrum* 40.4 (Aug. 2017), pp. 371–375. ISSN: 1432-122X. DOI: 10.1007/s00287-017-1045-2. (Visited on 12/08/2022).

[63] Eduard Hartwich et al. “Probably Something: A Multi-Layer Taxonomy of Non-Fungible Tokens”. In: *Internet Research* ahead-of-print.ahead-of-print (Jan. 2023), p. 26. ISSN: 1066-2243. DOI: 10.1108/INTR-08-2022-0666. arXiv: 2209.05456 [cs].

[64] Khalid Husain Ansari and Umesh Kulkarni. “Implementation of Ethereum Request for Comment (ERC20) Token”. In: *Proceedings of the 3rd International Conference on Advances in Science & Technology*. 3561395. Rochester, NY: Elsevier, Apr. 2020, p. 6. DOI: 10.2139/ssrn.3561395. (Visited on 06/16/2023).

[65] Aleksandr Kapitonov et al. “Robonomics Based on Blockchain as a Principle of Creating Smart Factories”. In: *Proc. Int. Conf. on Intern. of Thin.: Sys., Manag. and Sec.* Valencia, Spain: IEEE, Oct. 2018, pp. 78–85. ISBN: 978-1-5386-9585-2. DOI: 10.1109/IoTSMS.2018.8554864.

[66] Sergio Garcıa et al. “An Architecture for Decentralized, Collaborative, and Autonomous Robots”. In: *2018 IEEE International Conference on Software Architecture (ICSA)*. Seattle, WA: IEEE, Apr. 2018, pp. 75–7509. ISBN: 978-1-5386-6398-1. DOI: 10.1109/ICSA.2018.00017.

[67] Sergio García et al. “Robotics Software Engineering: A Perspective from the Service Robotics Domain”. In: *Proc. of the 28th ACM Joint Meeting on Eur. Soft. Eng. Conf. and Symposium on the Foundations of Software Engineering*. Virtual Event USA: ACM, Nov. 2020, pp. 593–604. ISBN: 978-1-4503-7043-1. DOI: 10.1145/3368089.3409743. (Visited on 06/27/2023).

[68] Sergey Lonshakov et al. *Robonomics: Platform for Integration of Cyber Physical Systems into Human Economy for Engineers, Smart Cities and Industry 4.0 Creators*. Dec. 2018. DOI: 10.13140/RG.2.2.23928.60169. (Visited on 10/16/2022).

[69] Space Decentral. *Space Decentral: To Space, Together*. 2020. URL: https://spacedecentral.net/ (visited on 10/17/2022).

[70] Consensys Space. *Open Source Space*. 2018. URL: https://www.consensys.space/ (visited on 06/15/2023).

[71] Renan Lima Baima and Esther Luna Colombini. “Modeling Object’s Affordances via Reward Functions”. In: *Proc. Int. Conf. on Sys., Man, and Cyb.* Melbourne, Australia: IEEE, Oct. 2021, pp. 2183–2190. ISBN: 978-1-66544-207-7. DOI: 10.1109/smc52423.2021.9658915.

[72] Ralph D. Lorenz. “How Far Is Far Enough? Requirements Derivation for Planetary Mobility Systems”. In: *Advances in Space Research* 65.5 (Mar. 2020), pp. 1383–1401. ISSN: 0273-1177. DOI: 10.1016/j.asr.2019.12.011. (Visited on 08/31/2023).

[73] Tobias Guggenberger et al. “An In-depth Investigation of the Performance Characteristics of Hyperledger Fabric”. In: *Computers and Industrial Engineering* 173 (2022), p. 108716. DOI: 10.1016/j.cie.2022.108716.

[74] Eduardo Castelló Ferrer. “The Blockchain: A New Framework for Robotic Swarm Systems”. In: *Proc. Future Technol. Conf.* Vol. 881. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2019, pp. 1037–1058. ISBN: 978-3-030-02683-7. DOI: 10.1007/978-3-030-02683-7_77.

[75] Alina Orlova, Roberto Nogueira, and Paula Chimenti. “The Present and Future of the Space Sector: A Business Ecosystem Approach”. In: *Space Policy* 52 (May 2020), p. 101374. ISSN: 0265-9646. DOI: 10.1016/j.spacepol.2020.101374. (Visited on 07/21/2023).

[76] Guido R. Hiertz et al. “IEEE 802.11s: The WLAN Mesh Standard”. In: *IEEE Wireless Commun.* 17.1 (Feb. 2010). Conference Name: IEEE Wireless Communications, pp. 104–111. ISSN: 1558-0687. DOI: 10.1109/MWC.2010.5416357.

[77] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. “Wireless mesh networks: a survey”. In: *Computer Networks* 47.4 (Mar. 15, 2005), pp. 445–487. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2004.12.001.

[78] Karthika K.C. “Wireless mesh network: A survey”. In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Mar. 2016, pp. 1966–1970. DOI: 10.1109/WiSPNET.2016.7566486. URL: https://ieeexplore.ieee.org/abstract/document/7566486 (visited on 10/25/2023).

[79] I.D. Chakeres and E.M. Belding-Royer. “AODV routing protocol implementation design”. In: *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* Mar. 2004, pp. 698–703. DOI: 10.1109/ICDCSW.2004.1284108. URL: https://ieeexplore.ieee.org/abstract/document/1284108 (visited on 11/10/2023).

[80] Syed Talib Abbas Jafri et al. “Split Hop Penalty for Transmission Quality Metrics in a Better Approach to Mobile Ad Hoc Networking (BATMAN) for IoT-Based MANET”. en. In: *Symmetry* 15.5 (Apr. 2023), p. 969. ISSN: 2073-8994. DOI: 10.3390/sym15050969.

[81] *IEEE 802.11s Mesh Networking, D1.06. Jul. 2007.* English. July 2007.

[82] Lihua Yang and Sang-Hwa Chung. “HWMP+: An Improved Traffic Load Sheme for Wireless Mesh Networks”. In: *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*. June 2012, pp. 722–727. DOI: 10.1109/HPCC.2012.102.

[83] Alireza Ghaffarkhah and Yasamin Mostofi. “Communication-Aware Motion Planning in Mobile Networks”. In: *IEEE Trans. Autom. Control* 56.10 (Oct. 2011), pp. 2478–2485. ISSN: 1558-2523.

[84] Meng Ji and M. Egerstedt. “Distributed Coordination Control of Multiagent Systems While Preserving Connectedness”. In: *IEEE Trans. Robot.* 23.4 (Aug. 2007), pp. 693–703. ISSN: 1552-3098.

[85] Kai Ding, Homayoun Yousefi’zadeh, and Faryar Jabbari. “Connectivity Maintenance in Mobile Networks”. In: *IEEE/ACM Trans. Netw.* 28.3 (June 2020), pp. 1269–1282. ISSN: 1558-2566.

[86] Abdullah Konak, Orhan Dengiz, and Alice E. Smith. “Improving Network Connectivity in Ad Hoc Networks Using Particle Swarm Optimization and Agents”. en. In: *Wireless Network Design: Optimization Models and Solution Procedures*. Ed. by Jeff Kennington, Eli Olinick, and Dinesh Rajan. International Series in Operations Research & Management Science. New York, NY: Springer, 2011, pp. 247–267. ISBN: 978-1-4419-6111-2. DOI: 10.1007/978-1-4419-6111-2_11. URL: https://doi.org/10.1007/978-1-4419-6111-2_11 (visited on 01/15/2024).

[87] Leonard Bryan Paet et al. “Maintaining Connectivity in Multi-Rover Networks for Lunar Exploration Missions”. In: *2021 IEEE 17th Int. Conf. Automat. Sci. Eng. (CASE)*. Lyon, France: IEEE, Aug. 2021, pp. 1539–1546. ISBN: 978-1-66541-873-7.

[88] Izzet F. Senturk, Kemal Akkaya, and Sabri Yilmaz. “Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information”. In: *Ad Hoc Networks* 13 (Feb. 2014), pp. 487–503. ISSN: 1570-8705.

[89] F. Amigoni et al. “Online Update of Communication Maps for Exploring Multirobot Systems Under Connectivity Constraints”. In: *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 513–526. ISBN: 978-3-030-05815-9. DOI: 10.1007/978-3-030-05816-6_36.

[90] Yupeng Yang et al. *Decentralized Multi-Robot Line-of-Sight Connectivity Maintenance under Uncertainty*. arXiv:2406.12802 [cs]. June 2024. DOI: 10.48550/arXiv.2406.12802. URL: http://arxiv.org/abs/2406.12802 (visited on 07/14/2025).

[91] Shreya Santra et al. “Experimental Validation of Deterministic Radio Propagation Model developed for Communication-aware Path Planning”. In: *2021 IEEE 17th Int. Conf. Automat. Sci. Eng.(CASE)*. Aug. 2021, pp. 1241–1246.

[92] Emanuel Staudinger et al. “Terrain-aware communication coverage prediction for cooperative networked robots in unstructured environments”. In: *Acta Astronautica* 202 (Jan. 2023), pp. 799–805. ISSN: 00945765. DOI: 10.1016/j.actaastro.2022.10.050. URL: https://linkinghub.elsevier.com/retrieve/pii/S0094576522006002 (visited on 11/27/2023).

[93] Yiannis Kantaros and Michael M. Zavlanos. “Distributed Intermittent Connectivity Control of Mobile Robot Networks”. In: *IEEE Transactions on Automatic Control* 62.7 (2017), pp. 3109–3121. DOI: 10.1109/TAC.2016.2626400.

[94] Thomas M. J. Fruchterman and Edward M. Reingold. “Graph drawing by force-directed placement”. In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. ISSN: 1097-024X.

[95] Philippe Ludivig et al. “Building a piece of the Moon: Construction of two indoor lunar analogue environments”. In: *IAF Space Exploration Symposium* (2020). URL: https://orbilu.uni.lu/handle/10993/45539.

[96] Thomas Krueger. *Simple delay simulator with FreeBSD*. Report. https://ideas.esa.int/servlet/hype/IMT?documentTableId=45087640621936072&userAction=Browse&templateName=&documentId=effe6948fbaaea4845bfb3065769966f. ESA, May 2021.

[97] Kell Ideas. *Leo Rover - Build and program your own robot*. https://www.leorover.tech/. Accessed on 2021-07-30. 2021.

[98] Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE. *FKIE multimaster for ROS*. https://github.com/fkie/multimaster_fkie. Accessed on 2022-10-20. Nov. 2017.

[99] P. Gläser et al. "Illumination conditions at the lunar poles: Implications for future exploration". In: *Planetary and Space Science*. Lunar Reconnaissance Orbiter - Seven Years of Exploration and Discovery (Nov. 2018). DOI: 10.1016/j.pss.2017.07.006. URL: https://www.sciencedirect.com/science/article/pii/S0032063317300478.

[100] ESA - European Space Agency. *2ND FIELD TEST - ESA-ESRIC Challenge*. https://www.spaceresourc of-1st-field-test. Accessed on 2023-06-28. 2023.

[101] Ernie Wright. *NASA Scientific Visualization Studio — Counting Craters on the Moon*. Sept. 2010. URL: https://svs.gsfc.nasa.gov/3662 (visited on 07/18/2024).

[102] Lynne E. Parker. "Multiple Mobile Robot Systems". In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Berlin, Heidelberg: Springer, 2008, pp. 921–941. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_41.

[103] Tiehui Zhang et al. "Group-Symmetric Consensus for Nonholonomic Mobile Multirobot Systems in Coopetition Networks". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 236.7 (Apr. 2022), pp. 3743–3754. ISSN: 0954-4062. DOI: 10.1177/09544062211045482. (Visited on 05/15/2023).

[104] George A. Akerlof. "The Market for "Lemons": Quality Uncertainty and the Market Mechanism". In: *The Quarterly Journal of Economics* 84.3 (Aug. 1970), pp. 488–500. ISSN: 0033-5533. DOI: 10.2307/1879431. (Visited on 10/16/2022).

[105] Eduard Hartwich et al. “Machine Economies”. In: *Electronic Markets* 33.1 (July 2023), p. 36. ISSN: 1422-8890. DOI: 10.1007/s12525-023-00649-0. (Visited on 07/31/2023).

[106] European Space Agency. *Blockchain and Earth Observation: a white paper*. en-US. Section: Publications. Apr. 2019. URL: https://eo4society.esa.int/2019/04/09/blockchain-and-earth-observation-a-white-paper/ (visited on 11/27/2023).

[107] Mariel Borowitz. “Strategic Implications of the Proliferation of Space Situational Awareness Technology and Information: Lessons Learned from the Remote Sensing Sector”. In: *Space Policy* 47 (Feb. 2019), pp. 18–27. ISSN: 0265-9646. DOI: 10.1016/j.spacepol.2018.05.002. (Visited on 12/08/2022).

[108] Marshall Smith et al. “The Artemis Program: An Overview of NASA’s Activities to Return Humans to the Moon”. In: *IEEE Aerosp. Conf.* 2020 IEEE Aerospace Conference. Big Sky, MT, USA: IEEE, Mar. 2020, p. 10. DOI: 10.1109/aero47225.2020.9172323.

[109] Amanda M. Leon. “Mining for Meaning: An Examination of the Legality of Property Rights in Space Resources”. In: *Virginia Law Review* 104 (2018), pp. 497–546. ISSN: 00426601.

[110] Bruce G. Cameron, Theodore Seher, and Edward F. Crawley. “Goals for Space Exploration Based on Stakeholder Value Network Considerations”. In: *Acta Astronautica* 68 (2011), pp. 2088–2097. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2010.11.003. (Visited on 07/24/2023).

[111] Erin C McKiernan et al. “How Open Science Helps Researchers Succeed”. In: *eLife* 5 (July 2016). Ed. by Peter Rodgers, pp. 1–19. ISSN: 2050-084X. DOI: 10.7554/elife.16800.

[112] Alexandra Höß et al. “The Blockchain Effect: From Inter-Ecosystem to Intra-Ecosystem Competition”. In: *ECIS 2021 Proceedings*. Research Papers. Marrakech, Morocco: AIS, June 2021, p. 16. ISBN: 978-1-73363-256-0. URL: https://aisel.aisnet.org/ecis2021%5C_rp/36 (visited on 12/11/2022).

[113] Raiane Coelho et al. “Integrating Blockchain for Data Sharing and Collaboration Support in Scientific Ecosystem Platform”. In: *Proc. of the Hawaii Int.Conf. on Sys. Sci.* Hawaii International Conference on System Sciences. Hawaii: IEEE, 2021, p. 10. DOI: 10.24251/HICSS.2021.031. (Visited on 06/22/2023).

[114] Natasa Zivic, Christoph Ruland, and Jochen Sassmannshausen. “Distributed Ledger Technologies for M2M Communications”. In: *Proc. Int. Conf. Inf. Netw.* 2019 International Conference on Information Networking (ICOIN). Kuala Lumpur, Malaysia: IEEE, Jan. 2019, pp. 301–306. DOI: 10.1109/icoin.2019.8718115.

[115] Johannes Sedlmeir et al. “The Transparency Challenge of Blockchain in Organizations”. In: *Electronic Markets* 32 (2022), pp. 1779–1794. DOI: 10.1007/s12525-022-00536-0.

[116] Dave Van Der Meer et al. “REALMS 2 -RESILIENT EXPLORATION AND LUNAR MAPPING SYSTEM 2”. English. In: *Proceedings of ASTRA 2023*. Leyden, Netherlands: ASTRA, 2023, pp. 1–8. DOI: 10.3389/frobt.2023.1127496. URL: https://orbilu.uni.lu/handle/10993/57411 (visited on 01/04/2024).

[117] Jeremi Garcet et al. “Lunar Volatiles Mobile Instrumentation (LUVMI) Project Results”. In: *IAC-19 SYMPOSIUMS*. Vol. A3. A3 IAF SPACE EXPLORATION SYMPOSIUM. Washington, DC, USA: IAC, Oct. 2019, p. 9. DOI: 10.3030/727220.

[118] Jason Goo et al. *Blockchain Lunar Registry*. https://diana.io. 2019. (Visited on 10/17/2022).

[119] Ian A. Crawford. “Lunar Resources: A Review”. In: *Prog. Phys. Geogr. Earth Environ.* 39.2 (Apr. 2015), pp. 137–167. ISSN: 0309-1333. DOI: 10.1177/0309133314567585.

[120] Martin J. Schuster et al. “The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5315–5322. DOI: 10.1109/lra.2020.3007468.

[121] Société Européenne des Satellites. *SES Satellite Telecommunications Network Provider*. Company. Mar. 2021. URL: https://www.ses.com/ (visited on 05/19/2023).

[122] NASA. *Space Technology Roadmaps and Priorities Revisited*. Tech. rep. The National Academies Press, 2016. DOI: 10.17226/23582. (Visited on 06/05/2023).

[123] Brian Israel. “Space Governance 3.0”. In: *Georgia Journ. of Int. & Comp. Law* 48.3 (2020), p. 16. ISSN: 0046-578X. URL: https://digitalcommons.law.uga.edu/gjicl/vol48/iss3/7.

[124] Lesley Conn. *Global Space Economy Nears $447B*. https://www. thespacereport.org/uncategorized/gl space-economy-nears-447b/. 2021. (Visited on 10/16/2022).

[125] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. “A Survey and Analysis of Multi-Robot Coordination”. In: *International Journal of Advanced Robotic Systems* 10.12 (Dec. 2013), p. 399. ISSN: 1729-8814. DOI: 10.5772/57313. (Visited on 04/18/2020).

[126] Shasha Li, Xiaodong Bai, and Songjie Wei. “Blockchain-Based Crowdsourcing Framework with Distributed Task Assignment and Solution Verification”. In: *Hindawi Security and Communication Networks* 2022 (Jan. 2022). Ed. by Zhili Zhou, p. 16. ISSN: 19390114. DOI: 10.1155/2022/9464308.

[127] Ryosuke Abe et al. “Fabchain: Managing Audit-Able 3D Print Job over Blockchain”. In: *IEEE Int. Conf. Blockchain and Cryptocurrency*. Shanghai, China: IEEE, May 2022, p. 9. ISBN: 978-1-66549-538-7. DOI: 10.1109/ICBC54727.2022.9805519. (Visited on 03/02/2023).

[128] Shahla Ghobadi and John D’Ambra. “Coopetitive Knowledge Sharing: An Analytical Review of Literature”. In: *Elec. Jour. of Know. Manag.* 9.4 (2011), pp. 307–317.

[129] Zee Zheng et al. *SpaceChain - Community-Based Space Platform*. Website. Sept. 2018. URL: https://spacechain.com/ (visited on 12/08/2022).

[130] Yalda Mousavinia et al. *Space Decentral: A Decentralized Autonomous Space Agency*. 2012.

[131] Liudmila Zavolokina et al. “Incentivizing Data Quality in Blockchains for Inter-Organizational Networks - Learning from the Digital Car Dossier”. In: *Proc. Int. Conf. Inf. Syst.* San

Francisco, USA: AIS, Dec. 2018, p. 18. ISBN: 978-0-9966831-7-3. DOI: 10.5167/uzh-157909.

[132] Yara Rizk, M. Awad, and E. Tunstel. “Cooperative Heterogeneous Multi-Robot Systems: A Survey”. In: *csur* 52 (2019), pp. 1–31. DOI: 10.1145/3303848.

[133] Thomas M. Roehr, Florian Cordes, and Frank Kirchner. “Reconfigurable Integrated Multirobot Exploration System (RIMRES): Heterogeneous Modular Reconfigurable Robots for Space Exploration”. In: *Journal of Field Robotics* 31.1 (2014), pp. 3–34. ISSN: 15564959. DOI: 10.1002/rob.21477. (Visited on 10/15/2022).

[134] U. S. P. Srinivas Aditya et al. “A Survey on Blockchain in Robotics: Issues, Opportunities, Challenges and Future Directions”. In: *Journal of Network and Computer Applications* 196 (Dec. 2021), p. 10. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103245. (Visited on 05/31/2022).

[135] Ph. Saint Aubert et al. “Centralized vs decentralized options for a european data relay satellite system”. en. In: *Acta Astronautica* 13.6-7 (June 1986). 0 citations (Crossref) [2022-12-08], pp. 387–401. ISSN: 0094-5765. DOI: 10.1016/0094-5765(86)90093-7. URL: https://www.sciencedirect.com/science/article/pii/0094576586900937 (visited on 12/08/2022).

[136] Renan Lima Baima et al. *Designing Trustful Cooperation Ecosystems is Key to the New Space Exploration Era*. arXiv:2402.06036 [cs, eess]. Feb. 2024. DOI: 10.1145/3639476.3639760. URL: http://arxiv.org/abs/2402.06036 (visited on 02/27/2024).

[137] Alessandro Farinelli, Luca Iocchi, and Daniele Nardi. “Distributed on-line dynamic task assignment for multi-robot patrolling”. In: *Autonomous Robots* 41.6 (2017), pp. 1321–1345. ISSN: 0929-5593. DOI: 10.1007/s10514-016-9579-8. URL: https://doi.org/10.1007/s10514-016-9579-8 (visited on 09/01/2023).

[138] Michael W. Otte, Michael J. Kuhlman, and Donald A. Sofge. “Auctions for multi-robot task allocation in communication limited environments”. en. In: *Autonomous Robots* 44.3-4 (Mar. 2020). tex.ids= otteAuctionsMultirobotTask2020, otteAuctionsMultirobot-

Task2020b, pp. 547–584. ISSN: 0929-5593, 1573-7527. DOI: 10.1007/S10514-019-09828-5. URL: http://link.springer.com/10.1007/s10514-019-09828-5.

[139] David Wettergreen et al. “Design and Experimentation of a Rover Concept for Lunar Crater Resource Survey”. en. In: *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. Orlando, Florida: American Institute of Aeronautics and Astronautics, Jan. 2009. ISBN: 978-1-60086-973-0. DOI: 10.2514/6.2009-1206. URL: https://arc.aiaa.org/doi/10.2514/6.2009-1206 (visited on 08/30/2023).

[140] N. Roy et al. “Coastal navigation-mobile robot navigation with uncertainty in dynamic environments”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. Vol. 1. ISSN: 1050-4729. May 1999, 35–40 vol.1. DOI: 10.1109/ROBOT.1999.769927.

[141] George Hart. “Goldberg Polyhedra”. en. In: *Shaping Space: Exploring Polyhedra in Nature, Art, and the Geometrical Imagination*. Ed. by Marjorie Senechal. New York, NY: Springer, 2013, pp. 125–138. ISBN: 978-0-387-92714-5. DOI: 10.1007/978-0-387-92714-5_9. URL: https://doi.org/10.1007/978-0-387-92714-5_9 (visited on 09/13/2023).

[142] *A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems*. en. DOI: 10.1177/0278364904045564. URL: https://journals.sagepub.com/doi/epdf/10.1177/0278364904045564 (visited on 08/28/2023).

[143] Robert Michael Zlot. “An auction-based approach to complex task allocation for multirobot teams”. en. PhD Thesis. Carnegie Mellon University, The Robotics Institute, 2006. DOI: 10.5555/1292932. URL: https://www.ri.cmu.edu/pub_files/pub4/zlot_robert_michael_2006_2/zlot_robert_michael_2006_2.pdf (visited on 06/21/2024).

[144] L.E. Cánez, K.W. Platts, and D.R. Probert. “Developing a framework for Make-or-buy decisions”. In: *International Journal of Operations & Production Management* 20.11

(Jan. 2000), pp. 1313–1330. ISSN: 0144-3577. DOI: 10.1108/01443570010348271. URL: https://doi.org/10.1108/01443570010348271 (visited on 06/07/2023).

[145] Robert A. Goehlich. *Make-or-Buy Decisions in Aerospace Organizations*. en. Wiesbaden: Gabler, 2009. ISBN: 978-3-8349-1530-6. DOI: 10.1007/978-3-8349-9479-0. URL: http://link.springer.com/10.1007/978-3-8349-9479-0.

[146] Renan Lima Baima et al. "Trustful Coopetitive Infrastructures for the New Space Exploration Era". In: *PROCEEDINGS OF THE 2024 ACM SYMPOSIUM ON APPLIED COMPUTING*. Ávila, Spain: ACM, Feb. 2024. DOI: 10.1145/3605098.3635887. URL: http://arxiv.org/abs/2402.06014 (visited on 02/27/2024).

[147] Juan Benet. *IPFS - Content Addressed, Versioned, P2P File System*. July 2014. DOI: 10.48550/arXiv.1407.3561. arXiv: 1407.3561. (Visited on 06/06/2023).

[148] Brian Yamauchi. "Frontier-based exploration using multiple robots". en. In: *Proceedings of the second international conference on Autonomous agents - AGENTS '98*. Minneapolis, Minnesota, United States: ACM Press, 1998, pp. 47–53. ISBN: 978-0-89791-983-8. DOI: 10.1145/280765.280773. URL: http://portal.acm.org/citation.cfm?doid=280765.280773 (visited on 08/31/2023).

[149] Michael K. Barker et al. "Improved LOLA elevation maps for south pole landing sites: Error estimates and their impact on illumination conditions". In: *Planetary and Space Science* 203 (2021), p. 105119. ISSN: 0032-0633.

[150] Laurent Perron and Vincent Furnon. *OR-tools*. May 2024. URL: https://developers.google.com/optimization/.

[151] Loïck Chovet et al. "Coopetitive Lunar Mapping: A distributed non-proprietary Multi-Robot Coordination using Blockchain-based Cost Optimisation". en. In: *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. Bari, Italy: IEEE, Oct. 2024, pp. 1276–1282. DOI: 10.1109/CASE59546.2024.10711434. URL: https://ieeexplore.ieee.org/abstract/document/10711434 (visited on 11/08/2024).

[152] Steven Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* 7.66 (May 2022). Publisher: American Association for the Advancement of Science, eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/10.1126/scirobotics.abm6074 (visited on 05/25/2023).

[153] Vincenzo DiLuoffo, William R Michalson, and Berk Sunar. “Robot Operating System 2: The need for a holistic security approach to robotic architectures”. en. In: *International Journal of Advanced Robotic Systems* 15.3 (May 2018). Publisher: SAGE Publications, p. 1729881418770011. ISSN: 1729-8806. DOI: 10.1177/1729881418770011. URL: https://doi.org/10.1177/1729881418770011 (visited on 07/12/2023).

[154] Wen-Yew Liang, Yuyuan Yuan, and Hsiang-Jui Lin. *A Performance Study on the Throughput and Latency of Zenoh, MQTT, Kafka, and DDS*. arXiv:2303.09419 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2303.09419. URL: http://arxiv.org/abs/2303.09419 (visited on 07/12/2023).

[155] Mihail Sichitiu. “Wireless mesh networks: opportunities and challenges”. In: (Apr. 2023).

[156] *eProsima Fast DDS Performance*. URL: https://www.eprosima.com/index.php/resources-all/performance/eprosima-fast-dds-performance (visited on 07/03/2023).

[157] *Fast DDS vs Cyclone DDS Performance*. URL: https://www.eprosima.com/index.php/resources-all/performance/fast-dds-vs-cyclone-dds-performance (visited on 07/03/2023).

[158] *Minimizing Discovery Overhead in ROS2*. URL: https://zenoh.io/blog/2021-03-23-discovery/ (visited on 07/04/2023).

[159] ZettaScale Technology. *Improving the communications layer of robot applications with ROS 2 and Zenoh*. 2022. URL: https://www.youtube.com/watch?v=1NE8cU72frk (visited on 07/04/2023).

[160] Yixuan Chen et al. “Non-Contact In-Vehicle Occupant Monitoring System Based on Point Clouds from FMCW Radar”. en. In: *Technologies* 11.2 (Apr. 2023). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 39. ISSN: 2227-7080. DOI: 10.3390/technologies11020039. URL: https://www.mdpi.com/2227-7080/11/2/39 (visited on 07/03/2023).

[161] M. Fotheringham and D. R. Paudyal. “COMBINING TERRESTRIAL SCANNED DATASETS WITH UAV POINT CLOUDS FOR MINING OPERATIONS”. English. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-4-2021 (June 2021). Conference Name: XXIV ISPRS Congress ¡q¿Imaging today, foreseeing tomorrow¡/q¿, Commission IV - 2021 edition, 5–9 July 2021 Publisher: Copernicus GmbH, pp. 129–138. ISSN: 2194-9042. DOI: 10.5194/isprs-annals-V-4-2021-129-2021. URL: https://isprs-annals.copernicus.org/articles/V-4-2021/129/2021/ (visited on 07/03/2023).

[162] Juan Jesús Roldán Gómez and Antonio Barrientos, eds. *Multi-Robot Systems: Challenges, Trends and Applications*. MDPI - Multidisciplinary Digital Publishing Institute, Mar. 30, 2022. ISBN: 978-3-0365-2846-5. DOI: 10.3390/books978-3-0365-2847-2. URL: https://www.mdpi.com/books/reprint/5252-multi-robot-systems-challenges-trends-and-applications (visited on 03/29/2024).

[163] *Distributed Robotic Systems in the Edge-Cloud Continuum with ROS 2: a Review on Novel Architectures and Technology Readiness*. ar5iv. URL: https://ar5iv.labs.arxiv.org/html/2211.00985 (visited on 03/29/2024).

[164] *MikroTik*. en. URL: https://mikrotik.com/ (visited on 11/10/2023).

[165] *ER-X-SFP Quick Start Guide*. URL: https://dl.ubnt.com/qsg/ER-X-SFP/ER-X-SFP_EN.html (visited on 11/10/2023).

[166] *Leo Rover — Robot Developer Kit — Open-source :ROS and for outdoor use.* en. URL: https://www.leorover.tech/ (visited on 07/04/2023).

[167] *Introducing the Intel® RealSense™ Depth Camera D455*. en-US. URL: https://www.intelrealsense.com/depth-camera-d455/ (visited on 11/10/2023).

[168] Maggie Peng. *RPlidar A2 - SLAMTEC Global Network Lidar Sensor RPlidar A2 cheap lidar scanner*. en-US. URL: https://www.slamtec.ai/home/rplidar_a2/ (visited on 11/10/2023).

[169] *NVIDIA Jetson Xavier Series*. en-us. URL: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/ (visited on 11/10/2023).

[170] *HWMPplus (Mesh) - RouterOS - MikroTik Documentation*. URL: https://help.mikrotik.com/docs/pages/viewpage.action?pageId=8978441 (visited on 04/12/2024).

[171] *ROS 2 Alternative middleware report - General*. en. Section: General. Sept. 2023. URL: https://discourse.ros.org/t/ros-2-alternative-middleware-report/33771 (visited on 11/09/2023).

[172] Vahid Khalilpour Akram and Moharram Challenger. "Connectivity Maintenance in IoT-based Mobile Networks: Approaches and Challenges". In: Sept. 2021, pp. 145–149.

[173] Theodore Rappaport. *Wireless Communications–Principles and Practice, Second Edition*. Prentice Hall, 2001.

[174] Bradford H. Parkinson et al. *Lunar Positioning, Navigation, and Timing: A Proposed Architecture and Potential Challenges*. Technical Report NASA/TP–20210014049. NASA, 2021. URL: https://ntrs.nasa.gov/citations/20210014049 (visited on 02/18/2025).

[175] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580.

[176] Jukka Suomela. "Computational Complexity of Relay Placement in Sensor Networks". en. In: *SOFSEM 2006: Theory and Practice of Computer Science*. Ed. by David Hutchison et al. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 521–529. ISBN: 978-3-540-31198-0. (Visited on 03/17/2025).

[177] David E. Smith et al. “Initial observations from the Lunar Orbiter Laser Altimeter (LOLA)”. In: *Geophysical Research Letters* 37.18 (2010). ISSN: 1944-8007.

[178] Michael K. Barker et al. “Improved LOLA elevation maps for south pole landing sites: Error estimates and their impact on illumination conditions”. In: *Planetary and Space Science* 203 (Sept. 2021), p. 105119. ISSN: 00320633.

[179] Clearpath Robotics. *Husky Spec Comparison*. URL: https://clearpathrobotics.com/husky-spec-comparison/ (visited on 01/20/2025).

[180] Clearpath Robotics. *Jackal UGV - Small Weatherproof Robot - Clearpath*. URL: https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/ (visited on 01/20/2025).

[181] *clearpathrobotics/clearpath_simulator*. Jan. 2024. URL: https://github.com/clearpathrobotics/clearpath_simulator (visited on 02/09/2024).

[182] David Rodríguez-Martínez et al. “Enabling Faster Locomotion of Planetary Rovers With a Mechanically-Hybrid Suspension”. In: *IEEE Robot. and Automat. Letters* (Nov. 2023), pp. 1–8.

[183] Tim Stevens. *Hakuto’s lunar rovers cross the sand in Hamamatsu*. Jan. 2015. URL: https://www.cnet.com/pictures/google-lunar-xprize-hamamatsu-pictures/ (visited on 02/09/2024).

[184] Shreya Santra et al. “Risk-Aware Coverage Path Planning for Lunar Micro-Rovers Leveraging Global and Local Environmental Data”. In: *2024 International Conference on Space Robotics (iSpaRo)*. 2024, pp. 42–47.

[185] Loick Chovet et al. “Performance Comparison of ROS2 Middlewares for Multi-robot Mesh Networks in Planetary Exploration”. In: *J. of Intell. Robot. Syst.* (2025). ISSN: 0921-0296.

Appendices