REALMS2 - Resilient Exploration And Lunar Mapping System 2 – A Comprehensive Approach

Dave van der Meer[1]*, Loïck P. Chovet[1]*, Gabriel M. Garcia[1]*,
Abhishek Bera[1] and Miguel Angel Olivares-Mendez[1]

***Abstract*— The European Space Agency (ESA) and the European Space Resources Innovation Centre (ESRIC) created the Space Resources Challenge to invite researchers and companies to propose innovative solutions for Multi-Robot Systems (MRS) space prospection. This paper proposes the Resilient Exploration And Lunar Mapping System 2 (REALMS2), a MRS framework for planetary prospection and mapping. Based on Robot Operating System version 2 (ROS 2) and enhanced with Visual Simultaneous Localisation And Mapping (vSLAM) for map generation, REALMS2 uses a mesh network for a robust ad hoc network. A single graphical user interface (GUI) controls all the rovers, providing a simple overview of the robotic mission. This system is designed for heterogeneous multi-robot exploratory missions, tackling the challenges presented by extraterrestrial environments. REALMS2 was used during the second field test of the ESA-ESRIC Challenge and allowed to map around 60% of the area, using three homogeneous rovers while handling communication delays and blackouts.**

I. Introduction

Recently, the Moon has regained the focus of space agencies and private companies for potential In-Situ Resources Utilisation (ISRU). Therefore, the European Space Agency (ESA) and the European Space Resources Innovation Centre (ESRIC) seek to increase the level of autonomy of robotic systems used for the exploration of space resources. ESA and ESRIC organised the Space Resources Challenge [1], where 13 research teams competed in a first field test to demonstrate their concepts of autonomous systems, leveraging the advantages of Multi-Robot Systems (MRS). The five best teams continued to a second field test [2] with the task of finding different resources within a large lunar analogue environment, shown in Fig 1.

Fig. 1. Second field test arena for the ESA-ESRIC Space Resources Challenge

During the first field test of the Challenge [2], the authors present the Resilient Exploration And Lunar Mapping System (REALMS) [3], a MRS using two rovers mapping the environment with Visual Simultaneous Localisation And Mapping (vSLAM). REALMS is based on the Robot Operating System (ROS) to communicate with other nodes on the robots. The first part of the challenge revealed multiple limitations of REALMS [3]. First, the ROS framework is designed for centralised systems and provides only limited support for independent MRS. Second, the robustness of the communication system needs to be improved. Third, REALMS offers limited scalability due to the design of the user interface and due to the high network load that severely limits the downlink budgets.

This paper proposes Resilient Exploration And Lunar Mapping System 2 (REALMS2), an improved version of REALMS that addresses these limitations. The revised architecture has increased coverage and resilience based on the Robot Operating System version 2 (ROS 2) framework that provides improved MRS support, making each robot a fully standalone system.

The contributions of this work are:

- A fully decentralized and scalable MRS architecture built on ROS 2, enabling modular deployment, fault tolerance, and seamless multi-robot coordination in communication-constrained lunar-like environments.
- A mesh-based communication layer using HWMP+, validated in integration with ROS 2, allowing dynamic relay formation and resilience to network degradation or loss.
- A complete field-validated system (REALMS2) tested during the ESA-ESRIC Challenge, including live map merging capabilities, bandwidth-aware robot control, and single-operator mission management under simulated lunar constraints.

The paper is structured as follows: Section II highlights

This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references 14783405, 17025341 and 17679211. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission. This project was partly funded by the ESA-ESRIC Space Resources Challenge, Contract No. 4000137334/22/NL/AT.

[1]Affiliated at Space Robotics (SpaceR) Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg {dave.vandermeer, loick.chovet, gabriel.garcia, abhishek.bera, miguel.olivaresmendez}@uni.lu

* Authors share equal contribution.

the state-of-the-art in planetary robotics. Section III briefly shows the system setup. Section IV describes the system validation through a series of experiments. Section V shows the use case during the ESA-ESRIC Space Resources Challenge. Section VI presents the results of the experiments and our participation in the challenge. Sections VII and VIII highlight the lessons learned from REALMS2 and its application in the challenge.

II. Related Work

Most planetary robotics missions consisted of a lander and a single robot. However, new approaches highlight MRS as the correct approach to ensure resilience, range, and cost efficiency as shown by recent projects such as NASA's CADRE [1] or Ingenuity [2] with Perseverance mission.

This section highlights the existing work in the field of MRS along with space-related applications. It also details the Mesh Network technology and ROS 2 used for MRS communication

A. Multi-robot systems

The authors of [4] distinguished four MRS architectures:

- Centralised: A single central node is in charge of operating all the robots. It offers a more optimised approach since all the knowledge is centralised, but it can lead to network bottlenecks, and any malfunction on the central node can lead to an overall failure.
- Hierarchical: This approach features smaller centralised systems coordinated by bigger central nodes in a hierarchical way. This allows for more stability toward failure since each central node maintains the capability to operate its fleet.
- Decentralised: Each robot is in charge of making its own decisions, depending on the data transmitted by the other robots. It is the most robust approach; however, it leads to suboptimal results and might lead to more network load because all robots need to gather data from the rest of the system.
- Hybrid: This approach combines the previous methods to achieve robustness with efficiency.

Space missions have unique characteristics that should impact the choice of architecture. In the case of a centralised architecture, a master node is required to handle all robot tasks and to respond quickly to emergencies. However, there is a delay in communication between the Earth and any astronomical object. This delay would force the master node to be hosted directly on the same celestial body. The key to success in space missions is redundancy, and having one master node would lead to a single point of failure, potentially affecting the whole system. According to these rules, the architecture targeted for space MRS should be at least partially a decentralised architecture, allowing enough redundancy and resilience. With this logic in mind, the LUNARES [5] project was presented. The researchers implemented a team of heterogeneous robots in charge of autonomous data sampling. One of the key elements of the system is the Ground Control Station, in charge of supervising critical operations. This study mainly demonstrates the capabilities of each robot, along with their ability to collaborate. ARCHES [6], which focused on the exploration of lava tubes, presented a more recent approach. ARCHES featured a drone for the first time. It demonstrated the growing interest in decentralised heterogeneous MRS for space applications.

B. Mesh Networks

Mesh networks represent a network topology in which every node can act as a relay, further propagating the network. It was introduced in the IEEE 802.11s norm [7] 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) [8]. It offers the following advantages :

- **Robustness**: Since every robot is creating and propagating the network, if any robot fails, rerouting is dynamically triggered.
- **Scaling**: Any additional node can be dynamically added to the system. This can lead to a better **coverage** with the addition of relay nodes
- **Redundancy**: Multiple robots can ensure a link. It ensures fail-safe behaviour, which is needed for any space mission.

The IEEE 802.11s norm [9] also introduces a mesh networking protocol called Hybrid Wireless Mesh Protocol (HWMP). It introduces a metric called airtime, which is used to define the best way to communicate messages. The authors of [10] studied issues related to HWMP. They highlighted that the airtime metric did not take into account traffic flow, leading to suboptimal routing. They also show that HWMP leads to more data overhead. To solve these issues, they introduce HWMP+, implementing two main improvements. They change the way the Packet Error Rate (PER) is computed to get a better estimation of the link quality. The second change is an observation of the traffic load on the link, impacting the computation of the metric.

Other approaches to mesh networking exist, such as AODV [11] and BATMAN [12]. These approaches have some advantages; For instance, they are compatible with any wireless device, in contrast to HWMP, which must adhere to IEEE 802.11s standards. However, these approaches also feature issues, such as the routing protocol not estimating the link quality.

III. Methods and Materials

We present 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.

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

[2] The ingenuity helicopter on the perseverance rover

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.

A. Rover architecture

The REALMS2 rovers consist of a variety of hardware and software components chosen for this specific mission of lunar exploration.

1) Hardware: The rovers are made of Commercial-of-the-Shelf (COTS) components. For a real lunar mission, space-grade hardware would be required. The robotic base is a Leo Rover [13], designed as a space-rover analogue platform. It features a native ROS 2 implementation running on a Raspberry Pi 4 board. The upgraded robots for REALMS2 feature many improvements, as visible in Fig. 2. To ensure complete mapping capabilities, each robot is equipped with an RPLIDAR A2M8 along with a RealSense D455 RGB-D camera. The sensors allow the robots to perceive their surroundings and provide insightful data, such as colour images, for future analysis.

The robot is equipped with a more powerful processing board, the NVIDIA Jetson Xavier NX, to process this amount of data. This board features a GPU, which ensures proper processing of the large amount of data provided by the sensors. The second board, the Raspberry Pi 4, is only in charge of the robot's control. This ensures independence between motor control and sensor processing so that the operator can always maintain control.

Each robot is equipped with a Mikrotik Groove 52ac router. This device corresponds to an antenna and a router capable of running the HWMP+ protocol. This allows each robot to act as a node in the mesh network, distributing the load and allocating the bandwidth more efficiently. Since every robot is active in the network, it can serve as a relay by optimising its position to extend the system's range.

Two different rechargeable batteries power the robots. The first is the default built-in battery that powers the internal Raspberry Pi and the motor controller to power the motors. The second battery is from a cordless drill and is used to power the Mikrotik antenna. This router connects all the onboard computers with the Mikrotik antenna, the Jetson Xavier and, therefore, the 2D LiDAR and the IntelRealsense camera.

2) Software: As shown in Figure 3, the lower part of the diagram illustrates the internal components and their interfaces for using the light and the wheels of the rover. This internal system is managed by ROS. Furthermore, the upper part of the diagram is composed of external components, including SLAM (RTAB-Map), map merging (M-Explore), and navigation (Nav2) capabilities, while ensuring communication with the operator. The external system is managed by the ROS 2. A ROS-ROS 2 bridge makes the communication between ROS and ROS 2 data possible.

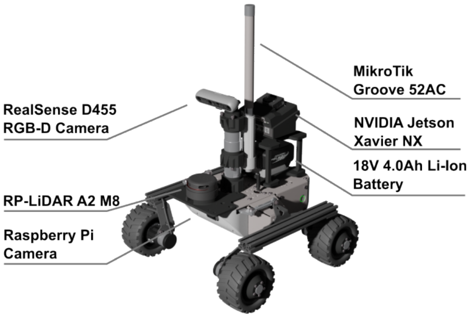

Fig. 2. Leo Rover setup used for REALMS2

ROS 2 is the successor to ROS, addressing many of its limitations by incorporating modern software architecture principles. By design, ROS is a fully centralised system relying on a master node, and making it decentralised would require using external packages, increasing the risk for potential issues [3]. ROS 2 relies on the Data Distribution Service (DDS) standard [14] for inter-node communication, which improves its real-time communication capabilities, security, and scalability.

However, ROS 2 still has some deficiencies compared to ROS, especially in terms of a more expensive computational load and data overhead due to the Quality of Service (QoS) policy, which ensures the quality of the messages [15].

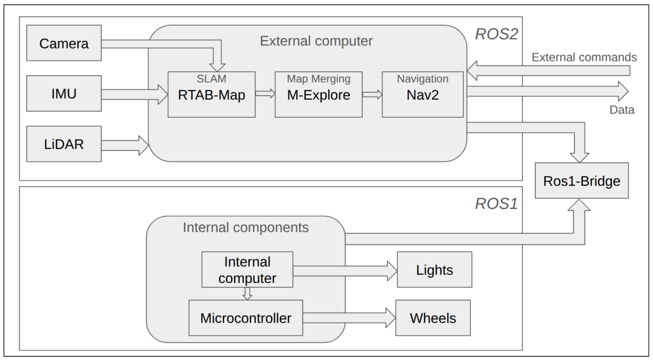

Fig. 3. REALMS2 global system overview

The entire navigation stack, which consists of two main components, is implemented on the Nvidia Jetson Xavier. A vSLAM software, in charge of the mapping and localisation of the robot in its environment. The vSLAM software used is Real-Time Appearance Based Mapping (RTAB-Map) [16] allowing proper sensor merging at a loop closure. The second component is the autonomous navigation software. Once the robot knows its surroundings and position, it needs to know how to reach its next goal. The software suite nav2 [17] includes all of the sub-elements to convert any goal received by the robot to a given set of commands. To improve the portability and streamline the startup sequence, the software is implemented in Docker containers. This simplifies the deployment of the entire architecture to a new robotic platform while ROS 2 ensures interoperability. Additionally, Docker containers can be configured to start immediately at the boot process of the robot, shortening the starting sequence.

A simulated communication delay of two seconds roundtrip time will affect the communication between the lander and the ground station. This challenge needs to be addressed by the ROS 2 message system. An additional ROS 2 package is implemented, the map merging package [18], in order to simplify the user interface, merging all the different maps as one. However, it requires each map to have at least 20% of overlap.

B. Lander

The addition of a Lunar lander is an improvement to REALMS. It consists of an Intel NUC, Core-i7 CPU, 32GB of RAM, a camera, and LiDAR on a static platform. The primary usage of the lander is to act as a network gateway between the ground station and the MRS. This recreates a setup similar to a real mission, where a lander would be the only device with an antenna powerful enough to communicate with Earth.

Additionally, the embedded computer in the lander can act as an edge computing device, providing more computing power than the robots. As a result, the lander can handle global tasks such as merging maps and backing up data. Finally, the lander has a LiDAR and a camera to provide the robot operators with first insights into the environment. With this information, the operators can make an exploration plan to optimize the mission time and focus on valuable resources.

C. User Interface

The operator controls the rovers using a front-end system using a graphical user interface (GUI). A back-end system forwards the commands and handles the communication with the different rovers.

1) Frontend: Foxglove Studio [19] with a custom panel is used as frontend. REALMS uses RVIZ as a user interface. It is the default visualisation software for ROS; it integrates well in the ROS ecosystem and is known by most users in the robotics community. However, Foxglove provides some benefits, such as a more intuitive interface for creating custom panels. The GUI allows the use of panels for different types of information and tabs to group panels together to provide a better overview to the operator.

Fig. 4 shows the frontend of REALMS2 using Foxglove. The first panel (red outline) shows the panel properties and enables the adjustment of the settings for the selected panel. The second panel (yellow outline) shows the custom robot selection panel, which allows the operator to choose which rover to send the commands. This panel is robot-agnostic and scans the network for all available robots by filtering them through their namespaces. The custom panel continuously looks for new namespaces in the system and adds them to the selection options. The backend will use the selected robot name as a namespace to forward all the commands to the respective rover. It also contains buttons to turn on or off the lights of the rover, reset the odometry or reboot the rover. The third panel (green outline) shows the camera views of each of the three rovers. The operator can supervise all rovers at the same time and interact if necessary. The fourth panel (blue outline) contains a small teleoperation panel that allows the movement of the selected rover through the GUI. The fifth panel (orange outline) shows the 3D view of the interface, displaying the 3D poses of the robots, the 2D projections of their respective maps, and the 3D point clouds of each rover. The map merging of REALMS2 allows the user to see each of the three maps in a single viewport. The operator can send autonomous navigation goals by clicking at a position in the 3D viewport, and the backend will forward the waypoint to the respective rover. The use of Foxglove with the custom panel and the backend system enhances the scalability and usability of the system and makes it easier for a single operator to control multiple rovers.

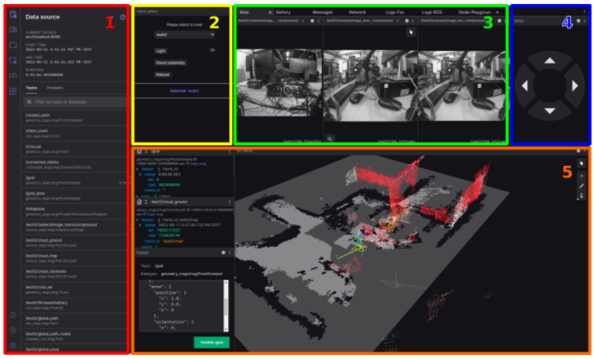

Fig. 4. Foxglove Studio - Main interface

2) Backend: The main function of the backend is to forward information from the rovers to the frontend and from the frontend back to the selected rover. REALMS requires at least one operator per robot. This means that two operators are required to control two rovers. During the ESA-ESRIC Challenge, the number of operators in the control room is limited to five people. REALMS2 implements a back-end system to handle multiple robots through a single interface. This allows the system to be more scalable.

The backend runs on the ground station computer and supports six features from the frontend: rover selection, teleoperation, headlight control, odometry reset, rover reboot, and network monitoring. The operator needs to select the rover to be controlled in the frontend. The backend takes this parameter and forwards all the commands from the frontend to the selected robot by using the robot's name as the namespace for the commands. If no rover has been selected, the namespace cannot be derived, and the backend does not send any commands to avoid unintended behaviour.

The primary function of REALMS2 is to send instructions to move the rovers. These commands can serve as position goals for autonomous navigation or as teleoperation instructions for manually controlling the rovers. In addition to forwarding commands to the selected robots, the backend monitors them, collects data and calculates bandwidth usage. This monitoring method does not create additional overhead on the network.

IV. Subsystem Tests

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.

A. 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

1) 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. 5. 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 meters. The chosen environment featured a direct line-of-sight (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 [20], a crater would need to have a diameter exceeding 5 km, and sometimes up to 20 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.

2) Mesh Network Routing Capability: To verify the capabilities of the mesh network to relay communication, the robot has been placed at the edge of the communication reach of 220 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 m away of the operator and 100 m of the teleoperated rover, as visualised in the lower half of Fig. 5. 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.

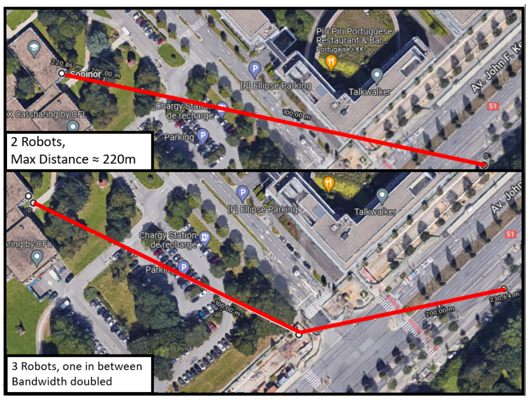

Fig. 5. Two graphs displaying the positions of the robots during the network evaluation. On the top, only two robots are used to measure the range. On the bottom one, a relay is placed in the middle, ensuring a better bandwidth

3) 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 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.

B. Mapping System Testing and Evaluation

The mapping of the system is done using RTAB-Map [16]. Each robot creates a local map. These local maps are then merged into a single global map using the *m-explore* package [18].

1) Mapping the lunar surface: The mapping system is based on REALMS [3]. It proved to successfully handle the extreme lighting conditions of a simulated lunar surface. Furthermore, lab experiments show that mapping of unstructured terrain at a speed of $0.05\ m/s$ is possible without loss of odometry. In the event of odometry loss, the mapping system provides an interface to reset the odometry of the robot to a known position, which can be recovered from the last known position of the robot. In the worst-case scenario, a new map needs to be created and added to the global map through the map merging system.

2) Online Map Merging: To test the map merging capabilities of REALMS2, two rovers have been placed in an outdoor environment. They start next to each other and are controlled through teleoperation to map the environment using vSLAM. At the start, none of the rovers is aware of the position of the other rovers. After mapping the environment with a large overlap of the mapped area, the two maps are matched against each other using a map merging algorithm. Fig. 6 shows the two maps and how their features are matched. During this experiment, the two rovers are controlled using the user interface discussed in subsection III-C. The operator could see the camera streams of both rovers and the maps generated by both rovers.

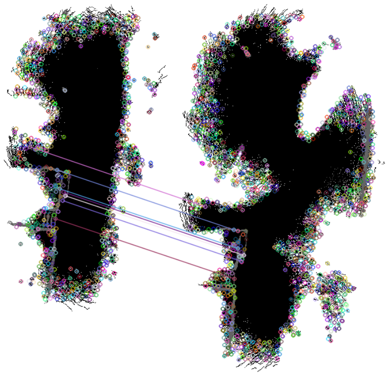

Fig. 6. Map merging - matches between two maps covering the same area

V. ESA-ESRIC Challenge

The final round of the ESA-ESRIC Challenge can be considered as the field test of REALMS2.

The challenge took place in an indoor environment in Esch-Belval, Luxembourg. It featured an exploration zone of 50 m by 36 m that was unknown and not accessible to the operators before the challenge. The area, Fig. 1, featured various assets, such as a lander, craters, boulders, potential resources, and a floor made of finely grinded basalt. The control room was connected to the lander through a network delay simulator as visible in Fig. 7. The goal was to explore the area and identify valuable resources during the 4-hour run.

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.

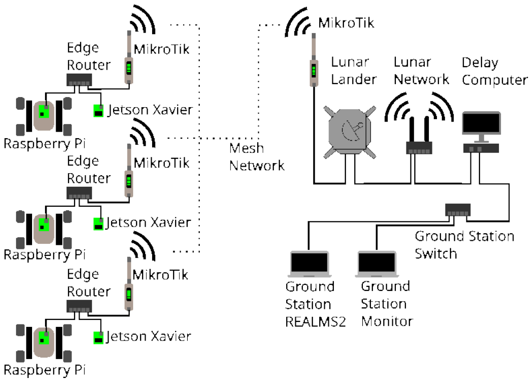

Fig. 7. Overview of the REALMS2 Network Architecture

VI. Results

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. 8 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. This coverage is slightly below the expected results, as the performance was severely impacted by mission-specific conditions, including restricted operator interaction time, failure injection, and the need to revert to manual teleoperation after the autonomous navigation stack failed to execute planned paths. Notably, this situation highlighted the **resilience** of the system, able to continue performing its tasks despite the various difficulties.

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.

VII. 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

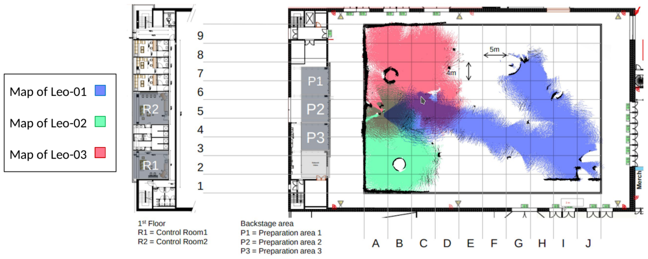

Fig. 8. Area Mapped by REALMS2 during the ESA-ESRIC Space Resources Challenge

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.

VIII. Conclusion

This work presents a complete architecture of a distributed MRS based on ROS 2. The SLAM capabilities of the system were enhanced, and improved resilience was demonstrated through the use of a mesh network. In this setup, each robot acts as an independent relay for data and commands. The use of Docker allows for an increase in modularity and thus improves the scalability of the system. In addition, this approach increases the efficiency of the exploration, allowing multiple rovers to cover a larger area. This architecture has been tested in a lunar analogue facility, the LunaLab, and deployed in an expansive lunar environment created for the ESA-ESRIC Space Resource Challenge. During the final round of the challenge, REALMS2 successfully mapped around 60% of the environment under realistic constraints, including induced communication blackouts and a forced system failure. Despite the fallback to manual teleoperation, the system maintained continuous operation, highlighting its robustness and capacity to adapt to degraded conditions.

Acknowledgment

The authors thank Space Application Services and their team in Belgium with special thanks to Jeremi Gancet, Fabio Polisano and Matteo de Benedetti for making the participation in the ESA-ESRIC Space Resources Challenge possible. The authors also want to thank ESA and ESRIC with special thanks to Massimo Sabatini, Franziska Zaunig, Thomas Krueger and Bob Lamboray for organising this event and hosting all team members in Esch-sur-Alzette, Luxembourg during the challenge. The authors thank Prof. Kazuya Yoshida from the Space Robotics Lab at the Tohoku University.

References

[1] ESA - European Space Agency. *1ST FIELD TEST - ESA-ESRIC Challenge*. https://www.spaceresourceschallenge.esa.int/general-3. Accessed on 2023-06-28. 2023.

[2] ESA - European Space Agency. *2ND FIELD TEST - ESA-ESRIC Challenge*. https://www.spaceresourceschallenge.esa.int/copy-of-1st-field-test. Accessed on 2023-06-28. 2023.

[3] D. van der Meer et al. “REALMS: Resilient exploration and lunar mapping system”. In: *Front. Robot. AI* 10 (2023). DOI: `frobt.2023.1127496`.

[4] Lynne E. Parker. “Multiple Mobile Robot Systems”. In: *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, pp. 921–941. ISBN: 9783540239574. DOI: `10.1007/978-3-540-30301-5_41`.

[5] 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`.

[6] 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 Robot. and Automat. Letters* 5.4 (Oct. 2020), pp. 5315–5322. ISSN: 2377-3766. DOI: `10.1109/LRA.2020.3007468`.

[7] 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`.

[8] 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`.

[9] IEEE - Standards Association. *802.11s-2011 - IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking*. 2011.

[10] 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`.

[11] C. Perkins, E. Belding-Royer, and S. Das. *Ad hoc On-Demand Distance Vector (AODV) Routing*. Tech. rep.

RFC3561. RFC Editor, July 2003, RFC3561. DOI: 10.17487/rfc3561. URL: https://www.rfc-editor.org/info/rfc3561 (visited on 07/06/2023).

[12] 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.

[13] Kell Ideas. *Leo Rover - Build and program your own robot*. https://www.leorover.tech/. Accessed on 2023-07-31. 2023.

[14] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. "Exploring the performance of ROS2". In: *Proceedings of the 13th International Conference on Embedded Software*. Pittsburgh Pennsylvania: ACM, 2016, pp. 1–10. ISBN: 9781450344852. DOI: 10.1145/2968478.2968502.

[15] Takuya Azumi Yuya Maruyama Shinpei Kato. "Exploring the Performance of ROS2". In: *Frontiers in Robotics and AI* 10 (2016).

[16] Mathieu Labbe and Francois Michaud. "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation". In: *Journal of Field Robotics* 36.2 (2019). ISSN: 1556-4959.

[17] Steven Macenski et al. "The Marathon 2: A Navigation System". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.

[18] Tim Clephas, Juan Galvis, and Carlos Andrés Alvarez. *m-explore ROS2 port*. https://github.com/robo-friends/m-explore-ros2. Accessed on 2023-09-13. 2021.

[19] Foxglove. *Foxglove Studio*. https://github.com/foxglove/studio. Accessed on 2023-09-13. 2021.

[20] 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).