An Internet of Robotics Things Platform for Connecting Agents in Disaster Scenarios

PhD Thesis Dissertation

(as a monograph)

by

Juan Bravo Arrabal

Under the supervision of:

Dr. Juan Jesús Fernández Lozano

Dr. José Antonio Gómez Ruiz

Mechatronics Engineering Doctoral Programme

Systems Engineering and Automation Department

University of Malaga, Spain

January 13, 2025

AUTOR: Juan Bravo Arrabal

http://orcid.org/0000-0002-4046-1528

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña JUAN BRAVO ARRABAL
Estudiante del programa de doctorado EN INGENIERÍA MECATRÓNICA de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: AN INTERNET OF ROBOTICS THINGS PLATFORM FOR CONNECTING AGENTS IN DISASTER SCENARIOS

Realizada bajo la tutorización de JUAN JESÚS FERNÁNDEZ LOZANO y dirección de JOSÉ ANTONIO GÓMEZ RUIZ Y JUAN JESÚS FERNÁNDEZ LOZANO.

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 13 de ENERO de 2025

Fdo.: JUAN BRAVO ARRABAL Doctorando	Fdo.: JUAN JESÚS FERNÁNDEZ LOZANO Tutor
Fdo.: DIRECTORES DE TESIS: JUAN JESÚS FERNÁNDEZ LOZANO, JOSÉ ANTONIO GÓMEZ RUIZ	

Edificio Pabellón de Gobierno. Campus El Ejido. 29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

CARTA DE AUTORIZACIÓN Y DECLARACIÓN SOBRE LAS PUBLICACIONES QUE AVALAN LA TESIS DOCTORAL

Programa de doctorado: Ingeniería Mecatrónica

Doctorando: Juan Bravo Arrabal

Tutor: Juan Jesús Fernández Lozano

Directores: Juan Jesús Fernández Lozano y José Antonio Gómez Ruiz

Por la presente, declaramos que las siguientes publicaciones que avalan la tesis doctoral no se han utilizado previamente como aval de ninguna otra tesis doctoral:

Bravo-Arrabal, J.; Fernandez-Lozano, J.J.; Serón, J.; Gomez-Ruiz, J.A.; García-Cerezo, A. "Development and Implementation of a Hybrid Wireless Sensor Network of Low Power and Long Range for Urban Environments". **Sensors** 2021, 21, 567. https://doi.org/10.3390/s21020567.

Juan Bravo-Arrabal, Pablo Zambrana, J.J. Fernández-Lozano, J.A. Gomez-Ruiz, Javier Serón, Alfonso García-Cerezo. "Realistic Deployment of Hybrid Wireless Sensor Networks based on ZigBee and LoRa for Search and Rescue Applications". in **IEEE Access**, vol. 10,pp.64618-64637,2022,doi: 10.1109/ACCESS.2022.3183135.

Bravo-Arrabal, J.; Toscano-Moreno, M.; Fernandez-Lozano, J.J.; Mandow, A.; Gomez-Ruiz, J.A.; García-Cerezo, A. "The Internet of Cooperative Agents Architecture (X-IoCA) for Robots, Hybrid Sensor Networks, and MEC Centers in Complex Environments: A Search and Rescue Case Study". ***Sensors*** 2021, *21*, 7843. https://doi.org/10.3390/s21237843.

Asimismo, autorizamos la presentación de la tesis doctoral.

En Málaga, a 13 de enero de 2025.

Fdo. Juan Jesús Fernández Lozano
Tutor/Director de Tesis

Fdo. José Antonio Gómez Ruiz
Director de Tesis

Dedicado a la memoria de mi abuela,
María Hoyos Mérida
(14/12/1924 - 02/12/2008):
porque la vida es un **b**idón,
donde cabe siempre lo importante y sobra espacio,
para que al echarlo al mar del olvido,
flote, y siga siendo visible.

Agradecimientos - Acknowledgements

To all the people who have supported, guided, and accompanied me throughout all these terrific years of learning, especially when I needed it most. Anything is possible with will, patience, humility, perseverance, and less sleep.

Thanks to my parents for giving me opportunities and pushing me forward when everything went against the tide. Thanks to my wife, Cristina, for encouraging and loving me in all my states and making me smile daily. I feel fortunate to have you by my side. Thanks to Rocío, Roberto, and Emma for their affection and positive influence. Also, thanks to my *gulden draken*: Pepe (these days in Dublin), JC (some days in Nîmes), and Mario (these days in Brussels). They were all by my side from the beginning to the end.

Thanks to my thesis supervisors, Jesús and Janto, for their trust and support, both at work and personally. I was lucky to be supervised by two excellent professionals, but above all, great people. You have been my special robotic crutch for five years.

Thanks to Alfonso for teaching me the path of research and allowing me to be part of an exceptional group of people with whom I co-worked before and after the pandemic in LAENTIEC. Serón, Gandarias, Pastor, Zambrana, Lin, Toscano, Sánchez-Montero, Vera, Padial, and Góngora are some of the surnames that have left a positive mark on me. Thanks also to my colleagues at the Telma Institute, especially to Álvarez-Merino. Also, I thank my brilliant students for their effort and perseverance during their respective work supporting this thesis: Cantizani, Mudarra, and Córdoba.

Of course, it is a tiny but honored space for the fantastic people who accompanied me during my stay in the SDU (Odense, Denmark). They taught me that engineering has no limits, that ambition can be generous, and that constant effort is the only way. Special mention goes to Christian Schlette for his exquisite treatment while guiding my work. Above all, I would like to thank Iñigo Iturrate and Zhuoqi Cheng for welcoming me as one of their own and broadening my vision.

I learned something from their perspectives, and even the minor details made me grow as a researcher and person.

Abstract

Field robotics must cope with complex and unpredictable conditions that must be measured and processed to share only critical information. The rise of Multi-Access Edge Computing (MEC) has enabled a direct connection between diverse infrastructures capable of producing and consuming data in the application scenario. Distributing the analysis and storage of information between the network edge (robots' operational environment) and the cloud (virtual platform providing services) allows decisions to be made and executed in real time, i.e., avoiding critical time lags so that information travels less and reaches its destination sooner. Robots are designed to perform a wide range of tasks without direct human supervision, such as navigation, obstacle avoidance, or Simultaneous Localization and Mapping (SLAM). This implies a high degree of autonomy that requires a diverse payload based on heterogeneous sensory nodes, ad-hoc processing, and communication units. Managing Big Data in real time is not feasible, especially when infrastructures with MEC nodes are unavailable. Therefore, it is necessary to research and develop new architectures capable of integrating distributed algorithms whose outputs can be efficiently communicated between edge devices and the cloud.

This thesis addresses the co-design of perception and telecommunication networks for heterogeneous robots. The proposed co-design is based on Edge-Cloud implementations to support cooperation between human and robotic agents in an emergency, exploring solutions for Search and Rescue (SAR) in Remote, Outdoor, Unstructured, and Disaster (ROUD) scenarios. Such environments lack resources due to isolation, difficult access, or poor structuring characteristics. Therefore, designing and deploying an agile Internet of Robotic Things (IoRT)-based service platform capable of interconnecting agents while performing their SAR tasks is necessary. The Internet of Cooperative Agents (IoCA) architecture, capable of executing algorithms in a distributed manner, serving rescuers and isolated victims in a disaster, is proposed. This work presents results in realistic scenarios in collaboration with emergency professionals, promoting the design and development of heterogeneous perception and control systems that improve the situational awareness of human and robotic agents.

Resumen

La robótica de campo debe hacer frente a condiciones complejas y, a menudo, variables que deben medirse y procesarse para compartir únicamente información crítica. El auge del multi-procesamiento en el extremo de las redes (MEC) ha permitido una conexión directa entre diversas infraestructuras capaces de producir y consumir datos en los entornos operativos. Distribuir el análisis y el almacenamiento de la información entre el borde de la red (entorno operacional de los robots) y la nube (plataforma virtual provista de servicios) permite tomar decisiones y ejecutarlas en tiempo real, es decir, evitar desfases críticos para que la información viaje menos y llegue antes a su destino. Los robots están diseñados para realizar una amplia gama de tareas sin supervisión humana directa, como la navegación, la evitación de obstáculos, o la localización y mapeo simultáneos. Esto implica un alto grado de autonomía que requiere una carga útil diversa basada en nodos sensoriales heterogéneos, unidades ad-hoc de procesamiento y de comunicación. Gestionar Big Data en tiempo real no tiene sentido, especialmente cuando no se dispone de infraestructuras con nodos MEC, por lo que es necesario investigar y desarrollar nuevas arquitecturas capaces de integrar algoritmos distribuidos cuyas salidas se comuniquen de forma eficiente entre los dispositivos en el Edge y el Cloud.

Esta tesis aborda el codiseño de redes de percepción y telecomunicación para robots heterogéneos. El codiseño propuesto se basa en implementaciones Edge-Cloud para favorecer la cooperación entre agentes humanos y robóticos, explorando soluciones para la búsqueda y rescate (SAR) en escenarios ailados, al aire libre, no estructurados, y en los que haya ocurrido una catástrofe. Dichos entornos carecen de recursos debido a su aislamiento, difícil acceso o escasas características de estructuración. Por ello, se requiere diseñar y desplegar de manera ágil una plataforma de servicios basada en el *Internet de las Cosas Robóticas* (IoRT) capaz de interconectar agentes mientras desarrollan sus tareas SAR. Se propone la arquitectura del *Internet de los Agentes Cooperativos* (IoCA), capaz de ejecutar algoritmos de forma distribuida, dando servicio a rescatadores y víctimas aisladas en una catástrofe. Este trabajo presenta resultados en escenarios realistas en colaboración con profesionales del sector de las emergencias, promoviendo el diseño y desarrollo sistemas de percepción heterogéneos que mejoren la conciencia situacional de los agentes, de tipo humano y róbotico.

Contents

List of Figures

List of Tables

Nomenclature

ABP Activation By Personalization

AES Advances Encryption Standard

BLE Bluetooth Low Energy

BSc Bachelor of Science

CN Concentrator Node

CPE Customer-Premises Equipment

CRA Compact Robotic Arm

DDS Data Distribution Service

DEM Digital Elevation Model

DSSS Direct Sequence Spread Spectrum

DT Digital Twin

EDs End-Devices

FTM Fine Time Measurement

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

H-WSN Hybrid Wireless Sensor Network

HA Human Agents

HRI Human-Robot Interaction

IaaS Infrastructure as a Service

IMECH Institute of Mechatronics

IoCA Internet of Cooperative Agents

IoRT Internet of Robotic Things

IoT Internet of Things

ISA Ingeniería de Sistemas y Automática

ISM Industrial, Scientific, and Medical

ITU International Telecommunication Union

JEMERG Jornadas de Emergencias

KPI Key Performance Indicator

LAENTIEC Laboratory and Area for Experimentation in New TechnologIes for Emergencies and Catastrophes

LAN Local Area Network

LiDAR Light Detection and Ranging

LNS LoRa Network Server

LoS Line-of-sight

LPWAN Low Power Wide Area Network

MAC Medium Access Control

MAVs Micro Aerial Vehicles

MEC Multi-access Edge Computing

MQTT Message Queue Telemetry Transport

MRS Multi-Robot System

MSc Master of Science

MU-MIMO Multi-User Multiple-Input Multiple-Output

MULE Mobile Ubiquitous LAN Extensions

NTRIP Networked Transport of RTCM via Internet Protocol

OTAA Over the Air Activation

PaaS Platform as a Service

PAN Personal Area Network

ROS Robot Operating System

ROUD remote, outdoor, unstructured and disaster

RTT Round-Trip Time

SAE Society of Automotive Engineers

SAR Search and Rescue

SCADA Supervisory Control and Data Acquisition

SF Spreading Factor

SG Sensory Groups

SLAM Simultaneous Localization and Mapping

TTN The Things Network

UE User Equipment

UMA University of Malaga

UME Unidad Militar de Emergencias

USAR Urban Search and Rescue

USB Universal Serial Bus

WAN Wide Area Network

WSN Wireless Sensor Networks

ZC Zigbee Coordinator

ZED Zigbee End-Devices

ZR Zigbee Routers

Chapter 1

Introduction

1.1 Background and motivation

This doctoral thesis has a solid practical character and all the results have been obtained in a series of realistic disaster response exercises that took place in the Laboratory and Area for Experimentation in New TechnologIes for Emergencies and Catastrophes (LAENTIEC) in Malaga (Spain). Thus, the contributions of this PhD thesis are the results of an active participation in all the International Conference on Security, Emergencies, and Catastrophes (JEMERG[1]) organized by the Chair of the same name, from the University of Malaga (UMA), between 2019 and 2024. The Department of Systems Engineering and Automation (ISA[2]) collaborates each year in JEMERG, coordinated by the chair coordinator, Dr. Jesús Miranda. The LAENTIEC operational area, spanning a usable surface of $115000m^2$, serves as the outdoor scenario for the JEMERG framework, characterized by an irregular terrain profile and dense vegetation in certain areas due to the proximity of a stream. A Command, Communication, and Control Area (CCCA) composed of various tents is mounted in the LAENTIEC area for supporting SAR teams. Every year, our research group provides novel services through heterogeneous robots from the Forward Control Center (FCC), prepared inside the ISA tent in the CCCA. This scenario has witnessed the results presented in this predoctoral research.

Since its first edition in 2006, JEMERG has become a reference event, gathering around 200 people in a multidisciplinary field exercise conducted in a realistic setting, including first responders, such as fire brigades, healthcare groups, the national police, and military teams such as the air force, the Military Emergency Unit (UME), and the Legion. Other participants include civil protection personnel, victim stand-ins (played by drama students),

[1] JEMERG: Jornadas de Seguridad, Emergencias y Catástrofes.

[2] ISA: Ingeniería de Sistemas y Automática.

authorized media, and visitors. Additionally, research groups and companies specializing in field robotics collaborate on various scenarios within JEMERG to develop new technologies that enhance Search and Rescue (SAR) tasks.

The research groups at the Institute of Mechatronics (IMECH) from UMA, including the UMA Robotics and Mechatronics Team (integrated in the ISA Department), provide their resources for these annual exercises, where the aim is to test and assess emerging technologies in development and research, working alongside actual responders in SAR missions. The tasks are performed under realistic conditions, adhering to strict time constraints and safety protocols. These missions involve searching for and rescuing individuals in emergency scenarios caused by environmental factors, terrorism, or natural disasters like fires, earthquakes, and floods.

Exteroceptive data are fundamental in disaster robotics, where robots must understand their operational environment. In particular, a mobile robot has to perceive information about its surroundings as it navigates throughout an area of interest, which may be isolated and therefore without coverage. Communicating this information back to the CCCA becomes a challenge, especially in environments with degraded or unavailable communication infrastructure.

The measured data are of a different nature including environmental (e.g., temperature, radiation, humidity, pressure and wind speed), chemical (e.g., gas concentration), geometric (e.g., reference frames, including distances and orientations between them), temporal (e.g., duration of a path, communication latencies, data rate, and sampling periods for sensors), and physical (e.g., forces, torques, light, sound, and power consumption).

Acquiring and processing this information, along with its proper formatting, is another challenging task. The data must be efficiently compressed and securely transmitted without compromising its quality, ensuring both data integrity and remote accessibility across diverse application scenarios. Furthermore, exploration mobile robots, including rovers and SAR robotic platforms, require situational awareness to effectively interact with their surroundings. Situational awareness empowers these robots to make informed decisions based on their spatial location on a map and the real-time conditions of their immediate environment. Situational awareness goes hand in hand with positioning and telecommunications.

In disaster scenarios such as wars or natural catastrophes, effective positioning and communication systems are critical for coordination and decision-making in SAR missions. However, these systems face significant challenges under such conditions.

All Global Navigation Satellite Systems (GNSS) provide geolocation capabilities without requiring an internet connection. For instance, a smartphone can record its GNSS position even in airplane mode. While this independence from terrestrial infrastructure makes GNSS

invaluable in disaster scenarios, its reliability can be compromised. Non-line-of-sight (LoS) issues caused by clouds, terrain, vegetation, or artificial obstacles often degrade the accuracy of GNSS signals. Additionally, geomagnetic storms and intentional signal disruptions, such as jamming in military conflicts, can further reduce their effectiveness. These limitations underscore the importance of developing secondary positioning systems to ensure redundancy.

Reliable communication systems are critical in disaster scenarios, enabling coordination, information sharing, and response planning. These systems, which may include satellite, cellular, and radio-based technologies, are especially vital in areas where terrestrial infrastructure, such as WiFi or mobile phone base stations, has been destroyed or is unreliable.

Adverse weather conditions, including heavy rain, snow, and atmospheric disturbances, can degrade signal quality and reliability. This is particularly relevant for systems that rely on electromagnetic waves, as the trade-off between bandwidth and atmospheric resistance often limits performance. It is not only the frequency of upload and download that is important, but also the type of architecture that must manage these communications. For instance, low-frequency bands, such as the L-band (1–2 GHz), are commonly used for robust communications and navigation systems due to their resilience to atmospheric interference, but they offer limited bandwidth. In this case, it is latency that can be prioritized and improved. On the other hand, high-frequency bands, like the V-band (50–75 GHz), provide greater bandwidth but are more susceptible to interference from rain, fog, and other atmospheric conditions. In this case, what must be prioritized and improved is coverage.

Geomagnetic storms add another layer of complexity, causing signal disruptions, communication interruptions, or even damage to satellites and ground-based equipment. These vulnerabilities underscore the need for co-designing multi-layered and resilient communication and positioning systems, especially in areas where terrestrial infrastructure are likely to fail.

Obstacles can severely hinder the flow of information, including sensor data essential for robotic systems to perceive and interact with their environment. Ensuring continuous connectivity is essential, as robots depend on both soft and hard real-time data, as well as synchronous and asynchronous communications, to build situational awareness, share information, and perform tasks autonomously.

To address these challenges in disaster scenarios, research is needed on the design, development and practical implementation of architectures capable of supporting perception, tele-communications, distributed processing and location awareness.

This thesis looks at complex scenarios, particularly Remote, Outdoor, Unstructured, and Disaster (ROUD) environments, aiming to enhance the situational awareness of het-

erogeneous mobile robots by considering not only their own status within their operational environment but also the status of their peers. By fostering this shared awareness, robots can better cooperate and coordinate in SAR missions, where time is critical to success.

This cooperative situational awareness involves monitoring each robot's pose (position and orientation in space), integrating sensory data, and sharing relevant information with the rest of the SAR agents, including humans. Such shared data enables robots to anticipate the actions and needs of their peers, optimize task allocation, and adapt to dynamic conditions collectively. This interconnected awareness is vital for ensuring efficient teamwork, minimizing redundancy, and improving mission outcomes in unpredictable and challenging environments.

1.2 Contributions

This thesis aims to boost the cooperation of heterogeneous agents, both robotic and human types, in ROUD environments through the deployment of an Internet of Robotic Things (IoRT) made up of elements (e.g., routers, gateways, actuators, and sensors) of varied power consumption, throughput, latency, and coverage, both onboard and within the robot's operational environment. This work focuses on resource-constrained scenarios addressing the co-design of perception and communication architectures, leveraging implementations across the Edge-Cloud continuum to facilitate the sharing of situational awareness between human and robotic agents in emergencies. The integrated sensory means include Wireless Sensor Networks (WSN) based on various technologies, high-throughput sensors such as Light Detection and Ranging (LiDAR) systems, and modern smartphones. These systems are used to gather information about the agents' environment and their position within it, enabling data synchronization for both offline and online operations. Furthermore, this thesis presents results from realistic scenarios developed in collaboration with professionals from the emergency sector in JEMERG.

The proposed IoRT includes different Hybrid Wireless Sensor Networks (H-WSN), i.e. with static and mobile End-Devices (EDs), including sensor and concentrator nodes, and high bandwidth ad-hoc sensors that integrate various technologies, depending on the needs.

In addition, it is proposed to use low-power networks, particularly Personal Area Networks (PAN) and Low-Power Wide Area Networks (LPWAN) to ensure long-term data acquisition in ROUD scenarios. It is proposed that SAR agents carry PAN and LPWAN EDs. PAN technologies are suggested as a close perception and detection system, while LPWAN technologies are designed to get data from remote areas to the operational environment. Therefore, it is essential to distinguish what is helpful to transmit and how to do it

(format, order, packet size, synchronism, power consumption, radio range, and sensitivity), considering the position of the EDs.

On the other hand, the robot's onboard computing equipment may sometimes require specific algorithms or services that are not locally available, being necessary to share computing resources belonged to other robots operating in the same scenario (edge) or to servers hosted remotely (cloud). While sending raw data might be essential for certain applications, bandwidth limitations often demand local preprocessing to transmit only the most relevant information. Multi-access Edge Computing (MEC) centers address these challenges by intelligently distributing processing tasks across the application scenario, including the onboard computers of deployed robotic agents. This approach helps reduce operational latencies and supports meeting real-time requirements effectively.

This PhD thesis contributes to answering next questions:

- What strategies can be employed to collect information in catastrophic scenarios, where areas of interest may be isolated or lack communication infrastructure, making data acquisition and transmission particularly challenging?
- What mechanisms can be used to securely format, integrate, and distribute information from diverse sources to both mobile robots and human operators in remote locations, ensuring data integrity and enabling effective use in SAR scenarios?
- How can teams of ground and aerial robots be remotely coordinated based on shared situational awareness from the operational area, in order to enhance decision-making during emergency response operations?

Therefore, this work addresses three fundamental aspects in field robotics and, in particular, in ROUD scenarios: i) perception through low-power H-WSN, i.e. with static and mobile nodes; ii) distribution of information and actions through the design and implementation of an Edge-Cloud architecture; iii) augmentation of SA, understood from the approach of the interaction of a robot with its environment, including digital twins.

The integration of H-WSN with high-bandwidth ad-hoc sensors is proposed to enhance the location awareness of robots, ensuring that their pose (position and orientation) remains continuously aligned with environmental measurements. The inclusion of both static and mobile nodes increases system redundancy and flexibility, enabling more effective deployment in dynamic scenarios. This diverse IoRT framework addresses key limitations faced by robots in ROUD environments, such as restricted internet coverage, bandwidth constraints, and latency, which can otherwise hinder their ability to act safely and effectively.

This work involves designing an architecture that seamlessly integrates the Robot Operating System (ROS), in their two versions, and the Message Queue Telemetry Transport (MQTT) protocol, enabling the Internet of Cooperative Agents (IoCA). The IoCA architecture facilitates collaboration between robotic and non-robotic agents (e.g., humans, rescue dogs), focusing on efficient information distribution and managing lightweight data transmission without relying on the Internet. To support this, an in-depth analysis of Personal Area Networks (PAN) and LPWAN technologies is conducted. Additionally, this thesis introduces 5G ad-hoc sensors by integrating modern smartphones into the robot workspace. Finally, the evolution of the IoCA architecture aims to increase the situational awareness of heterogeneous teams.

All in all, the main contributions are as follows:

- **Contribution 1**. Codesign, development, and deployment of H-WSN and soft real-time emergency robotic systems, effectively bridging the traditionally separate fields of telecommunications and field robotics. The H-WSNs are based on low-power technologies (including PAN and LPWAN), utilizing diverse topologies tailored for application and evaluation in ROUD environments under realistic conditions. Designed to operate without Internet connectivity, these networks ensure reliable performance even when traditional infrastructure is unavailable. Additionally, both online and offline multicasting techniques are developed to optimize communication efficiency.

- **Contribution 2**. Design, implementation, and evolution of an Edge-Cloud architecture to integrate a diverse IoRT, including H-WSNs and ad-hoc 5G sensors with low latency and adaptive bandwidth requirements (network slicing), enabling distributed processing of sensor information to operate successfully in scenarios with limited conditions. Key performance indicators such as latency, throughput, packet loss, coverage, quality of service, and processing load (whether distributed near or far from data generation) are carefully regulated to ensure optimal performance. The proposed Internet of Cooperative Agents (IoCA) is implemented on realistic SAR missions in ROUD scenarios.

  By prioritizing the distribution of processing tasks between the edge—represented by the robots' Local Area Networks (LAN)—and the cloud—comprising servers hosted on Wide Area Networks (WAN)—the system ensures reliable and efficient operations even when traditional infrastructure is unavailable.

- **Contribution 3**. Enhancing the situational awareness of heterogeneous agents (including humans, uncrewed ground and aerial vehicles, and robotic manipulators) for

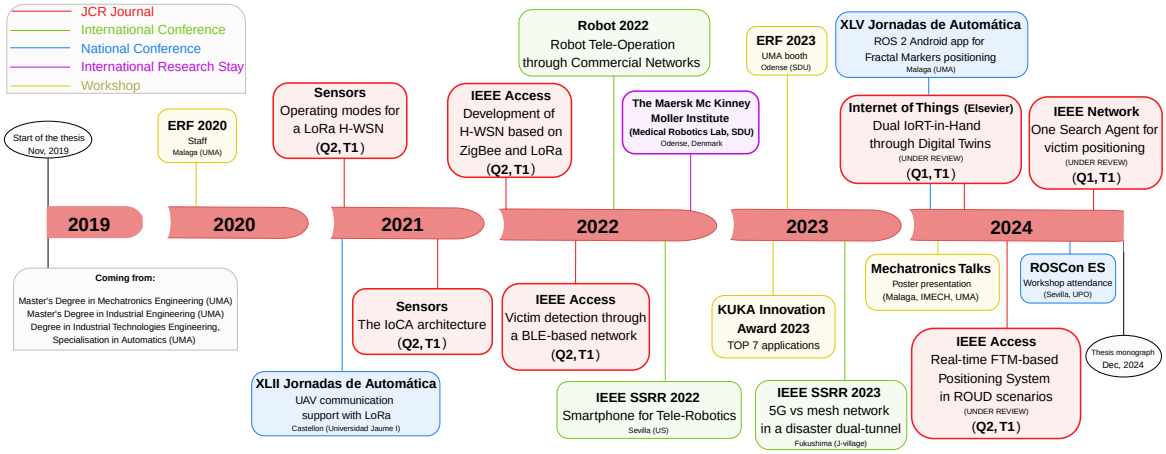


Figure 1.1 Timeline of this predoctoral research

performance in demanding applications requiring soft real-time capabilities, such as SAR and telemedicine, through the use of Digital Twins (DT) and telecontrol. This has been achieved by integrating ROS networks with MQTT clients and brokers.

Figure 1.1 illustrates the timeline of this thesis, highlighting the contributions and publications that have been tested and validated in realistic scenarios, demonstrating their applicability and effectiveness in real-world emergency situations. During this time, I completed a three-month stay at the Maersk Mc-Kinney Moller Institute, working in the Medical Robotics Lab at the University of Southern Denmark (Odense, Denmark), from September 12, 2022, to December 21, 2022. Table 1.1 presents the JCR articles published during the development of this thesis. Meanwhile, Table 1.2 lists the articles presented at national and international conferences in this period. Finally, Table 1.3 shows other JCR papers that have been submitted and are under review, while Table 1.4 includes a preprint, submitted to IEEE Communications Society.

1.3 Context and Scope

After earning a Bachelor (BSc) and Master of Science (MSc) in Industrial Engineering with a specialization in Automation, I pursued a second MSc in Mechatronics Engineering at the University of Malaga (UMA), where I developed a strong interest in research. My theses focused on telecommunications and field robotics. The MSc in Mechatronics Engineering from the UMA corresponds to the training period of the PhD Programme in Mechatronics Engineering delivered by this same University, where I enrolled in November 2019.

Table 1.1 JCR Publications

Work	Title	Authors	Journal	Year	Quartile
1	Development and implementation of a hybrid wireless sensor network of low power and long range for urban environments.	**Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Serón, J., Gómez-Ruiz, J. A., García-Cerezo, A.	Sensors	2021	**T1, Q2**
2	The Internet of Cooperative Agents architecture (X-IoCA) for robots, hybrid sensor networks, and MEC centers in complex environments: a Search and Rescue case study.	**Bravo-Arrabal, J.**, Toscano-Moreno, M., Fernández-Lozano, J. J., Mandow, A., Gómez-Ruiz, J. A., García-Cerezo, A.	Sensors	2021	**T1, Q2**
3	Realistic deployment of Hybrid Wireless Sensor Networks based on Zigbee and LoRa for Search and Rescue applications.	**Bravo-Arrabal, J.**, Zambrana, P., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Barba, J. S., García-Cerezo, A.	IEEE Access	2022	**T1, Q2**
4	Bluetooth Low Energy for close detection in Search and Rescue missions with robotic platforms: an experimental evaluation.	Cantizani-Estepa, J., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Fortes, S., Barco, R., García-Cerezo, A., Mandow, A.	IEEE Access	2022	**T1, Q2**

Table 1.2 Conference Publications

Work	Title	Authors	Conference	Year
1	Sistema de comunicación de respaldo mediante tecnología LoRa con hardware y software abierto para aplicaciones de robótica de emergencias.	Manrique Balmaceda, R. F., Vázquez-Martín, R., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., García-Cerezo, A.	XLII Jornadas de Automática (Castellón, Spain)	2021
2	Integrating ROS and Android for rescuers in a Cloud Robotics architecture: application to a casualty evacuation exercise.	Toscano-Moreno, M., **Bravo-Arrabal, J.**, Sánchez-Montero, M., Barba, J. S., Vázquez-Martín, R., Fernández-Lozano, J. J., García-Cerezo, A.	IEEE SSRR (Sevilla, Spain)	2022
3	Remote planning and operation of a UGV through ROS and commercial mobile networks.	Sánchez-Montero, M., Toscano-Moreno, M., **Bravo-Arrabal, J.**, Serón Barba, et al.	Iberian Robotics Conference (Zaragoza, Spain)	2022
4	Field report on experimental comparison of a WiFi mesh network against commercial 5G in an underground disaster environment.	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Mandow, A., Barco-Moreno, R., García-Cerezo, A. J.	IEEE SSRR (Fukushima, Japan)	2023
5	UR2A: comunicación bidireccional Android-ROS 2 para arquitecturas Edge-Cloud en sistemas robóticos conectados.	Córdoba-Ramos, M., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Mandow, A., García-Cerezo, A.	XLV Jornadas de Automática (Málaga, Spain)	2024

My work within the ISA Department has focused on the Robotics and Intelligent Control Systems research line, contributing to the design and development of software and hardware for emergency robotics. Additionally, I have been involved in multiple 5G projects, where I worked with heterogeneous robots for emergency response and industrial applications. Table 1.5 shows the National Projects that have supported this thesis.

1.4 Outline

This PhD thesis, titled *An Internet of Robotics Things Platform for Connecting Agents in Disaster Scenarios*, has been written as a monograph and divided into six chapters. Except

Table 1.3 Articles submitted to JCR journals (under review)

Work	Title	Authors	Journal	Quartile
1	Real-Time FTM-based Victim Positioning System Using Heterogeneous Robots in Remote and Outdoor Scenarios	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Toscano-Moreno, M., Serón-Barba, J., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Khatib, E. J., Barco, R., García-Cerezo, A.	Under review at IEEE Access (minor revisions requested).	**T1, Q2**
2	Real-Time FTM-based Victim Positioning System for OSA: One Search Agent	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Toscano-Moreno, M., Serón-Barba, J., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Khatib, E. J., Barco, R., García-Cerezo, A.	Submitted to IEEE Network	**T1, Q1**
3	The IoRT-in-Hand: Tele-Robotic Echography and Digital Twins on Mobile Devices	**Bravo-Arrabal, J.**, Cheng, Z., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Schlette, C., Savarimuthu, T. R., García-Cerezo, A.	Submitted to Internet of Things (Elsevier)	**T1, Q1**

Table 1.4 Preprint submitted to IEEE Communications Society

Work	Title	Authors	Journal
1	Strengthening Multi-Robot Systems for SAR: Co-Designing Robotics and Communication Towards 6G	**Bravo-Arrabal, J.**, Vázquez-Martín, R., Fernández-Lozano, J. J., García-Cerezo, A.	Submitted to IEEE Communications Society (preprint: arXiv:2504.01940)

Table 1.5 Research Projects

#	Project Title	Principal Investigators	Duration
1	RTI2018-093421-B-100: Towards resilient teams of UGV and UAV manipulators for search and rescue robotic tasks	Dr. Alfonso José García Cerezo, Dr. Jesús Fernández Lozano	2019-2022
2	8.06/5.56.5582: Evaluation of 5G communications in emergency and rescue applications and assessment of Cloud Robotics using 5G for emergency operations	Dr. Juan Jesús Fernández Lozano, Dr. Alfonso José García Cerezo	2019-2022
3	PID2021-122944OB-100: Reaching a new paradigm in cooperative cyber-physical human-robot systems for search and rescue	Dr. Alfonso José García Cerezo, Dr. Anthony Mandow Andaluz	2022-2024
4	5GVEC: Development of 5G-based technological solutions for the deployment of connected vehicles and validation of use cases	Dr. Ricardo Vázquez Martín, Dr. Juan Jesús Fernández Lozano, Dr. Alfonso José García Cerezo	2023-2025

for the present chapter and the one on conclusions and future work, each chapter begins with an introduction, outlining the scientific problem being addressed, summarizing the challenges, presenting the contributions made, and ending with an evaluation section on the obtained experimental results. Bibliographical references and the appendix are found at the end of the document.

Chapter 2, *Background and State of the Art in Emergency Robotics*, presents a literature review of existing challenges and solutions related to perception and Edge-Cloud architectures for emergency robotics.

Chapter 3, *Perception and Communication via Wireless Networks in Complex Scenarios*, presents the co-design, development, and deployment of low-power networks composed of

static and mobile nodes. The mobile nodes are mounted on mobile units (e.g., a UAV, a SAR dog, or a human). These H-WSNs can operate without Internet connectivity, maintaining a constant service and providing reliable information from ROUD environments, even without traditional infrastructures for long periods, by taking advantage of PAN, WLAN, and LPWAN technologies, integrated into various topologies. In addition, the chapter introduces online and offline multicasting techniques to optimize the efficiency of communications, ensuring the adaptability of systems to changing scenarios. Contribution 1 is explained.

Chapter 4, *Fostering Cooperation between Agents through MEC and Mobile Devices*, describes the development and implementation of the Internet of Cooperative Agents (IoCA) architecture, which arises from the need to relate the services provided and required by the different types of agents deployed in the application scenario, and how their cooperation is facilitated with an intelligent distribution between the edge and the cloud, where remote computing centers or Multi-Access Edge Computing (MEC centers) can provide resources and services based on Network Slicing, location awareness or control needs. In addition, this chapter introduces the development of mobile apps to promote the smartphone in robotics, commenting on their advantages and disadvantages. Contribution 2 is presented.

Chapter 5, *Situational Awareness for Control of Heterogeneous Robots*, responds to how agents interact with their operational environment through targets, thus increasing their knowledge of their workspace. The IoCA architecture has evolved to implement new characteristics, including control and Digital Twins. Contribution 3 is explained.

Chapter 6, *Final Remarks*, highlights the most relevant contributions of this thesis and proposes future research topics which, at least to date and to the author's knowledge, have not been covered in the literature.

Chapter 2

Background and State of the Art in Disaster Robotics

2.1 Introduction

This chapter provides an overview of the research conducted over the past years on field robotics for complex operational environments, which has served as the foundation for this doctoral thesis focused but not limited to remote, outdoor, unstructured and disaster (ROUD) scenarios. The term *remote* does not imply a great distance to the operational environment, where robots are deployed, but often refers to areas where the communication infrastructure is limited or unreliable, which complicates the provision of resources and remote control operations. Figure 2.1 shows some example of this type of scenario, including the situation on the island of La Palma (Canary Island, Spain), when the Tajogaite volcano erupted, causing chaos in nearby towns for 85 days (September 2021), or the floods caused by the phenomenon known in Spain as DANA (an isolated high-level depression), which resulted in the heaviest one-hour rainfall ever recorded in Spain—179.4 L/m^2 in Turís, Valencia—and caused numerous casualties, electricity and communications to be cut off, and the need for the deployment of 50 exploration Uncrewed Aerial Vehicles (UAVs) in the first hours of rescue, as well as more than 10,000 first responders, including military, police, forensic doctors and forest brigades.

All these scenarios have in common the saturation of human assets (SAR teams and victims) due to stress or fatigue, being key the collaboration of volunteers, the difficulty or impossibility of accessing specific hostile locations, the scarcity of infrastructure, which makes stable telecommunications unfeasible, and the need for a rapid response to locate and evacuate victims. In addition, the use of heavy machinery is not allowed, as they would

(a) Russia-Ukraine war (2022-present). [1]

(b) Sampoong Mall (South Korea, 1995). [2]

(c) Tajogaite volcano eruption (Spain, 2021).[3]

(d) DANA in Valencia (Spain, 2024). [4]

Figure 2.1 Some ROUD scenarios

destabilize the structures, risking the lives of rescuers and victims buried in the rubble. Thus, only by hand should debris be removed, while every second counts. With limited access to medical aid, food, and water, the risk of fatalities among victims rises significantly over time. The hazards of these rescue missions, both for the responders and the trapped individuals, can be as severe as those posed by the initial disaster itself.

Emergency robots can enter where a person or SAR dog has no access and gather vital information to assist in rescue efforts. A significant milestone in their deployment occurred on September 11, 2001, during the terrorist attack on the World Trade Center (New York, USA), marking the first use of Urban Search and Rescue (USAR) robots in a real-world crisis [1]. Robots of varying sizes and capabilities were deployed from tethered models to radio-operated units, and in shapes ranging from compact, lunchbox-sized devices to larger, lawnmower-sized platforms. These mobile robots played a crucial role, aiding in the search for survivors and assessing hazardous areas to protect SAR teams. Their presence highlighted the potential of robotic systems to assist in critical, life-saving missions within complex, high-risk environments.

[1]Source: South Asian Voices
[2]Source: Reddit: Catastrophic Failure
[3]Source: Atlas Obscura: La Palma Volcano
[4]Source: Huffington Post: DANA in Spain

2.2 Perception in Complex Environments

2.2.1 Multiple robots and sensors, and all of them heterogeneous

The vision of integrating robots into environments originally designed for human use has sparked significant interest in disaster scenarios, intensifying research efforts in the field of emergency robotics. Normally, the SAR strategy for a post-disaster scenario is designed for human jobs (firefighters, police, military, engineers, nurses, psychologists), but in the last two decades the design of these scenarios is evolving in the context of human-robot interactions (HRI), both at the level of rescuers and victims.

Despite notable progress over the past decades, several challenges still hinder widespread deployment in real-world disaster scenarios. A key priority is the development of robust and user-friendly robotic systems capable of operating under extreme and unstructured conditions. Effective human-robot collaboration is emphasized, with robots seen as tools to enhance, not replace, human capabilities. Intuitive user interfaces—such as gesture-based control—are considered essential to reduce operator cognitive load. The review in [2] also examines various robotic platforms—ground, aerial, maritime, and hybrid—highlighting their respective advantages and limitations in disaster response. Moreover, it addresses the importance of building operator trust, the lack of standardized testing frameworks, and persistent technological barriers such as limited autonomy, navigation in complex environments, and communication issues. Future research directions proposed include the integration of machine learning, multimodal perception, multi-robot coordination, and smart infrastructure to ensure the transition from prototypes to reliable, mission-ready systems.

However, perception in SAR robotics involves additional essential areas of research and development, such as real-time computing, efficient telecommunications, and distributed architectures, each aimed at enabling robots to navigate and interact more intuitively and effectively in ROUD scenarios.

The overarching challenge is to capture and relay high-quality information to the right place efficiently and sustainably while maintaining low power consumption to enable prolonged service in demanding environments.

The use of multi-sensor [3] is encouraged to integrate multiple data sources that provide a richer, more reliable dataset, enabling the robots to make better-informed decisions. This is particularly important in dynamic spaces where single-sensor data may be insufficient or unreliable. Effective sensor fusion allows for robust object recognition, obstacle detection, and environmental mapping, enabling robots to function more autonomously and with greater resilience to environmental noise or signal interference. The combination of different sensors leads to more accurate results, either by knowing the objects in the environment or by

referencing them concerning the changing position of the mobile robot. For instance, the APEIRON work [4] presented a rich multimodal dataset based on efficient sensor-driven perception that synchronously captures data from different sources, enhancing situational awareness by linking vision and network data for robust UAV perception.

Authors in [5] tackle SAR challenges in harsh and unpredictable environments, especially subterranean, stressing the importance of autonomously fusing sensory data in robust missions where human presence is risky.

These mobile platforms often embark on highly sophisticated sensory systems. However, it is not only high-data-rate sensors that robots live on to feed off their environment. To this end, wireless sensor networks (WSNs) are often deployed in the application scenario to provide robotic agents with additional information about their workspace. For example, in [6], the design of a distributed Zigbee-based WSN prototyping system is presented to track mobile SAR robots while searching for the heat of the living human body, showing the possibility to relate the location of the robot, without the need for Global Positioning System (GPS), when it performed a certain victim location. There is, therefore, a need to acquire information from the environment also with sensor equipment other than that deployed on board the robots themselves.

The mobility of ground and aerial robots breaks the limitations of WSNs with respect to target area inaccessibility, battery lifetime, hardware vulnerability, communication and computation capability, and can assist them to achieve higher performance. Both technologies (WSN and mobile robots) can be combined to improve the performance of different applications, such as remote environmental monitoring, navigation, and localization [7].

Authors in [8] present a comprehensive review and some guidelines about how to choose a wireless technology for emergency response. However, there are not enough realistic case studies in the literature.

Another fundamental challenge in field robotics is Simultaneous Localization and Mapping (SLAM), a process in which a mobile robot navigates through an environment from an unknown position while concurrently building a map of its surroundings. This requires the robot to estimate its own location and progressively construct an accurate representation of the environment, enabling it to navigate and operate autonomously in unfamiliar areas. SLAM algorithms are continually refined to provide more accurate and efficient mapping solutions, even in GNSS-denied environments.

Authors in [9] tested a high number of nodes of a WSN as landmarks to improve SLAM estimations using not only direct robot-node range measurements but also internode measurements, which significantly improve map and robot pose estimations accuracy.

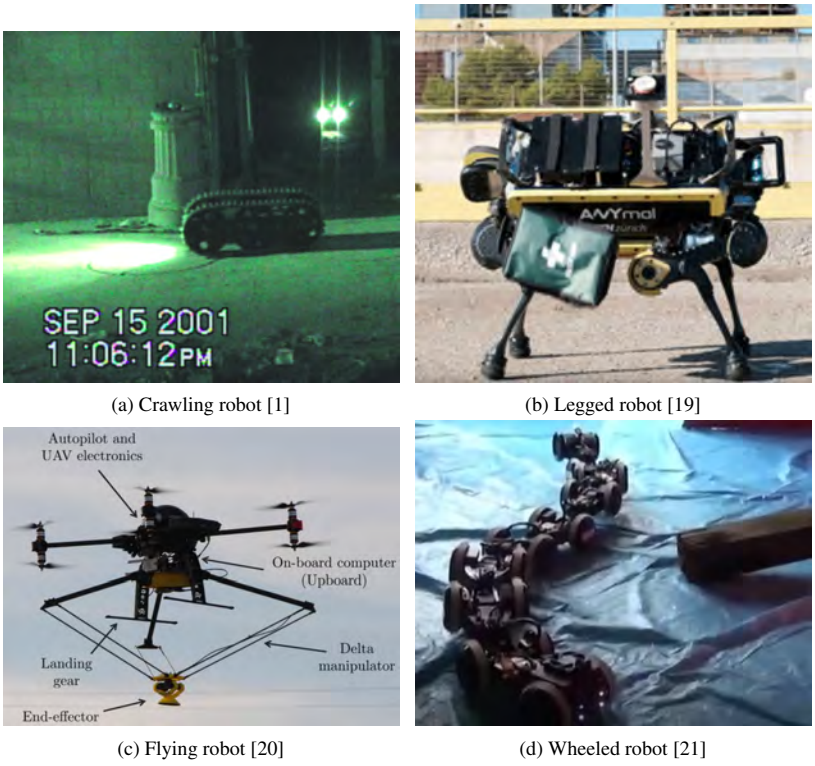
The study in [10] introduces a novel technique for reconstructing 3D objects of interest using a foundational 2D map, thus preserving the high speed of 2D mapping, which is important in post-disaster environments.

Semantic information is also crucial for helping SAR robots interpret their surroundings, facilitating more efficient path planning by generating routes that minimize travel time. Authors in [11] propose a semantic RGB-D SLAM system for use in rescue robots. This system creates detailed point-cloud maps enriched with semantic information. To enhance segmentation accuracy, the authors use depth data to determine if adjacent pixels in the semantic image represent the same object, showing that the system can produce accurate semantic maps and optimize navigation.

SAR scenarios can be also underground where there are multiple challenges in exploration and object localization. In [12], authors introduce COMPRA, a compact and reactive autonomy framework tailored for rapid deployment of Micro Aerial Vehicles (MAVs) in subterranean SAR missions. COMPRA equips an MAV to autonomously explore unknown areas while considering mission-specific criteria, such as object identification, battery life, and mission duration. This framework relies on a reactive local avoidance planner using enhanced potential fields and instantaneous 3D point clouds, alongside a heading regulation technique using real-time depth images. This setup enables the MAV to generate collision-free paths without depending on a global map and can handle situations where localization is imprecise.

All these perception systems are combined in ROUD scenarios to feed location awareness, which refers to the ability of mobile robots to identify and understand their geographical position in their operational environment [13–15]. Location awareness enables robots to interact with and respond to their environment dynamically. It is not limited to knowledge of coordinates but also includes an understanding of context, such as nearby objects, landmarks, or changes in the environment. This capacity for location awarenessfurther supports advanced location-based services, such as real-time navigation, multi-robot coordination, and augmented reality, enhancing both operational performance and user experience (e.g. a tele-operator).

Multi-robot systems (MRS) require a robust and intelligent control system [16] that enables seamless communication and task coordination between heterogeneous robots so that they can work efficiently as a SAR team. The problems of task allocation of MRS are discussed in [17], considering robots have different hardware configurations and physical capabilities (see Figure 2.2), including sensor diversity, actuation mechanisms, computing capacity, and maximum payload. In [18], another example of the effectiveness of MRS can



(a) Crawling robot [1]

(b) Legged robot [19]

(c) Flying robot [20]

(d) Wheeled robot [21]

Figure 2.2 Some locomotion modes in SAR robotics

be found, where two robotic agents cooperate to increase the efficiency of a subterranean exploration.

A comprehensive review of bipedal wheel-legged robots is presented in [22], underscoring the advantages of combining legged and wheeled mobility for improved terrain adaptability, as well as advancements in perception systems, control frameworks, and human-robot interaction (HRI).

Enhancing robots' perceptual capabilities to recognize gestures, facial expressions, and voice commands contributes to smoother, more natural interactions, making robots valuable collaborators in settings where seamless teamwork is essential. This is extended in social robotics [23, 24] but less so in SAR robotics, where uncrewed ground vehicles (UGVs) and uncrewed aerial vehicles (UAVs) are often teleoperated to perform the search part of SAR tasks. However, for the rescue part, interaction with the victims is important [25], e.g. by using advanced Natural Language Processing [26] to increase the efficiency of human-robot communication.

2.2.2 One small step for things, one giant leap for robotics

Energy-efficient perception and processing are crucial for ensuring that robots used in the mentioned literature can operate for extended periods, especially in resource-constrained environments. This involves optimizing the power consumption of both sensors and onboard processing units.

The Internet of Things (IoT) emerged to capture the state of any *thing* and its surroundings in people's lives, giving it access to the Internet to report on its information and make decisions based on various events. This concept has been widely used in multiple applications, such as home automation. The IoT makes people's lives easier by interconnecting their belongings, which have the character of a tagged entity, through the Internet. For example, the IoT informs the *home* thing that the *vehicle* thing will arrive home in 10 minutes to turn on the air conditioning before the person enters. Therefore, IoT provides an online service (in this case, keeping the come cool) to people to improve their quality of life through their belongings.

When a robot (any mechatronic entity with the capacity for perception, processing, and action) is introduced into the online service, being able to exchange information through the Internet, making autonomous decisions (without human intervention) even in more than one aspect, the Internet of Robotic Things (IoRT) arises [27].

Therefore, while both IoT and IoRT contribute to task automation, the distinction lies in the fact that, in IoRT, there is an intelligent entity, capable of physically interacting with the environment and executing more sophisticated tasks than a program managing a single, connected object (IoT). This shift represents an evolution in connectivity: moving from simply connecting smart devices to integrating robots with the ability to manipulate, sense, and respond dynamically to their surroundings, along with their full payload of sensors and tools.

In 2025, vehicles can move tele-operated or autonomously through any medium, including land, air, underwater, or even through the outer space. In any case, they need to carry sensors to capture information about their immediate or distant environment, such as temperature, gases, pressure, sound (shock waves in the case of rovers) and electromagnetic signals (images, radio waves, and so on). Depending on the level of autonomy, a vehicle can be considered a mobile robot, specifically if it integrates advanced sensors to know in detail its environment, control algorithms acting on the sensory feedback to regulate the behavior of the vehicle through its actuators that can be operated, in real time, remotely or through programming.

According to the Society of Automotive Engineers (SAE), there are six levels of driving automation [28], ranging from 0 (fully manual) to 5 (completely autonomous). Based on this

classification, a vehicle can be considered a robot at level 3, where humans must be available to take control if the system requires. These definitions can be applied to any vehicle, not only road vehicles; for example, uncrewed ground or aerial vehicles (UGVs or UAVs) must understand their environment to navigate autonomously and safely. However, robotic systems may still require specific human interventions, such as an emergency stop or teleoperation.

Situational awareness, vital for autonomous robotics, involves a robot's capacity to interpret and respond to its operational environment to perform complex tasks safely. Extensively studied in psychology, military, and aerospace fields, situational awareness in robotics has traditionally focused on perception and interaction [29]. By establishing a continuous, bidirectional flow between the IoRT and Multi-Access Edge Computing (MEC) centers, either onboard (Edge Computing) or on remote servers (Cloud Computing), robots gain the real-time processing and decision-making abilities necessary for true autonomy in dynamic environments.

In general, a WSN in which static and dynamic sensor/actuator nodes or end-devices (EDs) coexist is defined as a Hybrid Wireless Sensor Network (H-WSN) [30]. However, mobile nodes pose a challenge in terms of their integration with WSNs, where effectiveness depends largely on energy consumption, coverage, and connectivity [31, 32].

Following the rationale that a robot's sensors enable it to understand its immediate surroundings, though within a limited range, there are applications where access to broader, geographically and temporally extensive information can be highly beneficial. This capability is precisely what WSNs aim to provide, forming the foundation of the IoT concept and extending it further. While WSNs may offer limited data, they can empower robots with enhanced capabilities by giving them insights beyond their onboard sensors. Here, the transition from IoT to IoRT is evident: a robot acts based on information acquired through WSNs, but unlike IoT devices (such as an air conditioner or automated door), which respond passively, the robot (a physical, autonomous entity) interacts actively and changes the situation (their own and that of their environment). This shift underscores a new level of interaction where IoRT combines the real-time situational awareness of a robot with the expansive reach of IoT networks, allowing for more dynamic, informed, and impactful autonomous actions.

2.2.3 Low Power Technologies for H-WSNs

Table 2.1 shows some low power technologies, including PAN and LPWAN, (and even WLAN), typically used in WSN.

Table 2.1 Low Power Technologies in Wireless Sensor Networks

Technology	Operation Frequency	Transfer Speed	Packet Size	Penetrability	Range (km)
Wi-Fi 6 (2019)	2.4 GHz and 5 GHz	Up to 9.6 Gbps	Up to 2.312 B (802.11ax)	Medium-low	0.1 - 0.2
802.11mc (2018)	2.4 GHz and 5 GHz	Up to 9.6 Gbps (WiFi 6)	2.312 B (802.11ax)	Medium	0.05 - 0.1
NB-IoT (2016)	Within LTE bands	Up to 250 kbps	Up to 1600 bytes	High	10 - 15
LTE-M (2016)	Within LTE bands	Up to 1 Mbps	1-2 KB	High	10 - 15
LoRa (2012)	868 MHz (EU)	0.3-50 kbps	51-222 bytes	Very High	2 - 15
BLE (2010)	2.4 GHz	1 Mbps	Up to 251 bytes	Medium	0.05 - 0.2
Sigfox (2009)	868 MHz (EU)	100-600 bps	12 bytes of payload	High	3 - 50
Zigbee (2003)	2.4 GHz, 868 MHz (EU)	Up to 250 kbps	Up to 127 bytes	Medium	0.075 - 0.2
UWB (2002)	3.1 - 10.6 GHz	1 Gbps	Varies	High (short distance)	0.01 - 0.2

2.2.3.1 Personal Area Networks

Zigbee: concepts and requirements

The Zigbee Alliance was originally formed in 1997 by eight Promoter companies, including Philips, Motorola, Samsung, and Texas Instruments, with the purpose of enabling reliable, cost-effective, low-power, wirelessly networked monitoring and control products [33]. These products were based on an open global standard, with a focus on applications that required long battery life and minimal infrastructure. Zigbee operates on the IEEE 802.15.4 protocol [34], a standard designed to support low-cost, low-data-rate wireless networks with efficient power management, using the Direct Sequence Spread Spectrum (DSSS) to minimize interference and ensure reliable data transmission, even in environments with considerable noise. Zigbee became a key technology for home automation, building automation, and various monitoring and sensor-based applications. In addition, Zigbee allows for a simple and fast sensor network setup [35, 36].

However a Zigbee end-device (ZED) only can transmit and receive small data packets, with a data rate of up to 250 Kbit/s (using 2.4 GHz), within a range of over 10-20 meters depending on visibility conditions.

Its typical data rates range from 20 kbps to 250 kbps, depending on the frequency band, which allows it to maintain a low power consumption profile.

The ZigBee network topology is highly flexible and supports star, tree, and mesh configurations (see Figure 2.3). For applications requiring low latency and high throughput, a star topology is preferable. To cover large areas with a moderate number of devices, a clustered tree topology is best. Finally, for applications that require high robustness and fault tolerance, mesh is the best option, although it compromises performance in terms of throughput and latency. This allows for a scalable and robust system in which devices can

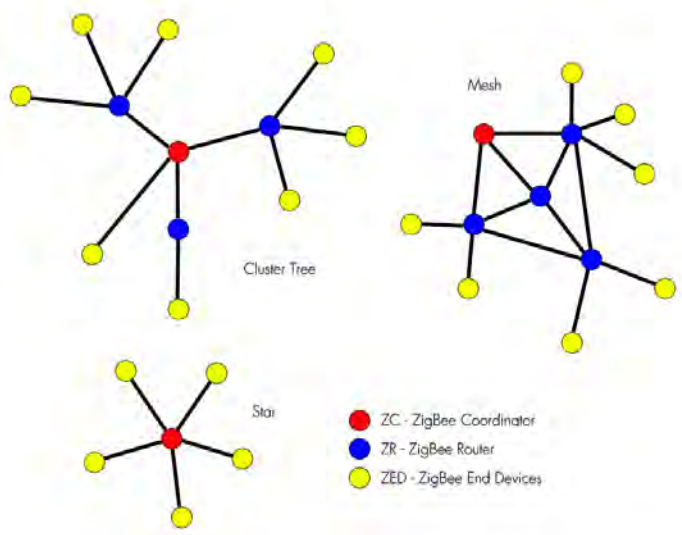


Figure 2.3 Zigbee topologies

dynamically communicate with each other, ensuring reliable data transmission across the network, even in the event of node failures.

Zigbee nodes are devices containing a single 802.15.4 radio, and they can control or monitor multiple things. In a Zigbee network, devices are categorized into three different types of nodes, each with specific roles to ensure the proper functioning of the network:

- Zigbee End-Devices (ZED), designed to run on minimal power, often capable of operating for several years on a pair of AA batteries. They typically remain in a low-power or sleep mode, only waking up when they need to communicate. Due to their low energy consumption, they cannot route data, but they can interact with routers and coordinators to send or receive information.

- Zigbee Routers (ZR) play a critical role in expanding the coverage and capacity of the network by routing packets between devices. Routers maintain a constant connection and help relay messages between different nodes in a mesh topology, ensuring the network's reliability and robustness. ZRs are typically powered by a continuous power source, as they do not have the same low-power constraints as ZEDs.

- The Zigbee Coordinator (ZC) is the most crucial node in the network because it is responsible for forming and managing the entire network. A Zigbee network has exactly one PAN coordinator, and it handles tasks such as assigning addresses to nodes and managing the network's overall structure. The ZC is typically a continuously powered device, and it may also act as a CN, managing communication between multiple devices and performing complex tasks such as data aggregation.

Zigbee supports both unicast (point-to-point) and broadcast (group-based) communications, enabling it to send messages to either a single device or a group of devices within a PAN network. This versatility is useful for various applications, such as controlling multiple devices simultaneously or selectively communicating with a specific node.

One of the key features of Zigbee is its ability to retain settings and network configurations after resets or power outages. This capability ensures that once the network is established, devices can quickly resume their normal operation without requiring reconfiguration, thus improving the overall resilience and usability of the system.

The process of commissioning refers to the initial setup of a Zigbee network, where devices are connected, configured, and authenticated to work together. This process can be complex, involving steps such as assigning addresses, establishing security credentials, and defining communication paths. Once commissioned, the devices in the Zigbee network can operate autonomously, performing the tasks for which they were programmed.

As the IoT continues to expand, Zigbee remains a pivotal technology in applications that require low-power, reliable, and scalable wireless networks. While other standards like Thread, Ultra WideBand (UWB), and Bluetooth Low Energy (BLE) have emerged as competitors in the short-range wireless communication space, Zigbee's robust mesh networking capabilities and mature ecosystem ensure its continued relevance in home automation, industrial control systems, smart lighting, and more.

By leveraging Zigbee's mesh networking, SAR teams can deploy devices that act as communication relays, allowing rescuers and their equipment to stay connected even in difficult terrains or large buildings where direct communication may fail. In [37], authors demonstrated that an artificial rubble in the soil environment is suitable for testing and verifying the wireless network for USAR missions. Attenuations of soil and building materials on 2.4 GHz Zigbee RF signals were measured. The experiments indicated that Zigbee technology implemented by the XBee-PRO modules can form a useful mesh wireless network for USAR robots.

The maximum data throughput is 250 kbps in 2.4 GHz band (see Figure 2.4) and the typical range is 10-20 meters. However, 250 kbps is the raw data rate at PHY level, i.e. at application level the data rate is lower due to protocol stack overhead.

Finally, according to the Zigbee Alliance, Zigbee technology relies upon IEEE 802.15.4, which has excellent performance in low SNR environments, such as SAR scenarios.

Bluetooth: concepts and requirements

In disaster robotics, efficient data collection and communication are critical for successful SAR missions. The increasing prevalence of personal Bluetooth Low Energy (BLE) wear-

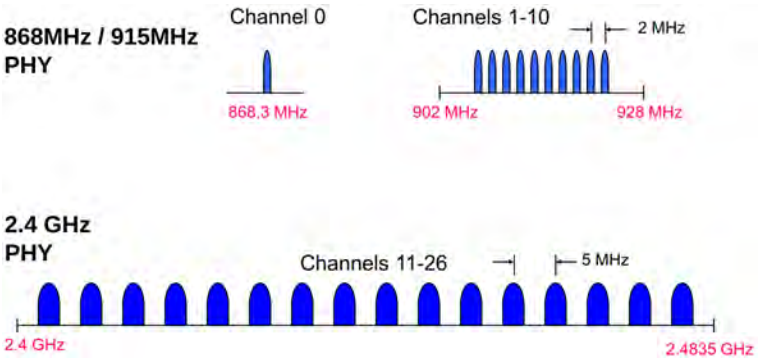


Figure 2.4 Zigbee bands and channels

ables (e.g., smartbands, smartwatches, earbuds) presents an untapped opportunity to use these devices as emergency radio frequency beacons for victim localization, particularly when individuals are unconscious or unable to signal for help.

Time-sensitive SAR operations can greatly benefit from UAVs or UAV swarms equipped with flying networks to determine victim locations [38]. In challenging environments, such as dense forests, UAVs require advanced sensing and trajectory planning to avoid collisions [39]. UAVs operating in formation clusters can adapt their shapes to meet specific task requirements, enhancing their efficiency [40–42]. However, the size and payload limitations of UAVs affect their flight time and maneuverability, emphasizing the importance of lightweight technologies like 5G sensors for high-bandwidth, low-latency communication [43].

Additionally, UAVs and UGVs can serve as mobile network infrastructure in areas without connectivity. For instance, UAVs configured as WiFi Access Points can establish emergency flying networks, capturing signals from devices such as smartphones carried by victims [38].

Research has also explored autonomous UAV motion planning and object detection in outdoor environments under detection uncertainty [44].

The potential of wireless-connected devices for localization has been recognized for a long time. Among these, BLE stands out as the most widespread technology in personal devices, making it particularly suitable for victim detection [45]. BLE localization has been extensively studied, from basic detection to precise localization and identification, with ongoing advancements in hardware and techniques to enhance accuracy.

Most studies on BLE localization focus on indoor environments, employing techniques such as mapping [46] and fingerprinting [47–50], which rely on prior knowledge of the terrain. However, these methods are unsuitable for unknown outdoor environments. An

exception is [51], which uses trilateration and other techniques specifically designed for outdoor scenarios without requiring pre-existing terrain data.

Other works, although less common, that focus on outdoor environments can be found, like [52], where BLE beacons are used in a previously deployed mesh network throughout an outdoor area to allow a moving device to receive their signals and trilaterate its position through RSSI.

Another example of BLE-based outdoor localization is presented in [53], where the positions of cattle are monitored using BLE devices attached to the animals. In this case, outdoor fingerprinting is employed, as the pasture represents a confined area. However, this method is not applicable to emergency scenarios occurring in unknown environments. Similarly, various other outdoor use cases exist where specialized BLE beacons are attached to the entities being tracked (e.g., construction site workers). Nonetheless, none of these studies address the localization of unmodified BLE personal devices belonging to potential victims, considering the constraints of emergency scenarios.

Regarding BLE localization in emergency contexts, some studies focus on locating human victims carrying a BLE device. For instance, [54] describes an approach where a UAV equipped with a BLE detector or a smartphone scans an area to locate a reported victim. This implementation is particularly notable because any registered smartphone can serve as a scanner by running a background routine. However, the victim's device must have been pre-registered in a database, and its lost or injured status reported to the operator to activate search-and-rescue protocols. Other works, such as [55], leverage BLE localization using RSSI to track injured victims who have already been found, rather than focusing on the initial localization phase of these victims.

However, no trials have been reported on detecting and localizing victims in unknown outdoor emergency scenarios where the victims' devices are neither pre-registered in a database nor equipped with purpose-built BLE devices.

2.2.3.2 Low Power Wide Area Networks

In recent years, several technologies have been competing within the Low Power Wide Area Networks (LPWAN) market, a subset of the IoT ecosystem aimed at connecting low-bandwidth, battery-operated devices with low bit rates over extended distances. LPWANs support many sensor nodes connected simultaneously in a wide area (as opposed to PANs), operating for a long time without requiring battery replacements, enabling remote data access and control.

LPWAN operate mainly in the sub-gigahertz electromagnetic spectrum band because they are designated to operate with low energy consumption. These technologies (Sigfox,

LoRa, Weightless, NB-IoT, Wize, LTE-M, or NB-Fi) facilitate widespread connectivity via sensors and actuators without relying on traditional WiFi or cellular networks. They can penetrate obstacles, such as walls or buildings, and achieve long ranges, essential for limited infrastructure. Thus, LPWAN nodes are interesting to be deployed as WSN in large scenarios, such as USAR [56].

However, some LPWAN technologies operate above 1 GHz, such as Random Phase Multiple Access (RPMA), which works exclusively at 2.4 GHz. NB-IoT and LTE-M (LTE for machines) can also operate at 1.8, 2.1, and 2.6 GHz since the cellular infrastructure supports them. These higher-frequency technologies often have greater capacity, which is beneficial for applications requiring more bandwidth and tend to be less susceptible to interference than lower-frequency bands. In addition, some cloud frameworks such as Thigstream, Sigfox Network, The Things Network (TTN), Helium Network, Senet, or Telensa support LPWAN applications offering IoT services.

A second classification for LPWAN concerns whether the technology operates in a licensed band. Historically, governments at the international level have reserved industrial, scientific, and medical (ISM) bands for ISM applications. According to Article 1.15 of the International Telecommunication Union (ITU) Radio Regulations, ISM applications are the operation of equipment or appliances designed to generate and use local radio frequency energy for industrial, scientific, medical, domestic, or similar purposes, excluding applications in the field of telecommunications [57]. Indeed, the home microwave oven, used to cook food at 2.45 GHz, is the most widely used ISM device. Industrial heating and hyperthermia therapy are also big ISM applications.

However, with increasing spectrum congestion and the rapid growth of IoT, today's ISM bands also encompass telecommunications. As a result, these bands are predominantly license-free and can be utilized for non-ISM applications, such as home automation (domotic) or smart agriculture. Nevertheless, telecommunications users in the ISM band must endure interference from ISM equipment and do not have legal protection against the operation of such devices, meaning they do not have priority in using these bands. Therefore, LPWAN technologies are designed to be resilient to interference and to minimize its impact. They are also oriented towards asynchronous communication, as it favors low power consumption: nodes only leave their idle state when they need to transmit or receive information, simplifying data traffic management and favoring network scalability.

Incorporating mobility in large disaster scenarios, such as USAR, is an efficient approach where data Mobile Ubiquitous LAN Extensions (MULE) can play a significant role [58]. Data MULEs are mobile entities (often autonomous vehicles or robots) designed to traverse environments and collect data from stationary or widely dispersed static sensor nodes. These

entities then relay the collected data to a central hub or processing unit. In scenarios like smart cities [59], data MULEs are particularly useful because they can optimize data collection by reducing the need for constant, direct communication between each sensor node and the central server (multi-hop routing [60]), which may otherwise face connectivity issues or excessive latency. Similarly, in large-scale disaster scenarios like USAR, where wireless sensor nodes are deployed at remote points of interest over expansive areas, the integration of mobile robots and WSNs [61] enables efficient data collection from the whole operational environment.

Therefore, mobile nodes pose a challenge in terms of their integration with WSNs, where effectiveness depends largely on energy consumption, coverage, and inter-connectivity [31, 32]. However, it is essential to differentiate between a data MULE, which primarily operates as an intermittent data collector rather than a constant network participant, and a mobile node capable of extending the network and receiving and transmitting information from different points of interest. In the former case, the device retrieves data stored in isolated or distant sensors and then offloads it at a remote data center with high connectivity; while in the second case, we speak of a more sophisticated device, capable of get and transmit data in real time through the network and, at the same time, it can perceive environment data. Thus, hybrid networks are composed of static and mobile sensor and/or concentrator nodes.

Some works [62–64] incorporate UAVs in H-WSN to broaden coverage, model different mobility patterns to predict performance, and emphasize Quality of Service (QoS). However, there is a scarcity of experimental case studies in the literature using LPWAN technologies with significant market potential [65], especially for large-scale and unstructured scenarios, which is important given the disparity between real-world and model-based behaviors [66]. Furthermore, H-WSNs that include at least one mobile node have been underexplored, despite their suitability for applications like urban monitoring and emergency response [67–69].

LoRa: concepts and requirements

LoRa is a long-range, low-rate wireless technology that operates at the physical layer of the Open System Interconnection (OSI) model. It transmits small data packets (up to 256 bytes) within the ISM band, using 868 MHz in Europe. LoRa employs Chirp Spread Spectrum (CSS) modulation, which encodes data in chirps to provide resistance to noise and interference. This modulation technique allows LoRa to divide the frequency spectrum into multiple channels, enabling efficient communication while balancing transmission range and data rate.

LoRaWAN is the network protocol that defines how the EDs interact with the gateways or concentrator nodes (CNs).

A LoRa ED is composed of a sensor or actuator, or both; a radio frequency module with a transmission antenna; a microcontroller to generate the packaged message from sensor readings; and a battery to be wireless and operational for years before needing replacement. By analogy to the OSI model (see Figure 2.5), LoRaWAN, built on top of LoRa modulation, refers to the link layer or Medium Access Control (MAC) layer, which establishes the performance of the network according to its three classes of ED [70–72]. Thus, LoRaWAN defines the message format and how EDs transmit them. In addition, LoRaWAN provides two layers of security: one at the network level to authenticate EDs and another at the application level for end-to-end payload encryption. However, only the payload (not headers) is encrypted at the application layer.

The key distinguishing feature among the three LoRaWAN device classes lies in how each class determines when an ED is allowed to receive data from a server through a gateway:

- **Class A**: All LoRa EDs support this default mode, enabling communication at any time with minimal energy consumption. Communication starts with an uplink (UL) from the ED to the LoRa Network Server (LNS) via a CN, followed by two reception windows (downlinks, DL). The first DL is for acknowledgments (ACKs), and the second can awaken the ED if needed. DLs are only possible after successful ULs, making Class A the most energy-efficient.

- **Class B**: Similar to Class A but includes scheduled reception windows using beacons sent by the CN. These beacons synchronize the EDs and open additional ping slots, allowing the LNS to send messages without waiting for a UL. Class B is suited for applications requiring regular data reception.

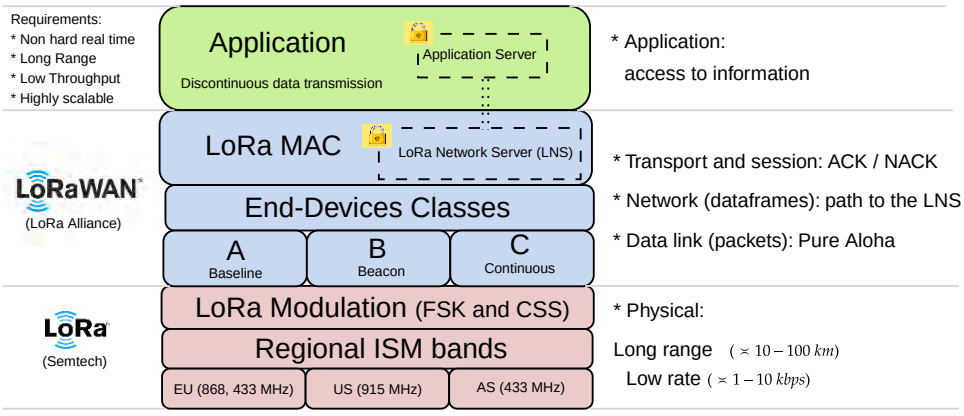


Figure 2.5 LoRa/LoRaWAN stack following the OSI model

- **Class C**: EDs continuously listen for incoming messages, except when transmitting detected events, resulting in the highest energy consumption. This class allows unlimited reception windows and is ideal for applications requiring low-latency communication, prioritizing data transfer over energy efficiency.

All three types of LoRaWAN EDs can coexist in the same WSN, maintaining an unchanged physical layer. Each ED can switch between classes (A, B, or C) depending on the application, but the CN cannot identify the class of an ED transmitting data [73]. Communication is always two-way but half-duplex, i.e., senders can transmit the values acquired by the sensors, and they can also receive data from servers through gateways, such as an acknowledgment of receipt (ACK), but never simultaneously or through the same channel. Thus, an important difference between an ED and a CN is that a CN can listen on multiple frequency channels at the same time.

A LoRa WSN has to comply with specific regulations to operate in the ISM band, which is free but subject to restrictions:

- **Duty Cycle:** The proportion of time an ED spends transmitting is determined by the operational frequency band and transmission requirements, with duty cycle values ranging from 0.1% to 1%. This ensures fair spectrum usage and compliance with regulations. Larger packets require longer transmission times, increasing the risk of collisions and violating duty cycle limits.

- **Signal Gain and Loss:** RF modules and antennas provide signal gain, while cables and connectors introduce signal loss. The Effective Isotropic Radiated Power (EIRP) must not exceed 14 dBm in Europe, as defined by the European Telecommunications Standards Institute (ETSI). Antenna gain (A) must be configured to ensure compliance with this limit.

- **Transmission Power:** The maximum allowed transmission power is 25 mW (14 dBm) for uplink (UL) messages. Downlink (DL) messages, sent from servers to EDs, are permitted up to 500 mW (27 dBm).

Multipath effects, caused by reflections, may result in packets being received multiple times, but CSS modulation mitigates this issue. Communication success depends on factors such as LoS, SF, dataframe length, and duty cycle [74].

LoRa packets propagate through free space, but obstacles weaken signal strength [73]. Fresnel zones, ellipsoid-shaped regions between antennas, must have wave offsets under 180° for optimal reception. Reflections, such as from the ground, result in CNs receiving both

direct and delayed waves. To reduce losses, antennas should be placed outdoors, vertically oriented, and at greater heights. Omnidirectional antennas are ideal for consistent coverage. Signal attenuation, quantified as free space path loss (L_{f_s}), depends on frequency (f in MHz) and distance (D in kilometers) [75, 76], as shown in Equation 2.1.

$$L_{f_s} = 32.45 + 20(log(D) + log(f)), \tag{2.1}$$

The Spreading Factor (SF) is a pre-programmed parameter in a LoRa WSN that defines the number of chirps (data bits) used by an ED to transmit a symbol, ranging from 7 to 12 bits. For example, an SF of 8 allows each symbol to represent 2^8 (256) possible values. Higher SFs improve receiver sensitivity, allowing CNs to detect weaker signals, thus increasing communication range. Each CN channel can decode signals with different SFs, enabling multiple EDs in the same area to transmit without interference.

The SF widens the frequency spectrum, with higher values increasing immunity to interference but reducing bitrate [77]. Although higher SFs improve signal sensitivity and range, they also increase power consumption. Shorter data frames minimize energy use and airtime, reducing collision risk in dense networks. Transmission time depends on the SF, bandwidth, and coding rate. Longer packets require more energy and time, impacting reception success rates.

LoRa employs Cyclic Redundancy Check (CRC) for error detection, adding extra bits to verify message integrity [78]. However, these bits increase packet size and power consumption. The success of LoRa communication depends on environmental factors (e.g., obstacles and interference), antenna gain, SF, and the inherent CSS modulation. In dense networks, CSS can result in frequency channel overlaps [79, 80], especially with LoRaWAN's ALOHA-based protocol. As EDs transmit asynchronously without coordination, collisions and packet loss due to channel overlap increase, which CRC cannot always resolve.

In addition, this LPWAN technology offers a security mechanism based on unique 128-bit key pairs, assigned to each ED when it joins the WSN. The LoRaWAN specification defines two distinct security layers:

- **Network Layer:** Message integrity is ensured using the Message Integrity Code (MIC), calculated with the Network Session Key (NwkSkey). This guarantees the authenticity of messages but does not encrypt the payload.
- **Application Layer:** The payload is encrypted using the Application Session Key (AppSkey), providing end-to-end encryption between the ED and the application server. This ensures that only the application server can decrypt the payload, preserving data privacy from the LoRa Network Server (LNS).

To communicate with the LoRa Network Server (LNS), an ED must first be activated by deploying session keys. There are two activation methods: Over the Air Activation (OTAA) and Activation By Personalization (ABP).

Over The Air Activation (OTAA)

OTAA is widely used for its high security when connecting an ED to the LNS. The activation process begins when the ED sends a join request to the LNS (see Figure 2.6), including its pre-programmed parameters:

- **Device Extended Unique Identifier (DevEUI):** A unique 64-bit identifier for each ED, similar to a MAC address, typically stored in ROM but modifiable in RAM for privacy concerns.
- **Application Identifier (AppEUI):** A 64-bit identifier linking the ED to a specific application server, analogous to a port number. EDs sharing the same AppEUI are associated with the same application.
- **Application Key (AppKey):** A unique 128-bit Advances Encryption Standard (AES) symmetric key stored in both the ED and the LNS, used to generate session keys for secure communication.

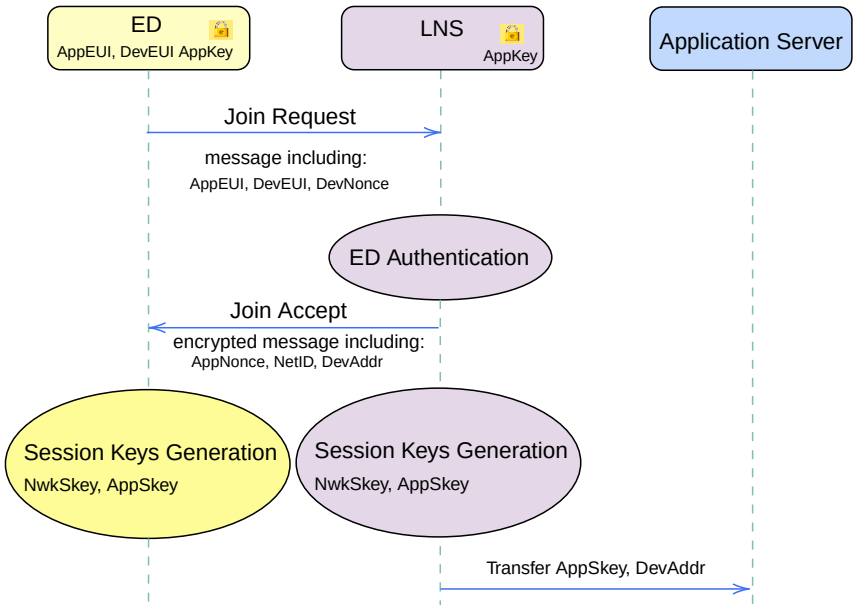


Figure 2.6 Over The Air Activation (OTAA) scheme.

The join request includes a one-time random number (DevNonce) to prevent replay attacks [81, 82]. Any CN receiving the request forwards it to the LNS, which validates the Message Integrity Code (MIC) using the AppKey. If valid, the LNS generates the network and application session keys (NwkSKey and AppSKey) for secure communication. These keys ensure message integrity during transmission and reception, protecting data from unauthorized CNs, and are required to decode packets.

Activation By Personalization (ABP)

In ABP mode, the ED does not need to initiate a join procedure with the LNS. Unlike OTAA, an AppKey is not required, as the ED is pre-configured with all necessary session keys (AppSKey and NwkSKey) and a DevAddr before deployment. These values are typically hardcoded into the device's firmware and remain unchanged unless the device is reprogrammed.

This configuration simplifies the setup process: the LNS only needs the DevAddr and NwkSKey to authenticate and validate messages, while the application server uses the AppSKey (and the known DevAddr) to decrypt the payload. As a result, no join request is needed, and the ED can begin transmitting data immediately.

To protect against replay attacks, frame counters are used to ensure each transmitted message is unique. While static keys can remain valid for extended periods, best practices often include updating keys upon device reset [83]. This combination of static keys and strict frame counter management helps maintain both security and data integrity in ABP mode.

2.3 Architectures for Robotic Resource Sharing

2.3.1 Edge Computing

Mobile robots rely on wireless connections to maintain mobility while receiving external information [84, 85]. However, two key limitations (latency and bandwidth) must be considered, as they are less significant in wired connections [86, 87].

The fifth generation of mobile telephony (5G) represents a transformative standard in wireless communications, designed to connect a wide array of devices, including smartphones, virtual assistants, autonomous vehicles, and industrial robots, within the IoT framework. Key features of 5G networks include low-latency transmissions (targeting delays as low as 1 millisecond) independent of the number of simultaneously connected devices, and significantly higher transfer speeds, reaching up to 20 times the capacity of 4G, with theoretical peaks of

20 Gbps [88, 89]. Additionally, 5G aims to optimize energy consumption, enabling more efficient communication among devices.

5G enhances mobile connectivity, reinforces device-level security, and brings network services closer to end users through advanced edge computing capabilities. These next-generation networks are distinguished by three core attributes [90]: (i) elevated throughput and data handling capacity, (ii) stringent QoS guarantees, and (iii) support for heterogeneous communication environments.

As billions of devices continuously generate increasing volumes of information, efficient traffic management becomes imperative. To address this demand, 5G broadens the available frequency spectrum and facilitates much higher data rates. Using an automotive analogy, in contrast to 4G, 5G provides more lanes (bandwidth) and enables faster-moving vehicles (data transmission).

Designed with interactivity and responsiveness in mind, 5G enables latency-sensitive applications such as autonomous vehicles, remote surgery, and immersive virtual reality. It supports seamless operation across diverse device categories, latency profiles, and network infrastructures—including the IoT—resulting in a versatile and adaptive ecosystem.

In outdoor scenarios, the introduction of 5G significantly boosts the adoption of both cloud and edge computing by enabling the offloading of compute-intensive processes—such as AI inference—from constrained mobile platforms to more capable servers located either in centralized data centers or closer to the access point. This architecture reduces device complexity and power consumption while maintaining low latency.

Building on these advancements, the forthcoming sixth generation (6G) aims to achieve breakthroughs such as sub-microsecond latency, data rates surpassing 1 Tbps, and enhanced energy efficiency. These milestones will be powered by AI-driven network orchestration and sustainable design principles, representing a significant leap toward autonomous, intelligent communication infrastructures [91].

Currently, applications utilizing 5G and edge computing can be classified according to their temporal requirements [90]: (i) hard real-time data, which demand strict adherence to latency thresholds; (ii) soft real-time data, which allow limited delays; and (iii) delay-tolerant or non-critical data. This taxonomy, together with distributed processing strategies, enables the development of scalable and responsive systems optimized for diverse data streams.

Within this framework, edge computing—supported by 5G—is particularly advantageous for processing hard and soft real-time data in scenarios where cloud latency exceeds acceptable bounds. In contrast, cloud computing remains a viable alternative when round-trip communication times fall within the system's temporal tolerance.

This approach focuses on placing computing and storage resources close to the devices that generate the data, an area commonly referred to as the edge. It is closely related to mobile devices and sensors [92]. Its main characteristics include:

- **Low latency:** Processing takes place very close to the data source, effectively reducing network latency and enabling real-time processing.
- **Reduced network bandwidth consumption:** By performing the primary processing locally, it is possible to select which information is forwarded to the rest of the network. Only the most relevant results are transmitted, thereby lowering the overall bandwidth consumption of the main network.

2.3.2 Cloud Robotics

The term *Cloud Robotics* was introduced in 2010 by Dr. James Kuffner [93], an influential figure in robotics and Chief Operating Officer at *Toyota Motor Corporation*, whose work underscores the potential benefits of shared infrastructure and services in this field. Cloud Robotics encompasses a broad spectrum of approaches: from leveraging cloud-based computational resources, known as *Infrastructure as a Service* (IaaS), to accessing repositories of task-specific robotic algorithms, referred to as *Platform as a Service* (PaaS), much like an application store [94]. Moreover, the concept of Cloud Robotics is closely related to other paradigms, including edge computing and fog computing.

Despite its name, Cloud Robotics has significant potential beyond traditional robotics, particularly in autonomous vehicles and transportation. For instance, Google's autonomous vehicle, Waymo, relies on Google's cloud infrastructure [95]. Transportation and mobility are widely recognized as fields where Cloud Robotics can have a transformative impact [96–98]. However, several challenges remain, primarily related to communication issues such as QoS (latency, bandwidth, energy efficiency), stability or potential disruptions, security, and heterogeneity in communication systems [99].

Current applications of Cloud Robotics can be categorized into the following areas [100]:

- **Access to Big Data**: Robotic systems leverage large cloud-hosted databases to perform tasks such as scene reconstruction, object detection, and augmented reality applications. These tasks depend on accessing extensive datasets for proper execution.
- **Cloud computation of parallelizable and sample-based algorithms**: Examples include Monte Carlo analysis, SLAM, RRT-based planning, and image processing, where computationally intensive tasks are offloaded to the cloud.

- **Collective learning**: This is particularly valuable for multi-robot architectures, enabling robots to share experiences and improve performance collaboratively.
- **Human computation**: In this approach, human intelligence is utilized to solve problems such as image labeling and data collection, which are later used to train robotic algorithms. A common example is *CAPTCHA* systems, where users are asked to identify objects in images to contribute to data annotation.

In the short history of Cloud Robotic, several notable projects have emerged. The *DAvinCi* project proposed a software framework for parallelizing the FastSLAM algorithm [101], distributing computational loads across multiple robots. The *RoboEarth* project [102] focused on knowledge representation and storage, enabling robots to share previously captured environmental information to create semantic maps and perform visual SLAM. Meanwhile, the *Ubiquitous Network Robot Platform* project [103] proposed integrating robots and sensors, utilizing the cloud to share data through a communication-centric architecture between various components. In addition, the algorithm presented in [104] improved the efficiency of smart factories based on multi-robot services hosted in the cloud, which can run at different speeds and with other workloads.

Among the various paradigms for distributing computational resources in robotic systems, four main architectures have been identified, each offering distinct advantages and trade-offs:

- **Cloud Computing**: emerged from the idea of providing robots with a *remote brain*. It involves collecting, processing, and storing all IoRT data in the cloud, understood as a remote platform that dramatically extends the computational and storage capabilities available locally. This is achieved through an Internet connection, allowing robots to offload intensive computations to the cloud. This approach enables the design of lighter, more cost-effective, and more intelligent robots, as the primary computational load is shifted to the cloud [105]. Nevertheless, cloud computing is unsuitable for hard real-time services due to latency issues.

  Cloud Robotics exploits the IoT for the convergence of shared infrastructure and services for robotics, ranging from using computational resources in the cloud (IaaS) to having a set of algorithms to deploy on a robot for a given task (PaaS) [106].

  The main drawback of this approach lies in the limited bandwidth of the Internet and the inherent latency associated with transferring data to and from the cloud. These factors can cause delays and interruptions, particularly when the distance between the data source and the cloud is too great, or when network congestion occurs [107]. In such cases, the use of edge computing is recommended, as it addresses these limitations by processing data closer to its source.

- **Fog Computing**: places computational and storage resources on local nodes, such as switches, routers, or gateways, positioned between end devices and the cloud. By processing data closer to where it is generated, Fog Computing significantly reduces latency and enables real-time analysis. This proximity to the data source also helps alleviate network congestion by minimizing the amount of data transmitted to centralized cloud servers. However, the trade-off lies in its limited storage and computational capacity compared to cloud-based solutions, making it suitable primarily for localized or time-sensitive applications.

  It is important to distinguish Fog Computing from **Edge Computing**, as the latter focuses on processing data directly on or near the end device itself, such as a sensor or robot. While Fog Computing acts as an intermediary layer between the cloud and multiple devices, Edge Computing operates at the extreme edge of the network, providing faster responses by eliminating additional network hops. This distinction highlights their complementary roles, with Fog Computing managing broader network data flows and Edge Computing addressing immediate, localized processing needs.

  Ken Goldberg and his team at UC Berkeley's AUTOLab have significantly contributed to the integration of cloud computing in robotics through the development of *FogROS* and its successor *FogROS2*. These frameworks enable robots to offload computationally intensive tasks to the cloud, enhancing performance without requiring substantial modifications to existing codebases.

  FogROS, initially developed for ROS 1, allowed researchers to specify which components of their software should be deployed to the cloud and to select suitable hardware configurations for execution. This enabled access to powerful computing resources such as GPUs and multi-core CPUs, resulting in significant performance improvements in tasks like motion and grasp planning. For example, FogROS achieved computation speeds up to 34.2 times faster, with minimal latency overhead [108].

  With the transition from ROS 1 to ROS 2 following the deprecation of the former, the team developed *FogROS2*, a complete redesign that leverages ROS 2's enhanced middleware, real-time capabilities, and modular architecture. FogROS2 introduced several improvements over its predecessor, including reduced communication overhead, faster startup times, and more reliable execution. Empirical evaluations demonstrated that FogROS2 reduced SLAM latency by 50%, decreased grasp planning time from 14 s to 1.2 s, and improved motion planning performance by a factor of 45 [109].

- **Multi-access Edge Computing (MEC)**: formerly known as Mobile Edge Computing, MEC is a network architecture that deploys computational and storage resources on

base stations or edge servers, positioned close to end devices. By processing data near the edge of the network, it reduces latency, optimizes bandwidth, and supports real-time applications such as autonomous vehicles, augmented reality, and real-time analytics. Initially focused on mobile networks, MEC has evolved to support multiple access types, including WiFi and fixed networks. It plays a key role in 5G by enabling localized services, ultra-low latency communication, and seamless integration with IoT and network slicing. MEC is also considered a key enabler for the realization of IoT systems, as it facilitates low-latency processing, scalable service provisioning, and localized decision-making—essential for supporting time-critical and bandwidth-intensive applications [110]. A master node is often used to coordinate resource allocation, manage network topology, and ensure responsiveness for time-sensitive tasks.

- **Hybrid Platforms**: combine cloud and edge computing by offloading time-critical operations to edge nodes, while delegating non-real-time processing and global decision-making to the cloud. Although powerful, hybrid systems are more complex to design and manage.

Current efforts focus on extending cloud robotics to dynamic environments by promoting the use of MEC centers for real-time applications [111]. Moreover, recent architectures have adapted concepts from distributed computing to the robotics domain [112]. For instance, Botta et al. [113] proposed an edge-oriented architecture in which computational resources are deployed in close proximity to vehicles, aiming to reduce latency and improve service continuity. Meanwhile, Yousefpour et al. [114] introduced Fog Robotics, a paradigm that positions processing capabilities near the communication infrastructure to enable low-latency decision-making and localized autonomy. In both approaches, the primary goal is to align computational and communication resources to support responsive and context-aware robotic behavior.

Networked robot cooperation in field applications such as agriculture [115] and disaster robotics [99] can benefit especially from advances in Cloud Robotics 2.2. In the case of SAR missions, H-WSNs can be highly valuable [116–118]. Furthermore, nodes equipped with complex sensors—such as LiDAR—can locally perform part of the data processing on the robot itself, following the dew computing paradigm, which emphasizes autonomous, offline-capable computation with optional cloud synchronization.

Incorporating cloud computing into Mobile Ad hoc NETworks (MANETs)—which operate without fixed infrastructure—can help overcome limitations in storage, processing, and energy resources [118]. To improve network coverage in such dynamic environments,

Table 2.2 Comparison of computing paradigms in distributed robotic systems

Paradigm	Location of Computation	Main Characteristics	Typical Examples
Cloud Computing	Remote data centers (centralized infrastructure)	High processing power, global scalability, high latency, dependent on stable connectivity	AWS, Google Cloud, Azure; robot path planning in the cloud
Edge Computing	Near the data source (edge servers, base stations)	Low latency, real-time response, intermediate processing, offloads cloud traffic	MEC in 5G, edge gateways in factories or hospitals
Dew Computing	On the end device itself (robots, sensors, smartphones)	Offline-capable, lightweight processing, syncs with cloud when needed	Local sensor data pre-processing on UAVs or IoT nodes

adaptive routing protocols have been proposed [119, 120], along with ad hoc strategies for deploying mobile nodes in H-WSN, such as mobile sinks [56, 65, 121–124].

In the context of hyper-connectivity, edge devices have diverse connectivity requirements that challenge Internet Service Providers (ISPs). Cloud-based features of 5G—such as network slicing and MEC—help address issues like latency, QoS, energy efficiency, and mobility [125, 126]. 5G enables flexible deployment of multiple virtualized end-to-end networks (slices), tailored dynamically to user requirements. Orchestration is key: ISPs use Network Slice Templates (NESTs) to define service attributes (e.g., latency, reliability, throughput), translate them into application-level parameters, and manage slice-specific resources (e.g., vehicles, robots, control centers). MEC contributes to real-time responsiveness while reducing cloud traffic [127].

Despite its potential, H-WSN deployment faces operational challenges requiring robust architectures, implementation strategies, and real-world testing. Integration of sensor networks with robots has been explored using platforms like ROS and FIWARE [128, 129]. For instance, LoRaWAN has been tested with UAV-based gateways and simulated satellite links [130], while cloud robotics testbeds have supported autonomous navigation in remote areas [131], and remote control of heterogeneous uncrewed vehicles [132].

The International Telecommunication Union (ITU) classifies cloud service requirements into: (i) Ultra-Reliable Low-Latency Communications (URLLC) for mission-critical tasks like remote surgery; (ii) enhanced Mobile Broadband (eMBB) for data-intensive applications such as VR/AR and streaming; and (iii) massive Machine Type Communications (mMTC) for large-scale, low-data, low-latency sensing systems [133].

2.3.3 Computing in Disaster Robotics

Emergency robotics focuses on deploying robots to assist in disaster response and SAR operations, reducing risks to human responders. Robots can access hazardous or confined areas to search for victims and assess damage, performing tasks that people or dogs cannot. They have been deployed in various disaster scenarios, from earthquakes to industrial accidents. For example, after the 2011 *Fukushima Daiichi* nuclear plant accident in Japan, mobile robots were sent into radiation-contaminated reactor buildings that were far too dangerous for humans [134, 135]. These robots conducted surveillance and inspection missions—but only after engineers solved critical issues such as hardware reliability, radiation shielding, and restoring communication for control and data transmission. Modern rescue robots come in many forms—UGVs, UAVs, humanoids, quadrupeds, and even marine robots—selected for their ability to reach otherwise inaccessible areas, such as flying over debris or operating underwater [136]. They are often equipped with cameras, sensors (e.g., thermal imagers, gas detectors), and tools to navigate collapsed structures or hazardous sites while streaming situational information to human commanders.

Effective disaster-response robotics rely on a hybrid computational architecture that strategically partitions workloads between edge-level CPUs/GPUs and centralized cloud servers. The following scenarios illustrate the complementary roles of local and remote computing in enabling real-time perception, planning, and coordination under critical conditions.

Life-Critical Perception and Control on the Edge

- **Real-time victim detection in smoke-filled buildings.** An aerial robot equipped with a thermal imaging sensor must process heat signatures in real time while navigating complex and cluttered environments, such as partially collapsed structures. To ensure timely detection of potential human presence, a low-latency edge computing unit with integrated GPU capabilities executes lightweight deep learning models (e.g., object detection networks) locally. Avoiding delays caused by external communication is critical, as transient thermal anomalies may disappear within fractions of a second. Meanwhile, the onboard CPU handles high-frequency control loops and integrates data from multiple sensors—such as IMU, LiDAR, and barometric altimeters—to maintain stable flight and situational awareness. Such configurations have been applied in several studies addressing real-time thermal-based victim detection and onboard inference in SAR robotics [137, 138].

- **Radiation hotspot mapping after a nuclear accident.** A ground-based robot operating in a contaminated environment must generate a 3D radiation intensity map while

simultaneously navigating through unstable terrain and avoiding obstacles. Onboard GPU resources are employed to accelerate real-time SLAM, while the CPU processes sensor readings from radiation detectors using probabilistic filters to estimate safe paths. Due to strong signal attenuation in such environments, offloading tasks to the cloud is typically infeasible, making local computation essential for autonomous decision-making and operator safety. A representative application case is presented by Schwaiger et al. [139], demonstrating the effectiveness of onboard SLAM and radiation filtering in high-contamination zones.

- **Dynamic swarm relays in underground tunnels.** In confined, GPS-denied environments such as subterranean tunnels or collapsed infrastructure, multiple small ground robots operate collaboratively to explore and maintain communication links. Each robot autonomously computes its exploration strategy using onboard processing and contributes to a dynamic multi-hop network. CPU resources handle local communication protocols and motion planning, while lightweight GPU acceleration supports tasks such as low-light image enhancement or terrain classification. This decentralized approach ensures operational resilience even under degraded or intermittent communication conditions. This concept has been successfully implemented in the ACHORD framework, where robots autonomously adapt their behavior based on local connectivity and deploy communication relays as needed to maintain a robust multi-hop network under severe communication constraints [140].

- **Offloaded trajectory control for resource-limited drones.** Aerial robots with minimal onboard computing capabilities may struggle to execute high-frequency model predictive control (MPC) loops. A Kubernetes-based edge architecture allows trajectory planning and control to be executed on nearby ground servers, enabling real-time adaptation to dynamic environments without overloading the drone's CPU [141].

Heavy Analytics and Global Coordination in the Cloud

- **Large-scale 3-D reconstruction for incident command.** When sufficient bandwidth is available—e.g., through a high-speed wireless backhaul or satellite uplink—field robots transmit selected image frames and LiDAR scans to cloud servers for advanced processing. High-performance GPUs in the cloud execute large-scale bundle adjustment and semantic segmentation pipelines, generating a detailed 3D representation of the disaster environment. This globally consistent map supports strategic planning and situational awareness for emergency responders, while offloading teraflop-intensive computations that exceed the robot's on-board capabilities. Similar 3D data fusion

challenges have been addressed in forest mapping, where UAV and backpack LiDAR point clouds are coregistered without markers using computationally intensive algorithms [142], demonstrating techniques that could support cloud-based integration of multi-platform scans in disaster scenarios.

- **Adaptive path replanning under evolving hazards.** In dynamic environments where terrain conditions change rapidly—such as aftershocks in earthquake zones or the spread of fire or gas leaks—updated environmental data is streamed to the cloud for centralized analysis. Cloud-based HPC clusters run probabilistic path-planning algorithms (e.g., sampling-based motion planners or optimization-based solvers) under constraints such as structural safety or air quality. Only the optimal trajectory is selected and transmitted back to the robot, enabling fast adaptation without overloading onboard resources. This architecture has been validated in multi-robot systems using a cloud-based framework that integrates D* Lite planning with fuzzy inference and online learning [143], enabling adaptive behavior in evolving environments while preserving onboard resources.

- **Federated learning of gas-sensor drift.** Robots deployed in different environmental conditions may experience variations in sensor accuracy over time. Each unit updates a local machine learning model to compensate for sensor drift, based on its own measurements and conditions. Periodically, these local models synchronize with a centralized cloud server, where GPU resources aggregate and retrain a global model. This federated learning approach preserves bandwidth and privacy while ensuring consistent performance across all deployed units in multi-day operations. Recent work by Chen et al. [144] introduces FedConD, a federated learning framework that detects and adapts to concept drift in asynchronous settings. Their method shows robust performance under evolving local data distributions—conditions analogous to gas-sensor drift in long-term robotic deployments.

- **Cross-team coordination of heterogeneous fleets.** Coordinating multiple robotic units—such as aerial scouts, ground transporters, and communication relays—requires solving complex multi-agent task allocation problems. In the cloud, CPU-based solvers handle constraint optimization models (e.g., MILP), while GPU-accelerated inference engines predict task success probabilities or resource utilization. This global coordination service assigns tasks dynamically based on robot capabilities, task urgency, and environmental factors, reducing onboard computation while improving operational efficiency and fleet resilience. A real-world demonstration of this paradigm is found in the work by He et al. [145], where a cloud-based framework enables real-

time coordination and collision avoidance for large-scale robot swarms. The system distributes computation-intensive tasks to cloud servers, allowing simple robots to act intelligently in dense environments through scalable control topologies and shared state updates.

Adaptive Computation and Sensing in Intermittent Networks

Disaster-response robots must adapt their computational load to unpredictable and often degraded network conditions. When connectivity drops—e.g., during tunnel entry, dense rubble traversal, or structural collapse—robots dynamically migrate from cloud-supported processes (such as global mapping or multi-agent coordination) to onboard alternatives like visual-inertial odometry or local SLAM. Upon reconnecting, buffered data (e.g., map fragments, telemetry logs, or sensory updates) is uploaded to the cloud for integration [146]. These dynamic transitions are facilitated by robotic middleware such as ROS 2, whose data distribution layer (Fast DDS) supports configurable QoS policies that prioritize critical data streams and tolerate latency or packet loss. This infrastructure enables reliable task migration and data synchronization even in lossy or intermittently connected environments [147].

This adaptive capability relies on orchestration between CPU and GPU resources: CPUs monitor network quality and reallocate processing tasks, while GPUs dynamically adjust inference pipelines based on available resources and mission phase. Frameworks such as FogROS2 [148] and its fault-tolerant extension FogROS2-FT [149] enable this type of real-time workload migration, allowing robots to operate seamlessly across edge-cloud boundaries.

In summary, CPUs deliver the low-latency, deterministic processing required for control, planning, and sensor fusion, while GPUs provide the parallel throughput essential for perception, learning, and large-scale analytics. By leveraging both processor types—at the edge and in the cloud—disaster robotics systems achieve a balance between autonomy, responsiveness, and intelligence. Designing architectures that support fluid task partitioning across these layers is essential for robust, context-aware decision-making in high-risk and dynamic environments.

Wireless sensor nodes complement robotic systems in SAR scenarios by providing persistent environmental sensing through static and mobile nodes [150, 151]. When integrated with middleware like ROS, WSNs enhance data availability and situational awareness in regions with limited robot access or intermittent communication. Mobile robots can also serve as data MULEs or dynamic nodes within a H-WSN, supporting coordinated sensing and enabling distributed decision-making under constrained network conditions.

Chapter 3

Perception and Communication via Wireless Networks in Complex Scenarios

3.1 Introduction

How do we obtain information from Remote, Outdoor, Unstructured, and Disaster (ROUD) scenarios where people are believed to be lost or trapped, and where rapid rescue is required? ROUD scenarios include risks for victims and rescuers who must remain for an unpredictable time in the catastrophic area. Measurement of the current state of the operational environment is critical during the Search and Rescue (SAR) stages, especially without pre-existing infrastructure. Thus, extracting information from the site so that emergency coordinators can act as efficiently as possible is the challenge.

Perception in SAR missions must rely on low-power End-Devices (EDs) due to the extended duration of these operations, which can span days or weeks, making recharging deployed EDs impractical. A WSN becomes hybrid when some of its nodes acquire mobility to access areas with low coverage, giving rise to a two-tier node hierarchy: (1) EDs, organized into Sensory Groups (SG), and (2) non-sensor nodes, which can act as relay nodes, gateways, and Concentrator Nodes (CN), depending on the network topology (e.g., mesh or star).

This chapter addresses the first contribution of this thesis: perception and communication in ROUD scenarios through the co-design, development, and deployment of Hybrid Wireless Sensor Networks (H-WSN) and Wireless Local Area Networks (WLAN) utilizing mobile units to ensure reliable information extraction in both online and offline operational modes.

In a SAR scenario, mobile units consist of SAR agents such as mobile robots, SAR dogs, or humans. The challenge lies in coordinating this type of SAR team to move and deploy both Sensory Groups (SGs) and non-sensor nodes. EDs can collect data from areas of interest

where accessibility poses significant risks or inefficiencies for first responders. Additionally, Uncrewed Ground and Aerial Vehicles (UGVs and UAVs) can operate either tele-operated or autonomously, reducing human exposure to danger by transporting SGs and non-sensor nodes to establish portable sensing and communication infrastructures. By delegating SG mobility and coverage tasks to UGVs and UAVs, first responders can focus on the critical mission of searching for, locating, and rescuing victims.

Regarding WSNs, Personal Area Networks (PAN) and Low Power Wide Area Networks (LPWAN) can assist a SAR team by acquiring information from points near or far, respectively, from the area of interest, requiring low-power, and being persistent.

Concerning communications in ROUD scenarios, mobile WLANs allow agents to be covered outdoors and indoors and transmit higher throughput than WSNs. The potential of WiFi mesh networks to extend coverage by incorporating mobile nodes is a subject worth exploring.

The rest of the chapter is organized as follows: Section 3.2 introduces the design and implementation of PAN networks in realistic ROUD scenarios, evaluating the results of the proposed Zigbee and BLE systems. Section 3.3 explains the importance of using mesh networks extended by mobile nodes to provide 5G coverage in underground environments. Section 3.4 presents the integration of an LPWAN-based H-WSN in SAR. Finally, Section 3.5 discusses the contributions and shows the conclusions.

3.2 Integration of PAN in ROUD scenarios

Although PAN technologies are designed for an individual person's workspace, they can be highly useful in ROUD scenarios when configured appropriately. Their low power consumption, short-range connectivity, and ability to form ad hoc networks can support different tasks. The challenge is to integrate PANs in SAR agents [152] to extend their individual agent's workspace.

3.2.1 Zigbee

3.2.1.1 Design and Implementation

The proposed Zigbee H-WSN does not include router nodes (ZR), focusing instead on direct communication between mobile nodes (each carrying a ZC) and sensor nodes (ZEDs) in ROUD scenarios. Consequently, this H-WSN is composed of a set of star networks, each governed by a ZC and connected to a common server (see Figure 3.1). By avoiding tree or mesh topologies, the complexity of network configuration is eliminated, which is particularly

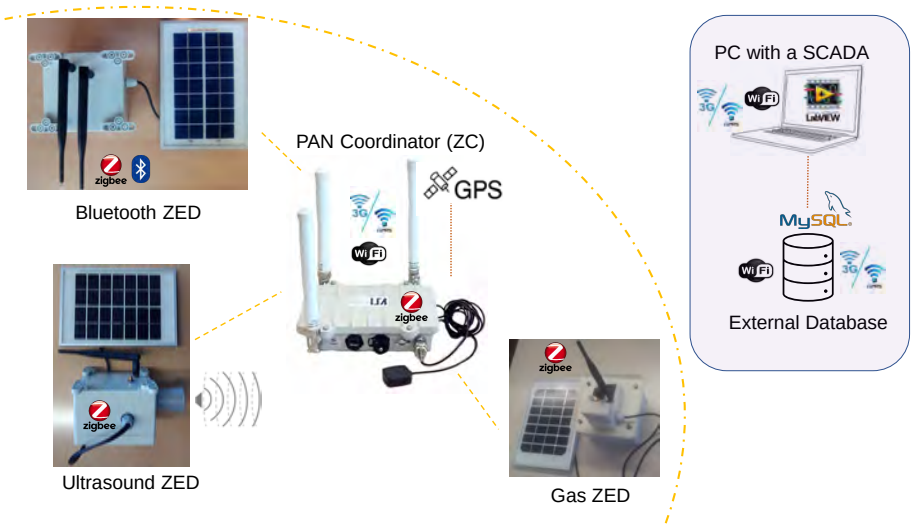


Figure 3.1 Star Zigbee-based WSN

crucial in ROUD scenarios where ZEDs lack Internet access, making remote reconfiguration for role changes impractical. Furthermore, tree and mesh networks increase communication latencies and energy consumption due to multi-routing. This approach aims to test coverage in ROUD scenarios without deploying relay nodes, evaluating a limiting case.

All ZEDs are built upon the same hardware, the Waspmote v1.2, with an additional communication module, the XBee Pro Series 2 from Digi International. The ZEDs are enclosed in IP67-certified cases, allowing for outdoor deployment under adverse weather conditions.

The designed ZEDs are equipped with specialized sensors tailored for specific purposes: environmental sensors for measuring gases, humidity, atmospheric pressure, and temperature; Bluetooth detectors capable of identifying devices within a range of up to 100 meters; and ultrasonic sensors to detect objects or individuals entering their detection radius. Thus, when a ZED detects some information, it storages or sends it to the ZC in charge of its PAN.

The ZC nodes are based on a multi-protocol router called Meshlium 3.5, including Zigbee/802.15.4, WiFi, Bluetooth, and 3G/GPRS protocols, from Libelium [153].

An open-source framework from Digi, called XCTU [154], allows to configure the firmware of the ZEDs, programmed in C, while the ZCs are configured through their internal web server services running on a Linux-based system. Leveraging open-source platforms enables scalability and expandability of the proposed H-WSN.

In operation, the ZEDs collect environmental data and transmit it via Zigbee to the ZC of their associated PAN. The ZC stores this data in its internal memory as table entries in a

local database. The stored data is then forwarded from the ZCs to an external database over WiFi or 3G, making the ZCs intermediaries between the field and the control center.

To visualize and manage this information in real time, a Supervisory Control and Data Acquisition (SCADA) system has been developed. The data is handled using a MySQL database that communicates continuously with a LabVIEW Virtual Instrument (VI). This setup allows the SCADA system to query the database in the background, eliminating the need for users to manually input SQL commands. The SCADA interface provides the SAR team with real-time insights into the area of interest, including graphical representations, overlaid map layers of the operational area, and the status of deployed ZEDs.

An implementation of the proposed H-WSN consists of two stars, each coordinated by one ZC, connected to the SCADA system.

PAN-A is managed by a static ZC, serving eight static ZEDs and one mobile. The mobile ZED was carried by a SAR dog, trained to aid in victim localization in SAR scenarios. This mobile ZED is capable of measuring gases and environmental values, as well as transmitting the dog's geolocation.

PAN-B coordinator is embarked on a UGV, capable of storing information in its local DB from an isolated ZED, which is intentionally located far from the static PAN-A coverage. This ZED is a single Bluetooth detector ZED, designed to identify nearby Bluetooth devices that could be worn by a potential victim (PV). The ZED stores data and transmit it to the mobile ZC via Zigbee when the mobile robot is in its range.

Figure 3.2 shows the modularity and connectivity of this H-WSN, where each ZC creates its own PAN. The ZEDs capture information from the surrounding environment and transmit data to the ZCs via Zigbee if there is a good LoS, the traffic being bidirectional.

Both ZCs are connected to the Cloud via 3G/GPRS, enabling remote access to data stored in a private external MySQL database, linked to a LabVIEW-based information system.

The PAN-A ZC was positioned at the FCC, two meters above ground level, offering a partial LoS over the operational area. In addition, the XBee radio link has a limited range (1200 m in ideal environmental conditions and LoS), and the 2.4 GHz band is highly congested in SAR environments, as it is used by numerous WiFi and Bluetooth devices operated by the rescuers themselves or by the deployed robots.

Thus, to overcome this limitation, the Rambler robot was equipped with the PAN-B ZC, designed to detect ZEDs that were out of range or lacked LoS with the FCC.

Preliminary tests with ZEDs configured for PAN-B helped identify optimal deployment locations outside the static PAN-A's coverage, for validating the system's ability to recover data reliably in challenging environments.

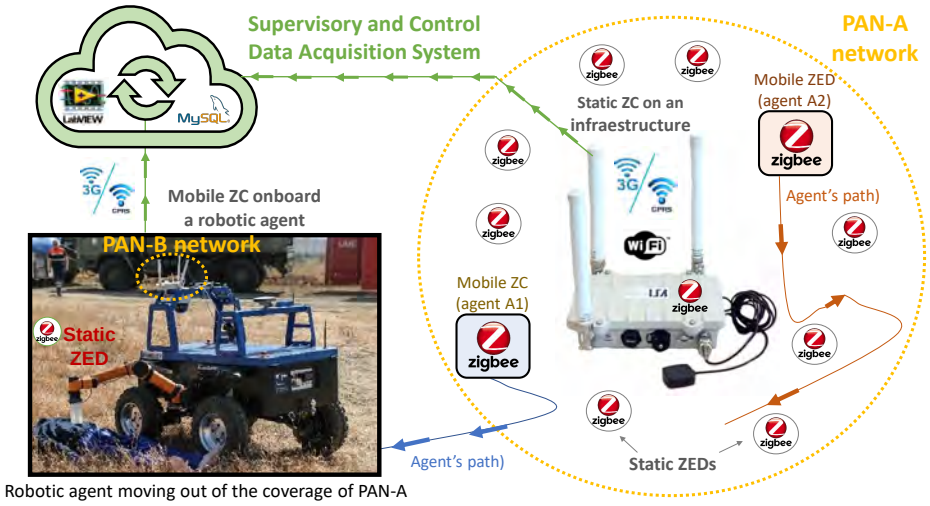


Figure 3.2 System architecture of the Zigbee H-WSN with one static ZC and a mobile one

The robot began its movement outside the range of any ZED registered in PAN-B. At the PoI, located beyond PAN-A's coverage area, the Bluetooth ZED was deployed for PAN-B, operating as a potential victim detector by identifying nearby wearable Bluetooth devices.

The rugged and uneven terrain, marked by numerous hillocks, often disrupts the LoS between the robot and the out-of-coverage ZED, resulting in intermittent communication outages. To mitigate data loss in the Zigbee H-WSN, a solution was implemented where ZEDs temporarily store data in their internal memory (a 2 GB SD card) during communication outages. Once the mobile ZC (dynamic PAN-B coverage) re-establishes connectivity, the robot retrieves the stored data while simultaneously collecting new information as it approaches, ensuring seamless data acquisition.

The Bluetooth ZED was placed in an area where a victim was potentially expected to be found, based on the type of terrain. The exact position of the lost ZED was known in advance (recorded before the experiments using an RTK GPS). Additionally, during the exercise, Rambler's position was monitored in real time via its RTK GPS. Therefore, it is possible to determine the distance at which the PAN-B coordinator (ZC) begins to provide coverage to the lost ZED. After the detection of a victim near the Bluetooth ZED, this node was used to gather information about the identity of the rescuers.

Although both ZC nodes create and manage their own PAN, their local DBs are synchronized using WiFi/3G connectivity with an external DB, which allows monitoring of all the information at the FCC through the SCADA developed. In this way, FCC personnel can monitor the status of the deployed ZEDs in real time and face the rescue of victims, providing technical support to the SAR team at a glance.

Figure 3.3 ZED scanning rescuers' identities and PVs

The lost ZED sent its gathered data, including the RSSI and MAC of the scanned Bluetooth devices worn by the rescuers, to the ZC on the mobile platform (see Figure 3.3), which in turn makes this information available to the FCC (through the Internet).

The PAN-A ZEDs were deployed around the experimental area, relatively close to FCC to guarantee coverage. In addition to these static ZEDs, a SAR dog was carrying a ZED (a gas module plus a GPS module), also registered in the PAN-A ZC. Both modules were installed on a harness especially developed for the application and it does not affect the dog's behavior. It includes two compartments, one at each side for each module, whose weight has been balanced to be comfortable for the SAR dog. Then, the dog is sent by its trainer to explore the area, in a series of short routes, defined according to a regular exploration pattern, starting from FCC. This mobile ZED is capable of measuring and transmitting (via Zigbee) the position of the dog and the following measurements: temperature (°C), relative humidity (%), absolute pressure (kPa), and the concentration (p.p.m.) of some gases (oxygen, carbon monoxide, nitrates, etc.).

Figure 3.4 shows all the sensor-nodes (ZEDs) deployed for both PANs. The straight-line distance between the PAN-A ZC and the PAN-B ZED was 274 meters. Rambler started its movement from the FCC and took 10 minutes to make the first reception, at which point it was 110 meters away. Along its path, it had to navigate several hills, losing the LoS.

In addition to the static ZEDs, Figure 3.4 illustrates the movement of the SAR dog, tracked via GPS, carrying a ZED from PAN-A. As the search for victims begins at the FCC, the mobile ZED transmits the data it collects. However, as the SAR dog ventures

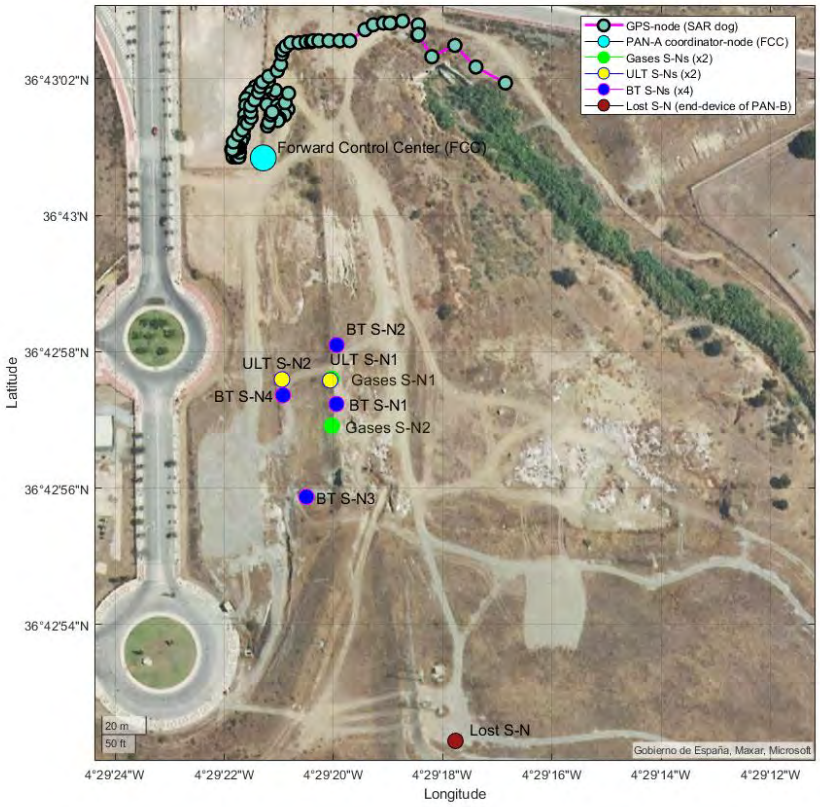


Figure 3.4 Zigbee sensor nodes deployed in the operational environment

deeper into the tunnel zone, the elevation difference increases, eventually leading to a loss of communication with the ZED it is carrying. Therefore, the data collected by the dog in remote or hidden areas out of reach of the PAN-A ZC will be retrieved once the dog returns to the coverage area, allowing the data to be synchronized at that point.

3.2.1.2 Experimental results

Figure 3.5 demonstrates that not only the range but also the LoS is important. For the representation of the direct LoS, it has been considered that the receiving antenna at the FCC is 2 meters above the ground on which it is located. Green LoS indicates good connectivity while red LoS means no coverage for the ED.

As soon as the SAR dog moves away from the coverage area of PAN-A, Zigbee packets from its ED are no longer received. It is observed that the last Zigbee packet detected for the SAR dog from the FCC was transmitted from 114.8 meters. It was impossible to establish

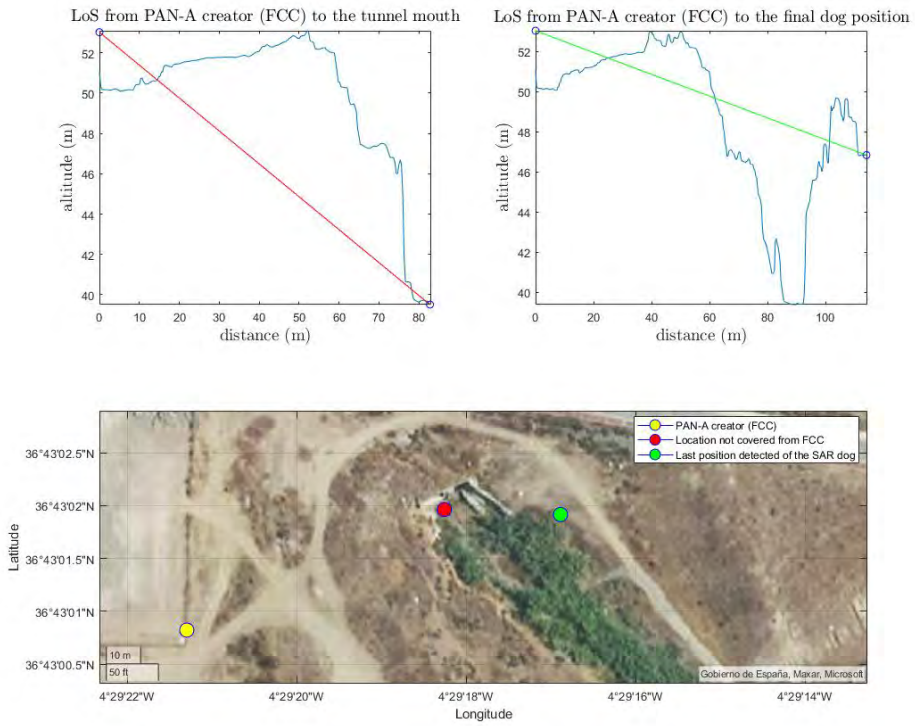


Figure 3.5 Importance of LoS and range between the ZED and the ZC (red means without coverage; green indicates good connectivity)

connectivity from positions that were closer but more hidden in the terrain. For this reason, when the SAR dog entered the area of the stream, looking for victims in front of the tunnel entrance, it was not possible to obtain information from the canine agent.

In total, 6862 Zigbee data frames were registered into the local DB of the PAN-A ZC located in the turret next to the FCC. Packets were captured over a period of 4 hours, 39 minutes, with the next distribution:

- A total of 86 packets were received from the SAR dog, which was in motion for 4 minutes and 34 seconds (see Figure 3.4). These packets contained georeferenced data.
- A total of 149 packets containing gas data were received from the ZED measuring gas in SN1, and 89 packets from the ZED measuring gas in SN2. These were the only two ZEDs among all deployed devices dedicated to gas measurement in PAN-A.
- A total of 200 packets with ultrasound data were received from the static ZED in SN1, and 1153 packets from the static ZED in SN2, both located in PAN-A.

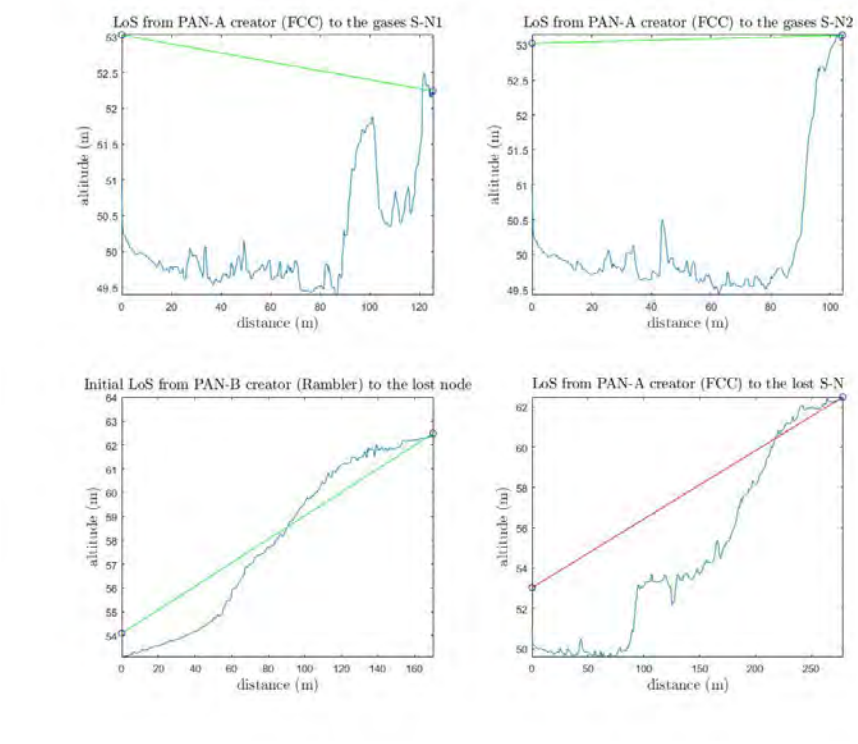


Figure 3.6 Ground profiles and LoS from both PAN ZC to different sensor nodes (ZEDs)

- A total of 851, 1422, 877, and 2035 packets with Bluetooth detections were received from the four Bluetooth sensor nodes, respectively.

Additionally, 113 Zigbee packets were captured by Rambler, the PAN-B ZC, over a period of 25 minutes, all originating from the same static position (corresponding to the lost sensor-node). In addition, during the Rambler's path towards the lost ZED, 41 of the 113 data frames were not received at the time of transmission but were recovered when the UGV regained coverage of the lost ZED and was able to transmit the data stored in its internal memory.

Figure 3.6 shows the ground profiles and LoS between antennas, the green LoS meaning that the connection and transmission were successful.

3.2.1.3 Evaluation

In this study, a Zigbee H-WSN was deployed and evaluated in a realistic SAR scenario, consisting of two independent PAN networks coordinated by two ZCs. PAN-A included a static coordinator, eight static ZEDs, a mobile robotic agent (Rambler) acting as a mobile sink, and a rescue dog carrying a ZED. PAN-B was designed to service an isolated ZED,

with Rambler providing the coverage. Although the XBee modules used claim a maximum range of 1200 m, Zigbee range limitations influenced node placement, with PAN-A nodes positioned at a maximum distance of 154.86 m and the SAR dog's ZED maintaining effective transmission up to 114.8 m under heavy interference. When the SAR dog returned to PAN-A coverage, it successfully transmitted the gathered packets remotely.

During the 6.5-hour exercise, nodes remained operational without significantly depleting their battery life, though extended operation over several days is necessary to draw definitive conclusions about power consumption.

The results of this study align with Fresnel zone theory [155], confirming that radio-communication is influenced not only by obstacles but especially by uneven terrain, which disrupts the line-of-sight (LoS) and reduces effective range and reliability. These conditions justified the use of Rambler as a mobile ZC to recover packets lost due to terrain unevenness. Rambler's GPS position, enhanced by RTK corrections, allowed precise measurement of distances between the robot and the static ZEDs, enabling dynamic coverage zones to be established as the robot detected Zigbee packets. This dynamic adaptability was essential to maintain reliable communication in such complex environments.

The importance of Rambler's mobility was underscored by its ability to replace human operators in extending coverage. Data synchronization with a central database at the FCC via 3G or WiFi provided real-time monitoring. The experiment demonstrates the effectiveness of adaptable network configurations in overcoming communication challenges in SAR scenarios, emphasizing the potential of mobile robots to enhance operational efficiency in hostile and complex environments.

The experiments have validated the effectiveness of deploying star-type Zigbee H-WSNs in complex scenarios where terrain variability, environmental interference, and the mobility of vehicles and people in the area significantly impact the signals. The proposed system uses an offline mode to collect information from hard-to-access areas or those with limited radio coverage, such as tunnels. The use of mobile ZEDs and ZCs enables the collection of data from remote locations. The star configuration, without using ZRs, has allowed the real coverage of Zigbee to be tested through XBee Series 2 nodes. Currently, more modern modules, such as XBee SC3, are available, offering extended ranges, making extending the proposed architecture with these modules worthwhile. Furthermore, once the terrain has been evaluated, it becomes easier to deploy ZR nodes, allowing the network topology to migrate to a mesh or tree configuration, resulting in greater efficiency. However, the high energy consumption of ZRs, which must always remain active, can be a drawback when deploying these nodes in ROUD scenarios. Nevertheless, testing this configuration with more modern modules powered by solar panels would be interesting in evaluating their operability

over several days. Additionally, the use of mobile robots, including UAVs, to act not only as mobile ZEDs and ZCs but also as mobile ZRs would enable coverage to be extended on demand from the FCC. Lastly, it would be worthwhile to investigate how to leverage Zigbee captures made by different ZCs (from separate PANs) to promote intelligent autonomous movements by exploratory robots. To this end, synchronizing all local databases (associated with each ZC), as performed in this work, has allowed unifying all the information from the independent networks into a SCADA system hosted at the control station, facilitating decision-making. Therefore, the scalability of the proposed system presents several potential lines for future development.

3.2.2 Bluetooth Low Energy as a sensor

Bluetooth is far more standardized than Zigbee due to its widespread use in a variety of devices from different brands and types, including wearable devices for first responders, such as biometric sensors. Bluetooth Low Energy (BLE) emerged as a low-power specification allowing ranges of up to 100 meters, making it possible to deploy sensor devices that transmit environmental data via BLE to mobile agents. Analyzing its viability in ROUD scenarios is necessary, for which a BLE-based sensory system is proposed. Specifically, a novel BLE-based system for close victim detection in ROUD scenarios is presented. The system utilizes different robotic platforms equipped with BLE scanners to detect BLE EDs, which could potentially be worn by victims.

3.2.2.1 Design and implementation

A group of BLE nodes, based on the ESP32 microcontroller, has been configured as BLE transmitters with a transmission power of 9 dBm to emulate victims in the context of SAR missions, and deployed in the terrain to be explored. UGVs and UAVs embark also an ESP32 microcontroller, but operating as a BLE scanner. The ESP32-based BLE scanner was selected as the winner following a comparative evaluation with a commercial BLE scanner (Meshlium) [156], by mounting them on mobile platforms (e.g. UAVs and UGVs) and comparing their operation.

The proposed BLE-based network allows to emulate victims in an outdoor area, whether they are on the surface or semi-buried. The ESP32 boards act as transmitter beacons, capable of keeping their MAC fixed, which is not possible with commercial smartphones for privacy reasons. The detection elements are different BLE scanners onboard robotic agents capable of detecting the presence of PVs in their coverage area, as their MAC does not change.

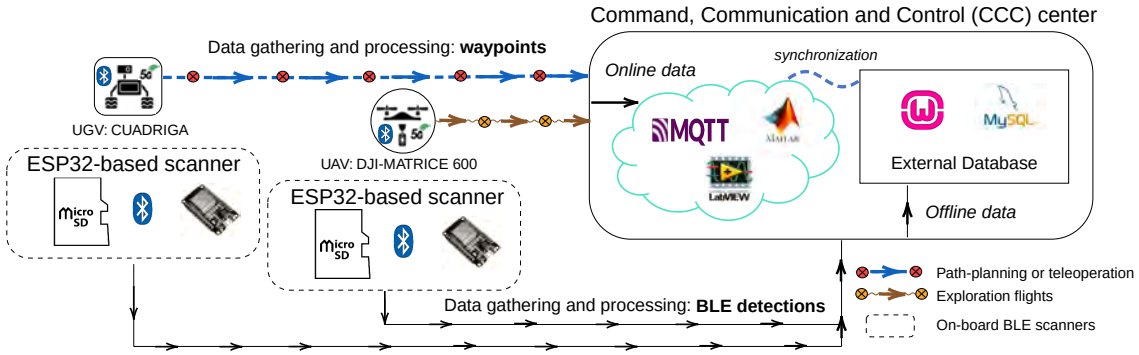


Figure 3.7 ESP32-based BLE detection system

The UGVs navigate to an area of interest, while the UAVs perform exploratory flights sweeping the same area of interest at low altitudes and low speeds. When a detection is made from any robotic agent, its RTK GNSS position is transmitted to the control and command center (CCC). Each pair, detection (RSSI) and agent's position (GNSS) is transmitted through a Message Queue Telemetry Transport (MQTT) broker linked to the Cloud. Each mobile robot has an MQTT client which publishes the information to the CCC on a topic. Therefore, all mobile robots require Internet connectivity to access the same MQTT broker, enabling them to send information that becomes accessible at the CCC. The bidirectional data flow through the broker enables the remote operator at the CCC to activate agents and guide them toward georeferenced points of interest.

The system (Figure 3.7) consists of integrating a UGV and a UAV with the CCC, so that a remote operator can know where they are in real time, and can match the detections that each makes as they move across the terrain. To do this, the CCC hosts the MQTT server to which both robots are connected, through an MQTT client integrated in their respective systems, running a common LabVIEW-based application. The existing interface in the CCC is developed in MATLAB, and allows the visualization of the path of the vehicles as well as the detections with their RSSI. All BLE detections and positional data are stored in an external MySQL database.

Additionally, the UGV's path waypoints, pre-generated by a global planner within the CCC, are transmitted in real time to the MQTT broker. For this purpose, an MQTT client is used in a LabVIEW program, installed on the UGV's on-board PC. This client publishes the waypoints (latitude, longitude and altitude) obtained with an Real Time Kinematic (RTK) GNSS, used for the terrestrial robot. In this way, the position of the UGV (with an error of a few centimeters) can be combined with the detections made from the two on-board BLE scanners. Thus, the UGV tracking is visible in the CCC.

In the case of the UAV (DJI Matrice 600, detailed in Appendix A), it also carries a battery-powered ESP32-based scanner, which is a minimal extra payload for the UAV. The data is stored on its micro SD card too. Flight data is extracted from the Remotely Piloted Aircraft System (RPAS), and processed after the flights. The UAV achieves highly accurate tracking thanks to its three GNSS antennas, which ensure precise positioning.

3.2.2.2 Experimental results

The proposed BLE-based victim detection system was tested and evaluated in a realistic SAR experiment conducted in the LAENTIEC operational area during the JEMERG XV exercise. The scenario simulated a fire involving car wrecks and victims dispersed along a stream surrounded by dense vegetation. The disaster area was set up near the entrance of a tunnel, located in a ground depression with a gully. This created a negative altitude for the UAV flight, as the UAV's altitude was measured relative to its take-off site.

The BLE-based PVs were positioned in a hidden area of operation susceptible to interference. This approach enabled controlled yet realistic evaluations of the system's performance under varying environmental and operational conditions.

Cuadriga navigated the LAENTIEC operational area, collecting environmental data along its route. Its path is determined by a planner that defines the trajectory based on either a destination specified by an operator at the control center or an event-triggered response in an area of interest. Events are prioritized based on their importance, influencing the robot's route accordingly (further details are provided in Chapter 5).

All the victims' positions were known, as they had been registered before the experiments to establish a ground truth. However, Cuadriga performed path planning without considering these positions. In practice, the tele-operator did not know the victims' locations and chose to create a path that covered as much of the area of interest as possible, increasing the probability of detection. On the other hand, during the autonomous routes, the UGV only focused on traveling from one point to another along an optimal path, without prioritizing points of interest related to the victims' positions.

UGVs traversed rough terrain at a slow pace during the search phase, moving at approximately $1m/s$. The strategy focused on covering as much ground as possible within the designated area of interest. While Cuadriga meticulously surveyed the challenging zones and navigated throughout the operational area, the UAV simultaneously conducted aerial reconnaissance over the same region.

Figure 3.8 shows the positions of the BLE nodes emulating the victims, along with the two exploration flights performed by the UAV and the autonomous path followed by Cuadriga.

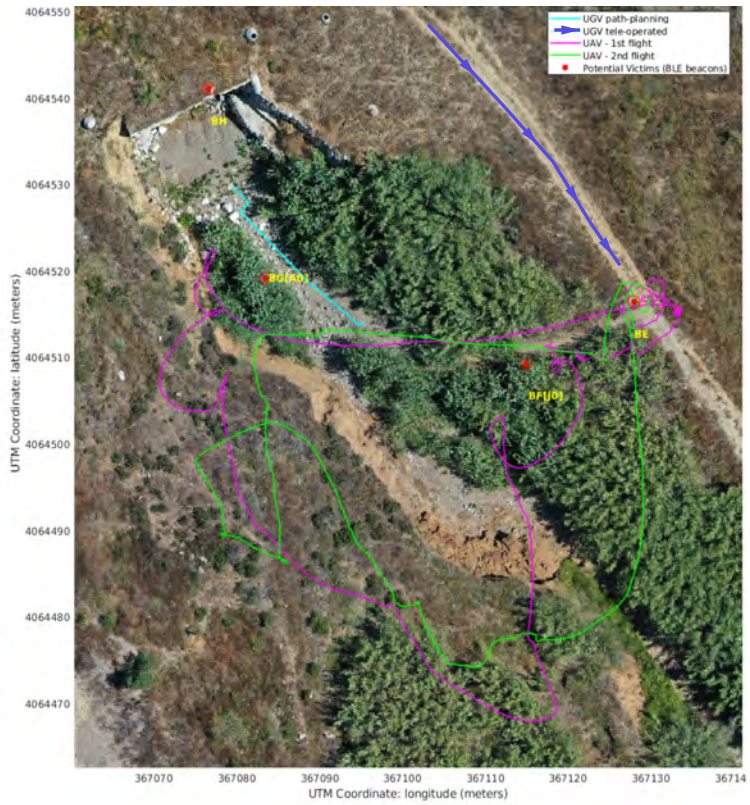


Figure 3.8 Path of the Robots

Both robots were equipped with the BLE ESP32-based scanner. Table 3.1 summarizes the PV detections achieved by each mobile platform.

The execution of the exercise proceeded as follows. The UAV was tele-operated, conducting two exploration flights (F1 and F2) around the operational area while carrying an ESP32 BLE scanner. These flights were used to detect PVs. Across the area, four static locations were established to place a set of six BLE transmitters emulating the PVs. These transmitters were labeled BE, BF, BG, and BH (positioned inside the tunnel on a rubble pile), along with two half-buried BLE nodes labeled A0 and J0.

Additionally, Cuadriga followed a designated path in a controlled area near the tunnel entrance to detect PVs. Furthermore, a transmitter node (BD) was mounted on Cuadriga to allow the UAV to detect its location. The results are presented in Figures 3.9 and 3.10. In Figure 3.9, the UAV's take-off and landing times are indicated.

3.2.2.3 Evaluation

The proposed system is designed as a complementary tool for locating victims, utilizing BLE scanners tested with ESP32-based BLE transmitter nodes deployed across a wide and diverse

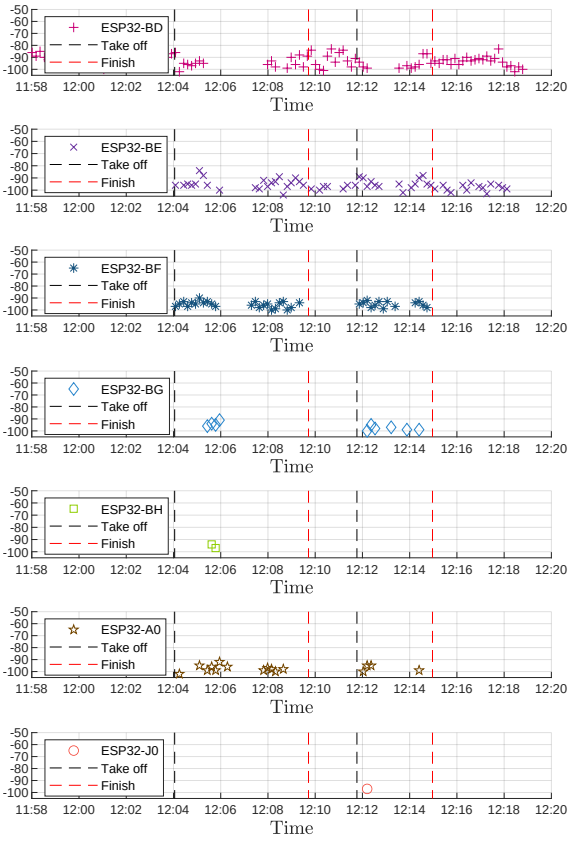


Figure 3.9 RSSI (dBm) from UAV

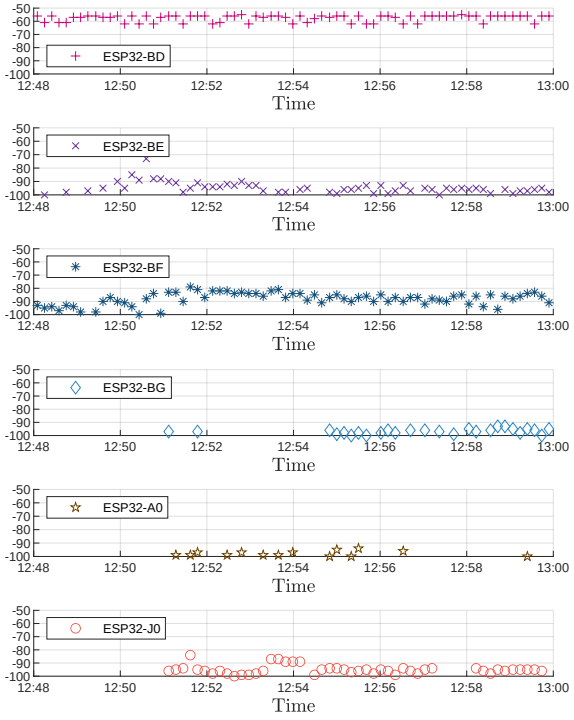


Figure 3.10 RSSI (dBm) from UGV

area to emulate PVs. The effectiveness of the cooperative system, comprising a UAV and a UGV, in detecting BLE nodes has been successfully demonstrated.

The BLE nodes were successfully detected even in conditions where dense dust, light debris, or vegetation obscured visibility from the air, making traditional search methods less effective. The UAV's BLE scanner, with a direct LoS to the ground, demonstrated higher sensitivity to transmissions compared to the UGV's scanner, which is more affected by interference and obstacles.

Table 3.1 Detections from the robotic platforms during the second setup.

Platform	**BD**	**BE**	**BF**	**BG**	**BH**	**A0**	**J0**
UGV (ESP32)	71	60	70	30	0	16	48
UAV (ESP32)	87	57	36	10	2	16	1

The UAV's speed is crucial for effective detection and synchronization. It must be moderate, not too fast to ensure sufficient time for scanning BLE signals within their periodicity, and not too slow to prevent battery constraints that need frequent landings for replacement.

3.3 A WiFi mesh network (802.11 s) to cover underground scenarios

Another critical challenge in SAR scenarios is the frequent loss or complete absence of communications, often due to the damaged terrestrial infrastructure. Victims are frequently isolated and unable to call for help if they are immobile or trapped, while rescuers often venture into areas without coverage, such as tunnels or underground spaces. Although low-frequency devices like walkie-talkies are commonly used, they lack the capability to transmit essential data, such as video, text messages, or images, needed for a swift and efficient response. Additionally, SAR teams are typically connected to a specific radio channel, and due to background noise and the nature of voice communication, the intended recipient of a message may not always be clear. In contrast, a text message ensures clarity and eliminates the ambiguity caused by noise or overlapping transmissions. This highlights the necessity of designing interconnected systems and robots capable of providing robust telecommunications in complex environments. If communications are not possible, the operational capacity of the IoRT is severely limited, restricting their ability to aid in SAR operations.

This section evaluates and compares the behavior of two wireless networks inside an underground scenario: a novel WiFi mesh network based on UGVs with two different roles (scout and repeater) and a 5G commercial network. Two human agents (HA) streamed video to the Internet through the mesh network, composed of static and mobile nodes. Key performance indicators (KPI), such as latency and packet loss, were measured for both HAs for different video resolutions, and for a 5G customer-premises equipment (CPE) onboard the scout-UGV. Both networks have been tested in a prepared disaster scenario in an experimental terrain at the University of Malaga [157].

3.3.1 Design and implementation

The proposed system is based on WiFi, particularly on IEEE 802.11s, a standard that allows wireless nodes to be interconnected in a mesh network. All the devices were chosen to be compatible with 802.11mc, which includes Round-Trip Time (RTT) and Fine Time Measurement (FTM) technologies, useful for accurate time-of-flight measurements of radio

signals between devices and access points. These functionalities are helpful to detect and locate victims in SAR situations (see Chapter 4).

The implemented system is based on a WiFi mesh network composed of several routers, developed by Google, that are Multi-User Multiple-Input Multiple-Output (MU-MIMO). The network is made up of static and mobile nodes. The static part of the network is deployed by human agents, who must establish the network connection from points with Internet coverage to areas of interest without coverage. Subsequently, mobile robots will act according to two different roles. On the one hand, a robot can be located in a position with good coverage and extend the mesh network to the area without coverage. The second role is to explore the underground area, introducing mesh nodes to establish dynamic coverage.

The system was tested in a SAR exercise in a catastrophic scenario involving a vehicle embedded in a dual tunnel (see Figure 3.11). The wrecked vehicle is just in front of an inner opening (internal door), which connects both tunnels (A and B) right in the middle of the dual-tunnel.

The designed mesh network is composed of four WiFi mesh routers: two are static, and two are mobile. Both static relay nodes were placed at the south entrance to guarantee LoS into the dual-tunnel. The mobile mesh nodes were placed on two UGVs: Rambler, acting as a repeater-UGV, and a scout-UGV (Rover J8). Appendix A includes the characteristics of the UGVs used for experiments.

In addition, two human agents (HA), from now on, HA1 and HA2, operated two smartphones, i.e. two user equipment (UE), from now on, UE1 and UE2, respectively, while moving through the dual-tunnel. This situation could correspond to two victims trying to communicate with the outside world, or even two SAR agents trapped and attempting to establish contact using their smartphones. To provide them with Internet service, the following setup was proposed.

First, the repeater-UGV (Rambler) must move to an area where it has good 5G coverage to guaranty good latency and throughput for the 5G router it carries (CPE Pro 2 H122-373) developed by Huawei: a mobile router with 5G and WiFi 6 (802.11ax) connection. Rambler also carries the master node of the WiFi mesh network. This node creates its own WiFi network but does not have its own Internet connection. Thus, the master node is connected to the 5G CPE via an Ethernet cable, functioning as an Internet extender. However, the mesh network's master distributes Internet through a wireless network independent of the CPE's WiFi network (which is also available close the the repeater-UGV). Therefore, the master node creates and manages a WiFi mesh network that is extensible through relay nodes: two static at the tunnel south entrance, at a fix distance from Rambler, and one mobile node

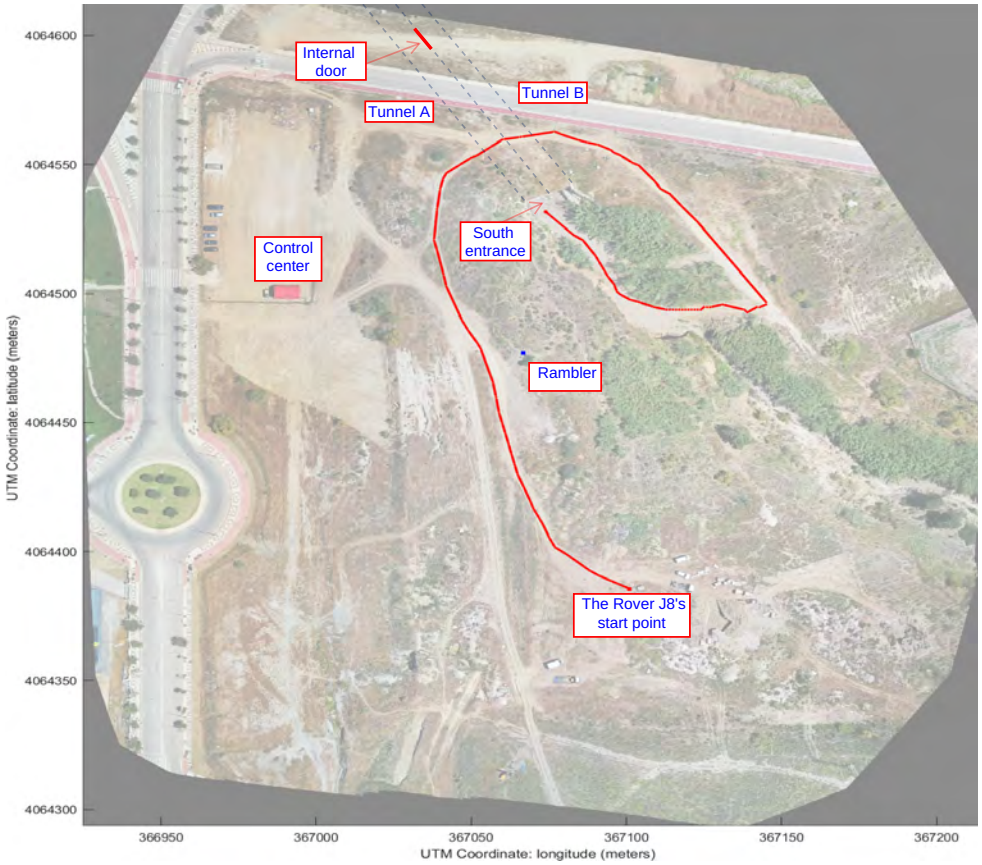


Figure 3.11 Orthophoto showing the trajectory followed by Rover J8 to reach the south dual-tunnel entrance. Rambler remains static once the master is linked to some mesh node.

carried by the scout-UGV (Rover J8), thus changing its position relative to the rest of the mesh nodes.

The scout-UGV's role is to introduce a relay node into the tunnel as it moves from its south entrance to the end. In addition, Rover J8 carries a CPE to test the 5G commercial network inside the tunnel. In this case, the CPE is not connected via Ethernet to the mesh node because its purpose is to provide Internet access to the dual-tunnel via 5G. In contrast, the mesh node receives Internet connectivity from the repeater-UGV. Thus, the scout-UGV extends two different networks inside the tunnel: the WiFi mesh network, whose Internet connection comes from Rambler (located outside the tunnel), and the scout-UGV CPE's 5G network.

The experiment supervisor makes use of a 5G CPE similar to those on UGVs, and can receive and send commands, such as ping tests, to the deployed UEs (connected to the WiFi mesh network) and to Rover J8's computer (connected to the Rover J8 CPE's network), as they are all included in a Virtual Private Network (VPN), which is based on ZeroTier.

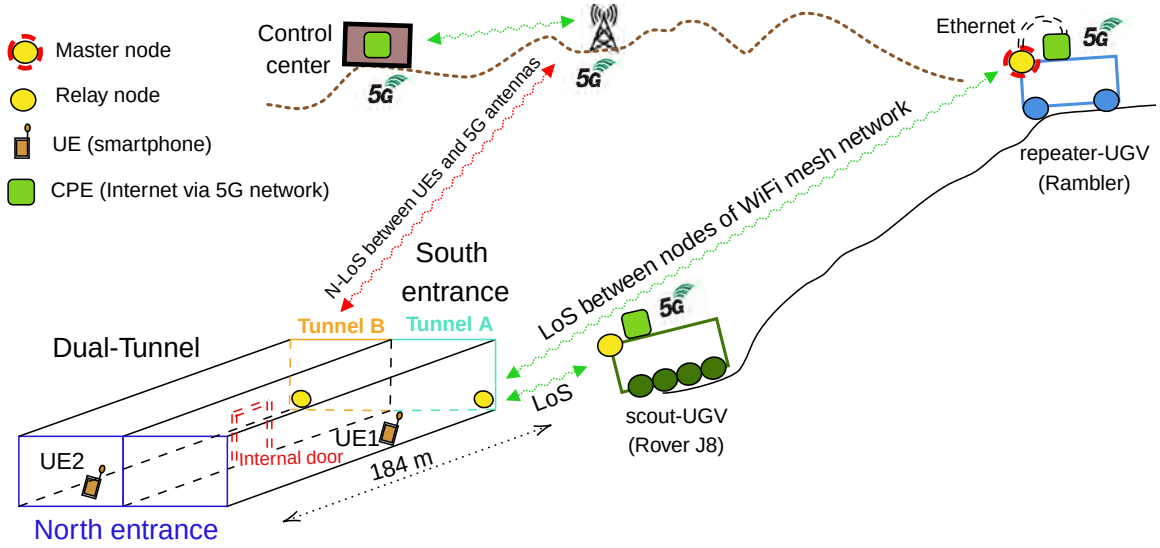


Figure 3.12 Scheme of a WiFi mesh network providing Internet to a disaster underground scenario using two UGVs

HA1 and HA2, connected to the mesh network deployed by the SAR agents, streamed video to the FCC where images and KPIs were monitored, processed and stored in real-time.

3.3.2 Experimental results

The experiments were performed in the LAENTIEC operational area, which includes two parallel sewer tunnels 184 m long, 6 m wide and 4 m high. The thickness of the concrete walls is 50 cm. The tunnels are internally connected by a rectangular door-like opening at half their length (internal door). The test ground was set up as a simulated disaster site, with rubble mounds and a crushed vehicle to mimic the conditions of emergency missions.

The experiments took place during the XVII JEMERG, held on June 9, 2023. Figure 3.11 shows an aerial view of the experimental setup.

Before the experiment started, the static relay nodes were deployed in the outer corners of the south entrance of tunnels A and B (see figures 3.11 and 3.12). Rambler was also set up and placed in a position `(36.71668869, -4.48848178)` where the 802.11 master node had LoS with both static relay nodes. They were at a horizontal distance of approximately 62 m from Rambler, which is a feasible distance to interconnect two mesh nodes stably, according to preliminary tests. These three static mesh nodes remained in their positions for the whole exercise.

HA1 and HA2 carried two smartphones (UE1 and UE2, respectively), and they had to connect them to the WiFi mesh network when it was available. Both operators were

Table 3.2 Configurable Parameters for IP Webcam

Category	Parameter	Description/Options
Video	**Resolution**	320x240 up to 4K (device-dependent)
	Image Quality	Low, Medium, High
	FPS	1 to 30 frames per second
	Orientation	Vertical or Horizontal
	Video Codec	MJPG or H.264 (if supported)
	Night Mode	LED activation or auto-brightness adjustment
Audio	**Microphone**	Enable/Disable microphone
	Audio Quality	Low, Medium, High
	Echo Cancellation	Background noise reduction
Connection	**Port**	Default: 8080 (can be changed)
	Authentication	Set username/password
	Cloud Streaming	Remote access via services like lvideon
Sensors & Advanced	**Motion Detection**	Trigger alerts or recording on motion
	Notifications	Email or Telegram alerts
	Phone Sensors	Access accelerometer, gyroscope, battery data
Protocols	**RTSP**	Integration with apps via: rtsp://[IP]:8080/h264_pcm.sdp
	HTTP/HTTPS	Browser access
	FTP	Save captures/videos to FTP server

instructed to move through the tunnels while streaming video using the Android application IP Webcam (see Table 3.2), at three different resolutions: 352×288, 640×360, and 640×480 pixels. By default, the app streams video in MJPEG format.

The main objective was to generate video traffic at varying resolutions, each under the same quality conditions, to assess the impact on network performance. MJPEG was selected as the streaming format due to its simplicity and broad compatibility with standard browsers and tools, which facilitated rapid deployment in the testing environment. Although MJPEG is relatively inefficient in terms of compression, it was suitable for local testing scenarios where sufficient bandwidth was available. More efficient codecs—such as H.264, or ideally H.265—could have reduced the data rate significantly, enabling higher resolutions with less bandwidth consumption. However, MJPEG offered a practical balance between compatibility, ease of integration, and responsiveness for the field technicians. A stream quality setting of 50% was selected to balance image fidelity and bandwidth consumption across all resolutions, with the HAs streaming video at a fixed frame rate of 30 fps.

Rover J8 was tele-operated in LoS using a radio link to get to the tunnel from another location (see Figure 3.11), joining the human operators once they were already inside the tunnel.

The system setup consisted in several steps:

1. HA1 and HA2 positioned static relay nodes at the entrances of tunnels A and B, ensuring LoS with both the tunnel interior and the master node on Rambler. Then, they notified, using the walkie-talkie, that the mesh network was ready.

2. HA1 and HA2 entered tunnel A through its south entrance and connected their smartphones (UE1 and UE2) to the WiFi mesh network as soon as it became available. They verified with the control center that communications through the VPN were working correctly.

3. HA1 and HA2 went through tunnel A up to the internal door, crossed it, and went back through tunnel B to the south entrance of the dual-tunnel, to perform an initial throughput test. Once they knew the required bandwidth (Mbps), which HA1 and HA2 determined based on feedback from the supervisor indicating that the received video had minimal lag and was clear enough to visualize the agents' surroundings, the supervisor requested that they keep transmitting simultaneously and proceed to enter through tunnel B.

3.3.2.1 Development of the exercise

The exercise lasted approximately 21 minutes, excluding networks and agents deployment, i.e., from the moment the HAs enter tunnel B (at 13:34:49, local time) to the moment Rover J8 exits tunnel A (at 13:54:58). A detailed event log for the experiment is presented in Table 3.3.

3.3.2.2 Key performance indicators

This SAR exercise aimed to measure four KPIs to compare the behavior inside an underground tunnel of our WiFi mesh network against a commercial 5G network, including packets transmitted, packets received, RTT, i.e., latency, and throughput. At the same time, two smartphones carried by two HAs streamed video at different resolutions to a remote control center.

These KPIs were measured for UE1 and UE2, as well as for the CPE on-board Rover J8. All relevant events in terms of performance are highlighted in Figure 3.13, 3.14, and 3.15 with a vertical dashed line.

From the analysis of the event log, and the latency figures, some results can be highlighted (see Table 3.4):

- Latency raised in UE1 and UE2 as the human operators moved along the tunnels, reaching an average value of 457.46 and 334.07 ms for RTT during the time they were on north entrance, and without Rover J8. However, RTT had an average of around 186 ms for the rest of the experiment, which can be practical for many applications.

Table 3.3 Event log for the exercise considering agents inside the tunnels

Time	Device	Highlighted events during the course of the experiment's items
13:34:49	UE1	HA1 sets a resolution of 640x360, and walks along tunnel B close to the inner wall of the dual-tunnel, followed by HA2. For $2'17''$, UE1 moves up to it reaches the internal door, crosses it and looks back to the south entrance.
sS	UE2	HA2 sets a resolution of 352x288 for UE2, and walks following HA1. Then, he also cross the internal door to tunnel A and points with his camera to the south entrance of tunnel A.
13:37:06-13:39:44	CPE	The scout-UGV arrives to the dual-tunnel and enters the tunnel A.
	UE1	HA1 stands next to the wrecked vehicle (embedded in the wall), to make way for Rover J8. A latency peak can be seen in Figure 3.13 and 3.14 (central area of blue period), due to this.
	UE2	HA2 stands under the internal door's frame, losing LoS with respect to any node of the mesh network. He does this, to let the Rover J8 through. A latency peak can be seen in Figure 3.13 and 3.14 due to the agent hiding behind the vehicle.
13:39:40	CPE	The scout-UGV completes its first movement since entering tunnel A. At this point, it stops, two meters from the internal door. This position was considered adequate to cover the rest of the tunnel, where UEs have worst KPIs, considering only the static relay nodes.
	UE1	HA1 starts walking from the internal door to the north entrance of tunnel A. He walked $26''$ towards the end of the tunnel A, while showing with his smartphone camera the robot standing in the middle of the tunnel A. The video smoothness is goot at the control center, before changing the resolution to a higher one.
	UE2	This operator walks alongside HA1, while transmitting video to the control station, showing HA1 (moving) and Rover J8, which remains static in the center of the tunnel. The transmitted video is smoothly viewed at the control center, from where a request is received to increase the resolution by one step before continuing to walk towards the end of the tunnel A.
13:40:06	UE1	HA1 changes resolution to 640x480. Rover J8 is in the center of tunnel A providing better KPIs, especially throughput, as the node it carries improves the coverage of the tunnel, specifically in its end.
	UE2	HA2 changes resolution to 640x360.
13:43:22	UE1	HA1 reaches the north entrance of tunnel A and moves his UE (which continues to transmit in 640x480 resolution) showing his surroundings, including HA2
	UE2	HA2 reaches the north entrance of tunnel A, close to HA1, and moves his UE (which continues to transmit in 640x360 resolution) showing his surroundings, including HA1.
13:44:07	UE1	HA1 changes resolution to 640x360.
	UE2	HA2 changes resolution to 640x480.
13:49:52	CPE	The scout-UGV has moved, stuck to the outer wall of tunnel A (west side), and has now stopped 20 metres from the north entrance, where both HAs are. However, HAs have been moving through tunnel B.
	UE1	HA1 is showing how Rover J8 stops after advancing a few meters from the internal door, to improve the coverage of both UEs, while they stay at the north entrance of tunnel A, transmitting video at the highest possible resolution (he tests 640x640) but the execise supervisor request to reduce the throughput since the bandwidth does not seem to support that resolution, along with what the other agent was already transmitting. HA1 quickly establishes the above resolution (640x480).
	UE2	HA2 is in tunnel B now, having exited tunnel A through its north entrance, and is now transmitting video to the control station from the north entrance of tunnel B without Los to the Rover J8's node. The video flow is not as smooth, but it is correctly maintained without too many cuts, thanks to the existing node at the beginning of this tunnel. However, because the partner raised its resolution as well, the joint throughput was not well supported by the mesh network.
13:50:01	UE1	HA1 moves away from the scout-UGV and goes outside through the north entrance of tunnel A. Subsequently, it passes to the parallel tunnel (B), where HA2 is already located. HA1 approaches HA2 to check together if the video resolution configured is being well received at the control center, after the preliminary issues.
	UE2	HA2 remains 15 meters away from the north entrance of tunnel B, transmitting video. He waits for the arrival of HA1 to check the resolutions and set up a new one. It is time to decide if it is possible to go up one more step, or if it is advisable to go down one more step.
13:50:20	UE1	HA1 changes resolution from 640x480 to 352x288, and maintains it until the end of the experiment.
	UE2	HA2 changes resolution from 640x480 to 640x360, and maintains it until the end of the experiment, as the engineer at the control center comments that the bandwidth in the dual-tunnel seems to be exceeded by the data rate generated by the HAs.
13:52:50 (see Figure 3.16)	CPE	Rover J8 starts its movement from the north entrance of tunnel A towards the south entrance of same tunnel. It will complete its journey in $2'10''$ (from 13:52:50 until 13:55:00), having HAs behind it from the middle of the tunnel, escorting them to leave tunnel A by its south entrance.
	UE1	HA1 starts walking from the internal door of the dual-tunnel, following Rover J8 to leave tunnel A, while streaming video.
	UE2	HA2 starts walking from the internal door of the dual-tunnel, following the scout-UGB to leave tunnel A, while streaming video.
13:54:00 (see Figure 3.16)	CPE	Rover J8 has approximately 46 metres to go to the south entrance of tunnel A (it is just between the internal door and the south entrance).
	UE1	Ha1 is close to the internal door but attached to the outer wall of tunnel A.
	UE2	HA2 is just at the central door, also walking throughout tunnel A.
13:54:45 (see Figure 3.16)	UE1	HA1 remains attached to the inner wall of tunnel A. He stays behind the Rover J8's operator, for safety reasons, at a distance of approximately 10 metres from the scout-UGV.
	UE2	HA2 remains attached to the outer wall of tunnel A. This agent stays behind the Rover J8's operator, for safety reasons, at a distance of approximately 14 metres from the robot.
13:54:58 (see Figure 3.16)	CPE	Rover J8 leaves the dual-tunnel.
	UE1	HA1 stands at the south entrance showing the remote control center how the Rover J8 is leaving the site through the gully.
	UE2	HA2 stands at the south tunnel entrance showing the remote control center how the scout-UGV is leaving the site through the gully.

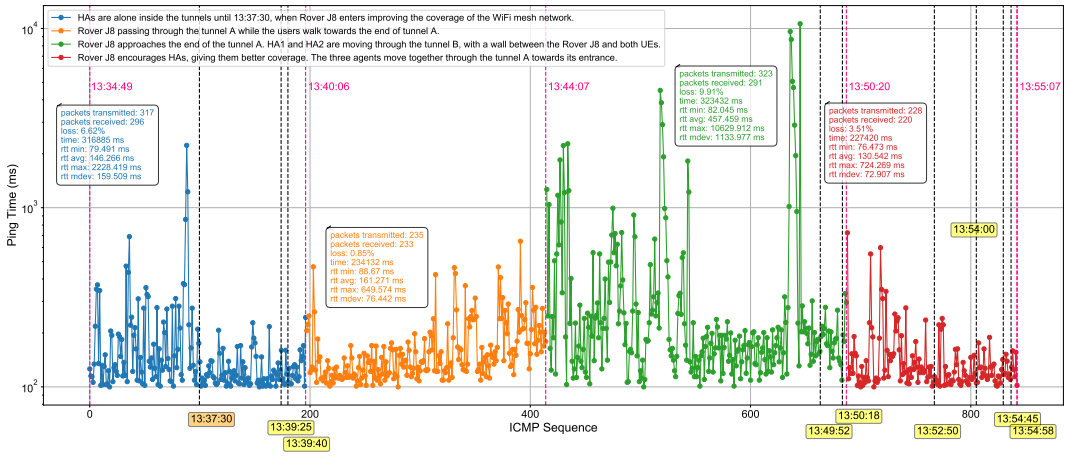


Figure 3.13 latency (ms) from control center to the UE1

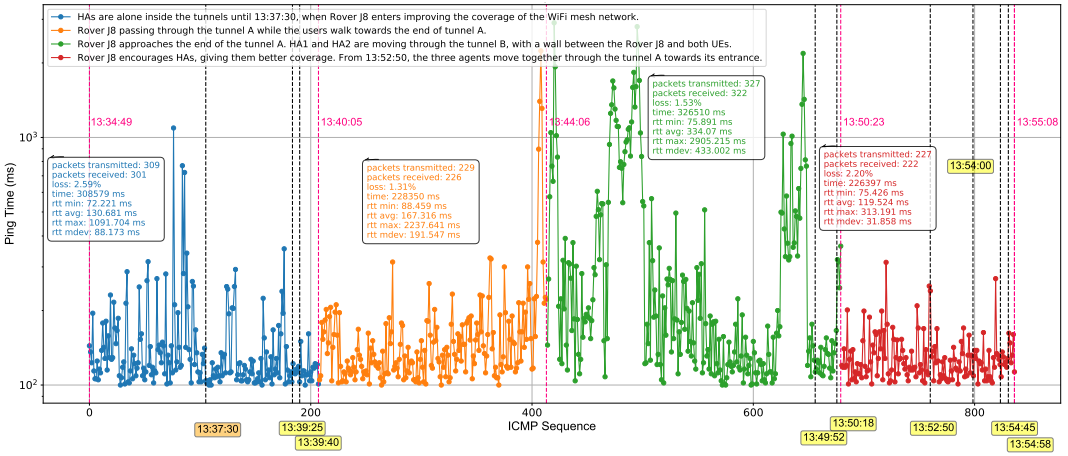


Figure 3.14 latency (ms) from control center to the UE2

Table 3.4 Latency test results for different devices.

Device	Period (hh:mm:ss)	Tx Packets	Rx Packets	Loss (%)	Time (ms)	RTT Min (ms)	RTT Avg (ms)	RTT Max (ms)	RTT Mdev (ms)
Smartphone 1	13:34:49-13:40:06	317	296	6.62	316885	79.49	146.26	2228.41	159.51
	13:37:30-13:44:07	235	233	0.85	234132	88.67	161.27	649.57	76.44
	13:44:07-13:50:20	323	291	9.90	323432	82.04	457.46	10629.91	1133.97
	13:50:20-13:55:07	228	220	3.51	227420	76.47	130.54	724.27	72.90
Smartphone 2	13:34:56-13:40:05	309	301	2.58	308579	72.22	130.68	1091.70	88.17
	13:40:05-13:44:06	229	226	1.31	228350	88.46	167.32	2237.64	191.54
	13:44:06-13:50:23	327	322	1.52	326510	75.89	334.07	2905.21	433.00
	13:50:23-13:55:08	227	222	2.20	226397	75.42	119.52	313.19	31.86
Rover J8's CPE	13:37:30-13:40:06	154	59	61.68	155015	360.14	3187.94	20182.63	5085.17
	13:40:06-13:44:06	221	31	85.97	224104	8329.17	23836.36	47741.68	11949.57
	13:44:06-13:50:25	324	97	70.06	328106	721.26	12164.28	35629.51	9827.40
	13:50:25-13:55:08	214	147	31.30	215185	69.09	4583.79	20894.63	6873.22

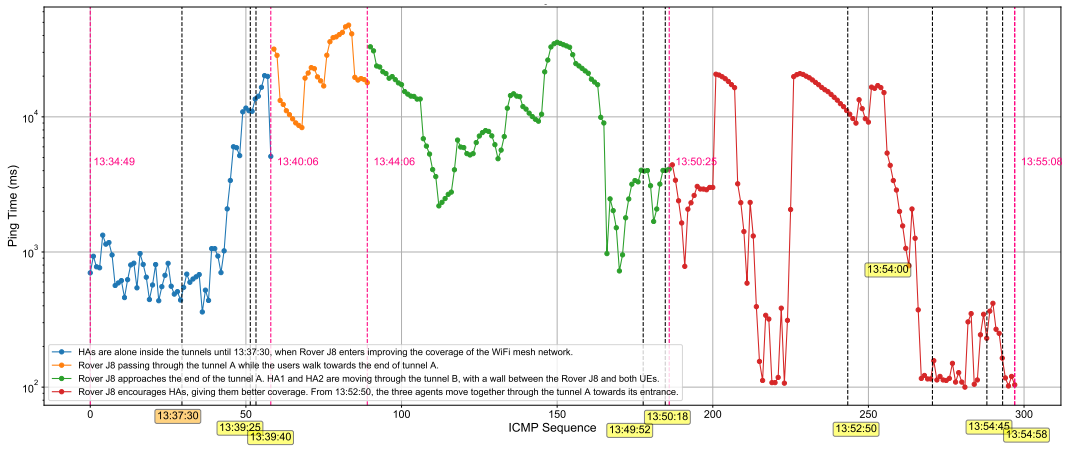


Figure 3.15 latency (ms) from control center to the Rover J8's CPE

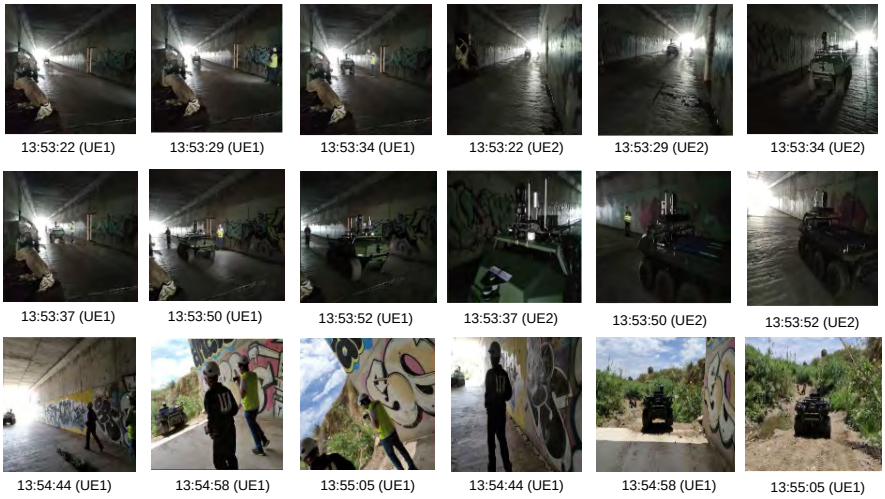


Figure 3.16 Sequence of frames received at the control center from each UE during the Rover J8 tunnel exit path.

- Video streaming was received on the control center at the different resolutions tested. Reception was smooth, except when HAs were at the end of tunnel A or B, transmitting at resolution 640x480, and Rover J8 was still in the central position of tunnel A.
- Latency was high for the CPE onboard Rover J8 for the whole experiment, with latency values well over seconds.
- Latency improved for both UE1 and UE2 when Rover J8 entered the tunnel, due to the mobile mesh node.

- Mesh network nodes made use of a 5 V, 3 A wireless battery, which enabled a very efficient deployment. They supplied the mesh nodes for the preparation and execution of the exercise (1 hour), without any problem.

3.3.3 Evaluation

The experiment involved testing a WiFi mesh network against a commercial 5G in 184-meter-long shallow buried tunnels of concrete. Two human agents streamed video to the Internet through the mesh. The configuration of the presented mesh is novel as the extension of the network is done with two UGVs with different roles: a repeater-UGV, with good coverage, which carries the mesh master node, and a scout-UGV, which explores the area until it reaches the tunnel, providing coverage to the users inside the tunnel. Thus, the network is deployed from an outdoor area to an indoor area, thanks to the mobile relay node on-board the scout-UGV, getting the repeater-UGV to maintain LoS with the scout-UGV.

The designed setup proves to be effective in enhancing the KPIs of the static mesh network. It provides coverage that allows smooth video streaming from two HAs in disaster-stricken areas where the 5G network is not operational, which was demonstrated by introducing a 5G CPE on-board the scout-UGV into the tunnel.

In addition, video transmission at a resolution of 640x480 proceeded without interruptions to a remote control station. Furthermore, the mesh network consistently maintained a latency of less than 200 ms throughout the dual-tunnel scenario.

To evaluate the impact of different video encoding formats on network performance and system integration, a series of tests were conducted using the IP Webcam application on a Google Pixel 7 Pro. The goal was to simulate real-world conditions in local and constrained environments, considering both codec selection and compression level.

MJPEG transmits each frame as an independent JPEG image, making it particularly suitable for real-time applications such as computer vision and browser-based monitoring, due to its low latency and broad compatibility. However, it is relatively inefficient in terms of compression compared to more advanced formats. As a result, MJPEG is better suited for local scenarios, while H.264—by leveraging inter-frame compression—achieves higher efficiency and is generally preferred in remote or bandwidth-constrained settings.

In this study, MJPEG was chosen because two technicians were required to conduct local tests inside the tunnel, and the format's direct compatibility with standard browsers and tools simplified their workflow. Moreover, the available bandwidth was sufficient to support low-resolution video streams without affecting system performance. This configuration enabled rapid deployment and reliable real-time visual feedback during field operations.

For long-range or bandwidth-sensitive transmissions in ROS 2-based systems, more advanced codecs such as H.265 (HEVC) would be preferable. H.265 offers up to a 50% reduction in bitrate compared to H.264 for equivalent video quality, thereby helping to minimize network congestion and improve scalability in multi-robot deployments. However, H.265 was not adopted in this case due to compatibility limitations with available ROS 2 tools and the lack of universal hardware decoding support, which could compromise real-time performance and system robustness.

When using the IP Webcam application, the selected video codec (e.g., MJPEG or H.264) defines the compression method, while the configured quality percentage (e.g., 40% or 100%) adjusts the intensity of that compression. Lower quality values reduce bandwidth usage at the expense of image fidelity, whereas higher values maintain better visual quality with increased data rates. This distinction is important when balancing image clarity with network efficiency.

To support these considerations, Table 3.5 presents the estimated bandwidth required to stream video from the device at 30 and 60 frames per second, across two compression levels (100% and 40%) and three resolutions. As expected, MJPEG results in significantly higher bandwidth consumption, particularly at high frame rates and high quality. H.264, in contrast, maintains a much lower data rate under equivalent conditions, reaffirming its suitability for resource-constrained or remote environments.
The estimated bandwidth in megabits per second (Mbps) for each video stream was calculated using the following expression:

$$\text{Bandwidth (Mbps)} = \frac{W \cdot H \cdot \text{FPS} \cdot \text{bpp}}{10^6} \tag{3.1}$$

Table 3.5 Estimated bandwidth (in Mbps) for MJPEG and H.264 formats at two quality levels on a Google Pixel 7 Pro.

Quality	Resolution	FPS	H.264 (Mbps)	MJPEG (Mbps)
100%	352×288	30	0.36	3.04
		60	0.73	6.08
	640×360	30	0.83	6.91
		60	1.66	13.82
	640×480	30	1.11	9.22
		60	2.21	18.43
40%	352×288	30	0.18	1.22
		60	0.36	2.45
	640×360	30	0.42	2.76
		60	0.83	5.53
	640×480	30	0.56	3.69
		60	1.11	7.37

where W and H denote the video resolution in pixels, FPS is the frame rate in frames per second, and bpp stands for bits per pixel, which depends on the codec and compression level used. For MJPEG, typical values of 1.0 bpp and 0.4 bpp were assumed for 100% and 40% quality, respectively. For H.264, the estimated values were 0.12 bpp and 0.06 bpp under equivalent conditions. These assumptions are consistent with empirical data presented in [158], which provides a thorough analysis of coding efficiency and bitrate reduction across modern video standards.

It is worth noting that actual bandwidth consumption was not measured in real time during the experiments due to limitations in the monitoring infrastructure. Therefore, the bandwidth values reported in Table 3.5 should be interpreted as theoretical approximations used for comparative evaluation purposes.

However, real-time throughput can be monitored from the receiving PC using the `iftop` utility, which displays live traffic statistics per IP address and network interface. By applying a filter to the smartphone's source IP address, it is possible to observe dynamic variations in bandwidth caused by changes in resolution, video compression, or scene complexity. For instance, the following command[1] captures the incoming data rate over a 60-second interval and stores the result for further analysis:

```
sudo iftop -i {interface_name} -F {IP_address}/32 \
  -t -s 60 > remote_smartphone_throughput.txt
```

Finally, the WiFi mesh network opens up possibilities for future remote tele-operation of the scout-UGV, enabling real-time video visualization from its onboard cameras over the Internet.

Future research will aim to optimize mesh network efficiency in tunnels by exploring different configurations to increase bandwidth. FTM-based multilateration will be integrated to estimate relative distances among mesh nodes, from the outdoor repeater-UGV to interior agents. The study may also investigate autonomous UGV navigation indoors, combining FTM and UWB technologies to provide coverage, Internet access, and victim localization. Additionally, future work could evaluate efficient video compression formats like H.265 (HEVC) on compatible mobile devices, enabling high-resolution streaming with lower bandwidth. These streams could be integrated into ROS 2 nodes to improve real-time monitoring and situational awareness.

[1] Although the local network may use a /24 subnet mask (e.g., 255.255.255.0), the `-F` option in `iftop` expects a network address in CIDR notation. Using `{IP_address}/32` ensures that only traffic involving the specified host is monitored, rather than all devices on the subnet.

3.4 Integration of an LPWAN-based H-WSN in SAR

Providing Internet access is often impractical, creating the challenge of extracting critical information from ROUD scenarios that lack infrastructure. These scenarios usually have restricted accessibility, making it difficult or unfeasible for SAR agents to deploy traditional mesh or tree topology networks for environmental monitoring or victim detection, as has been the case with PAN networks.

Even in these cases, it is not feasible to make use of data MULEs, as the vehicle is not able to get close enough to the area of interest due to physical barriers, environmental hazards, or operational risks. Moreover, even if the data MULE could operate, there might be no infrastructure to offload the collected data, whether due to the complete lack of network coverage, physical inaccessibility of infrastructure, or the excessive distance to the nearest facility, rendering physical transportation of data infeasible.

Consequently, there is an urgent need for innovative, durable communication systems capable of operating in harsh and isolated environments, ensuring reliable collection and delivery of critical information without relying on conventional networks or proximity to established infrastructures.

This thesis proposes using LPWAN technologies to gather information from scenarios without existing infrastructure by leveraging remote systems located outside the operational environment. For this purpose, an H-WSN is designed where mobile platforms have a double role, being a mobile data source and sink. These mobile nodes can operate with and without the Internet. Furthermore, the implemented network, which is a star-of-stars, is designed to favor the cooperation of the network's static and dynamic CNs. To this end, a novel multicasting strategy is established to allow data acquisition and sharing, regardless of whether they have Internet access.

For the development of this contribution, LoRa technology has been chosen as the most suitable technology for ROUD scenarios among all LPWANs, as analyzed in state of the art (see Chapter 2).

3.4.1 LoRa

3.4.1.1 Design and implementation

A LoRa H-WSN is designed to propagate and gather information through static SGs and mobile nodes (vehicles equipped with SGs and/or CNs) in ROUD scenarios.

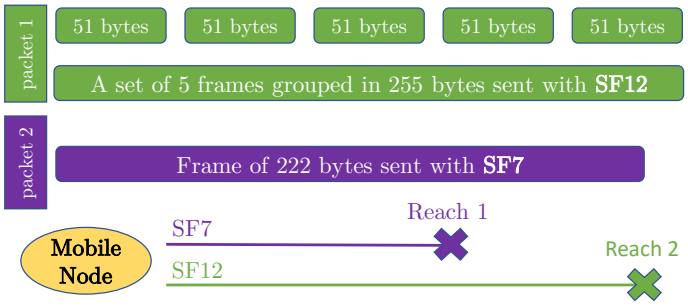
EDs within a static or mobile SG generate data frames of varying lengths, depending on the SF used (ranging from 12 to 7), within LoRa's allowable range of 51 to 222 bytes.

The SG consolidates all measurements from its co-located EDs into a unified data frame called a virtual frame. This approach is feasible because all EDs within an SG share the exact physical location, whether in a static setup or onboard a vehicle. For mobile SGs, the CN onboard the vehicle is the most reliable receiver of these virtual frames due to its proximity, ensuring minimal packet loss. In this context, the vehicle acts as a ground truth for the data collected, as no other CN in the environment will consistently receive as many packets. Beyond serving as a primary sink for its SG, the mobile CN can sweep the scenario and compare its receptions with those of remote static CNs. This capability enables the creation of a detailed LoRa coverage map in ROUD scenarios. By collecting georeferenced data, including RSSI, SNR, and packet reception rates (PRR), mobile CNs complement the static CNs in identifying areas with weak or absent LoRa coverage. The resulting data, aggregated from mobile and static CNs, facilitates the generation of coverage maps critical for optimizing H-WSN performance in dynamic and unpredictable conditions. This dual role of mobile CNs (as reliable sinks for their SGs and tools for mapping LoRa coverage) enhances the functionality and adaptability of LoRa H-WSN in challenging environments.

Virtual packets (see Figure 3.17) are created by merging small LoRa packets from the EDs in an SG with its GNSS position. This process, managed by software in the local databases of mobile CNs, consolidates sensor data, such as temperature from ED A, CO_2 concentration from ED B, and geo-location from ED C, into a unified virtual packet.

Using small packets enables a higher SF (e.g., SF12), increasing range and reception reliability. This method ensures that multiple sensor measurements are transmitted farther and with greater redundancy. Onboard the UGV, the mobile CN transmits these virtual packets, associating all sensor data with the platform's geo-location. Operators at the FCC can then monitor the combined outputs of the SG, represented as a single entity, as it moves with the mobile platform.

Each CN can function either as a simple CN or as a true link to a remote server, depending on the vehicle's coverage, whether it has access to a WLAN or a WAN. In the event that the



SF	Bit/s (data rate)	Size (bytes)
12	250 (0)	51
11	440 (1)	51
10	980 (2)	51
9	1760 (3)	115
8	3125 (4)	222
7	5470 (5)	222

Figure 3.17 Small frames grouped to relate their information to the vehicle GNSS position

vehicle does not have access to the WLAN or WAN, its role as a data sink will be limited to acting as a data MULE, albeit with the advantage of covering large ranges.

Instead of placing more CNs in the ROUD scenario, CNs can be placed far away from it. In this way, the idea is to provide LoRaWAN coverage of the scenario from remote infrastructures with LoS, while CNs at the edge (interpreted as locations close to the static SGs of the H-WSN) allow the NLoS areas of the ROUD scenario to be served with greater penetrability. Figure 3.18 shows the proposed LoRa-based H-WSN, where the FCC is connected to all the MQTT brokers, i.e., those running at the edge (in the CNs close to the SGs in the ROUD scenario), and those being executed in the cloud, e.g. on the The Things Network (TTN) platform [159]. edge CNs maintain local databases connected to their own MQTT broker, which can be accessed by the CCCA when the edge CN establishes connectivity to the WLAN or WAN.

The LoRaWAN architecture follows a star-of-stars topology, where CNs act as transparent gateways relaying messages between EDs and a central network server via standard IP connections. EDs communicate wirelessly in a single hop with one or more CNs. However, if a CN lacks access to WLAN or WAN (which is expected in a ROUD scenario), it cannot use the LoRaWAN layer. In such cases, a protocol modification is required, enabling the CN to function as a data MULE, storing received messages and forwarding them to the server once connectivity is restored, using only the LoRa-based single-hop link in the interim.

Thus, there are two key challenges:

- Ensuring continuous service from all CNs.
- Enabling mobile CNs to serve as ground truth for SGs located within the same vehicle.

To meet both challenges, it is necessary to implement multicasting across the entire network. The proposed solution addresses three scenarios. In the first, a mobile CN outside any WiFi coverage (WLAN) acts as a data MULE, storing information locally and transferring it to an external database once it reaches a coverage area. In the second, a mobile CN within WLAN coverage but lacking direct WAN access uses its MQTT client to send data through a local broker connected to a remote client at the FCC. Finally, in the third scenario, a remote CN with Internet connectivity, typically positioned in an urban area with LoS to the operational environment, routes the received packets to a cloud-based application server (e.g. TTN), ensuring data accessibility via the Internet. In all cases, these configurations ensure that every CN provides continuous and reliable service to the ROUD scenario.

When a mobile platform embarking both a CN and an SG enters a specific area that is under the coverage of a static CN, both the mobile and the static CNs must be able to receive the same LoRa packet transmitted by the same ED (i.e. multicasting) at any given

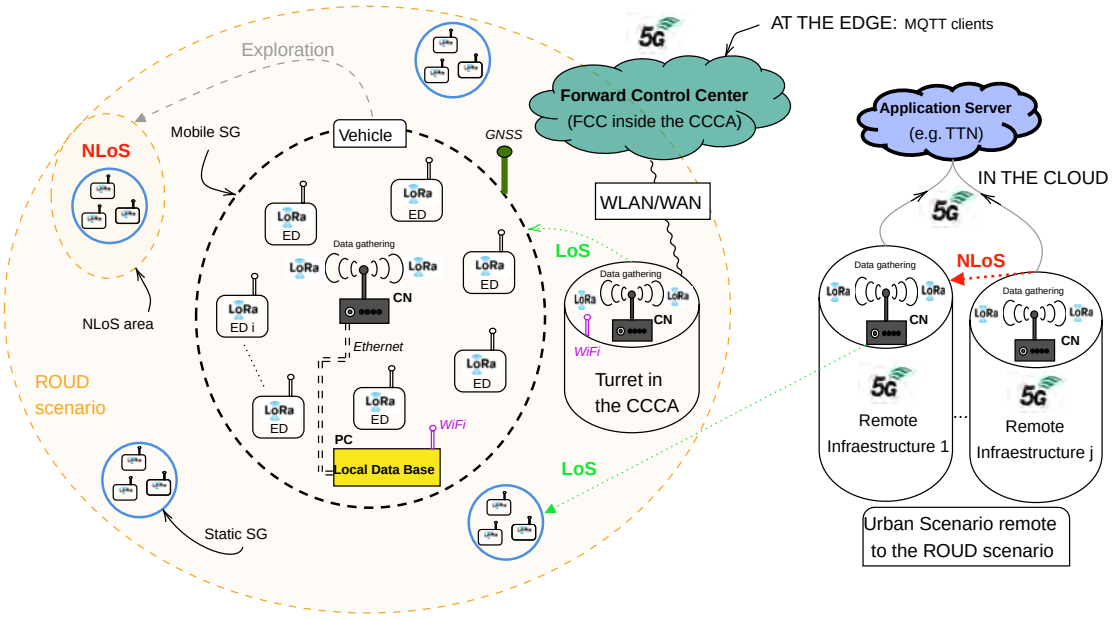


Figure 3.18 LoRa H-WSN architecture for ROUD scenarios

time, enabling detection of areas without LoRa coverage [56]. The way in which the LoRa EDs are activated (ABP or OTAA) has an impact on the type of listening that the CNs will do, especially when they are not connected to the Internet. It is thus essential to set up a multicasting-oriented configuration to be operational even without Internet connectivity.

In addition, considering that the on-vehicle CN must be able to receive all the signals sent from the SG located next to it, except for packets lost due to collisions, it is proposed to use this mobile system as a ground truth to create a coverage map with the LoRa packets sent from the vehicle's SG, validating the coverage provided to this mobile SG by the static CNs.

In OTAA, each LoRa ED must first perform a join procedure with the LNS to establish unique session keys. When each CN is linked to a separate network server lacking a shared backend, the ED must rejoin whenever it moves under the coverage of a different CN. Since the session keys are specific to the original LNS and cannot be shared among servers without a common backend infrastructure, multicasting across separate servers is not feasible. Consequently, if a mobile SG transitions from one CN's coverage to another's, the new CN will lack the necessary keys, preventing it from decoding the ED's messages and hindering communication handover.

OTAA allows EDs to switch networks without reprogramming, as session keys are dynamically regenerated during reconnection, preventing compromise before joining. It provides better performance in rural areas across all SFs [160] but is limited in multicast scenarios when multiple CNs must receive the same message without Internet access.

On the other hand, ABP allows EDs to use pre-set session keys, making it particularly useful when there are various CNs operating online and offline. In the proposed setup (see Figure 3.19), multiple CNs can receive messages from the same ED if configured with the same session keys (AppSKey and NwkSKey). In case a CN does not have an Internet connection, the message reaches its local LNS. In addition, every CN hosts an MQTT broker whose TCP socket (IP and port) is known at the FCC, where the GUI displays the information from all the CNs.

Thus, even if some (or all) CNs do not have an Internet connection, any ED will be able to exchange messages, since all CNs know all EDs. This makes it possible to generate an allowlist with the addresses of the EDs that will be able to exchange messages (encrypted and signed), providing security to the H-WSN.

Thus, edge CNs not connected to the Internet may also receive the same packet as other gateways. Nevertheless, for accurate representation in the CCCA, identifying and filtering repeated packets is essential. To achieve this, the developed application at the FCC reads the sequence number of each received packet and filters out duplicates accordingly.

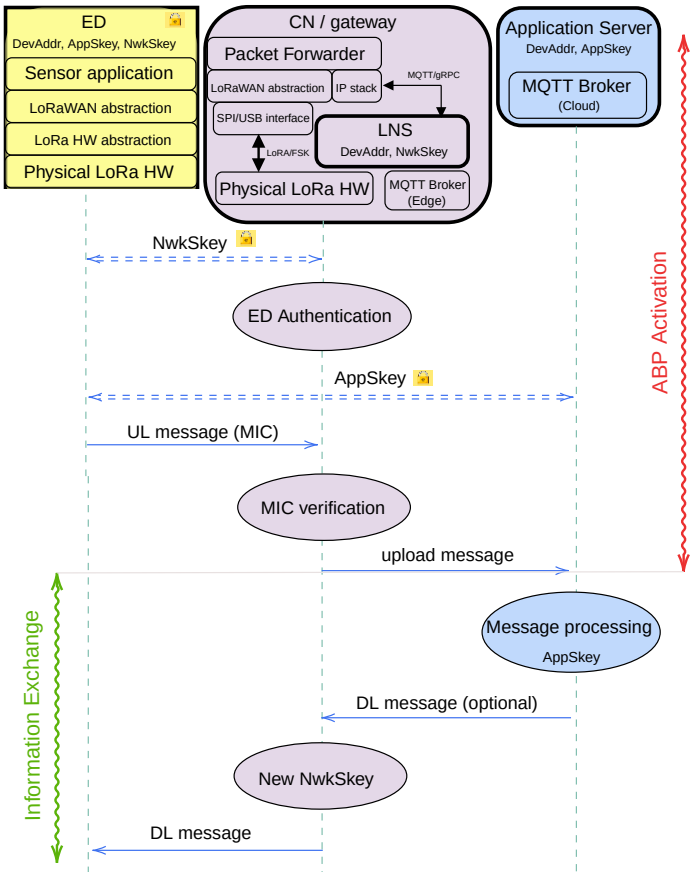


Figure 3.19 Activation By Personalization (ABP) scheme.

The proposed LoRa H-WSN has been developed based on hardware components provided by Libelium and Multitech. The transmitter sensor nodes have in common the Waspmote v12 module, including a LoRa communication module (Microchip), and a GNSS module. The CNs are based on the Conduit IP67 Base Station (Multitech MTCDTIP 220L), which is a ruggedized IoT gateway (Linux-based) that has been configured to recognize the DevEUI of the transmitter nodes. This gateway is equipped with internal memory, a key feature that enables it to function as a data MULE.

The LoRa EDs are based on Libelium's Waspmote Plug & Sense! units, each tailored for specific tasks. The Ambient Control LoRaWAN EU monitors temperature, pressure, humidity, and brightness using versatile probes, with minimal impact on transmission speed. The Smart Environment PRO LoRaWAN EU specializes in air quality, measuring gases like CO, NO, CO, and O, along with standard environmental parameters. The Radiation Control LoRaWAN EU focuses on detecting radiation using a Geiger sensor. Additionally, any of these EDs can function as a simple GNSS receiver, configured to transmit only GNSS data to minimize packet length.

3.4.1.2 Experimental results

The proposed LoRa H-WSN has been deployed in a realistic SAR scenario in the LAENTIEC, providing real-time information to a SAR team to assess the current situation of the operational environment. The deployed LoRa H-WSN consists of three CNs (two static and one mobile), two mobile SGs, and 14 static EDs, grouped into seven SGs. Several human agents are in charge of deploying EDs by groups in terrain in different locations, trying to cover as much area as possible within the LAENTIEC operational area. The SGs' positions are registered with an RTK GNSS portable receiver or, in the absence thereof, using the GNSS module of a smartphone (which leads to an implicit error of about 5 meters).

Two vehicles were deployed for system validation as part of the H-WSN setup, each equipped with SGs and carrying different types of LoRa EDs to transmit various measurements and packet lengths from their respective locations.

One of the vehicles, Rambler (see Appendix A), played a dual role as a mobile SG and CN, being able to retrieve data from concealed areas or locations with higher obstacle density, where static CNs might face coverage challenges. The static LoRa CN, mounted on a 14-meter-high turret for good LoS (see Figure 3.20), also provided WiFi connectivity to agents in the LAENTIEC area.

In addition, a non-robotic 8x8 extreme terrain crewed vehicle (ARGO XTI) served as a mobile SG for operations closer to the FCC. This vehicle was always driven within the coverage of the static CN, so there was no need to include a mobile CN on it.

Figure 3.20 Static CN and a group of EDs on the ground

To validate the interoperability of CNs operating online and offline, a new static CN was installed on the roof of a nearby building, connected to the Internet and integrated with an application server on TTN. By registering the EDs on the TTN application server, any public gateway registered on the TTN network that is in the environment and can receive signals from them (by having LoS with the devices) will be able to uplink the LoRa packets, which will be decrypted on the server.

Both static and mobile CNs in the LAENTIEC operational area are connected to the same LAN as the host in the FCC. Thus, they are linked to an external database in the FCC, whose data tables are managed through a GUI, showing the information in graphs.

However, these CNs do not have Internet connectivity to emulate a network deployed in precarious conditions due to a real disaster. Each CN hosts a local MQTT broker linked to its local LNS.

In the case of Rambler, maintaining a WiFi connection is crucial because its mobile CN needs to link its local LNS with the centralized application server. When the Rambler moves too far from the WiFi access point, it must return to an area with better reception to offload its stored data. In other words, it temporarily acts as a data MULE. In this setup, the local LNS saves incoming LoRa packets to a plain text file using an edge MQTT broker accessible from the FCC (see Figure 3.21). Thus, when the robot regains access to the WLAN, it publishes the saved information to the external database or application server through MySQL queries and MQTT topics, respectively.

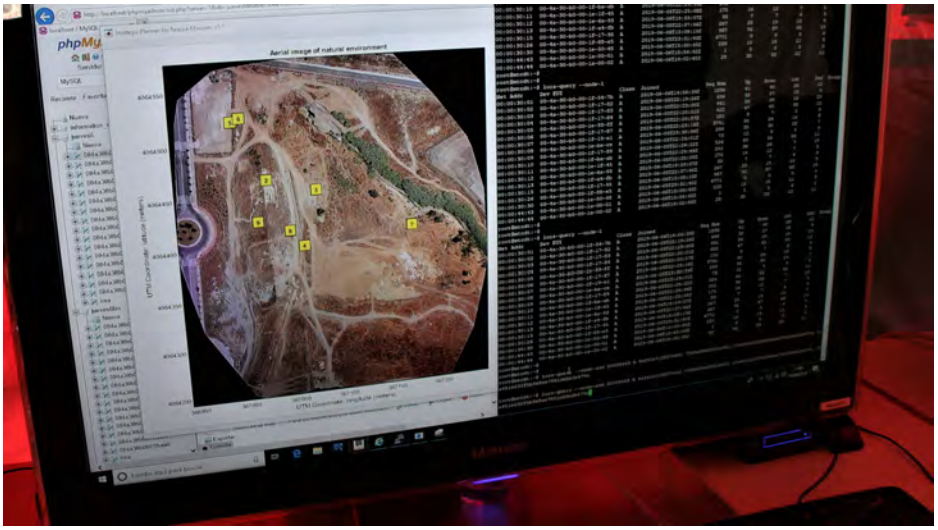


Figure 3.21 GUI in the FCC showing CN data and EDs' positions

A total of 28 EDs (see Table 3.6) were used to transmit LoRa packets with information about the temperature, humidity, pressure, and radiation in the SGs surroundings. All the sensor nodes (set to work as class A EDs) in the different static and mobile SGs have been set up to test different configurations (SF, data rate, packet length) to check the effect of interference from the rest of civil and military signals present in the SAR scenario.

Each SG receives the name ISA-N, where N denotes the group number and is visible on each flag placed next to the group. Thus, for instance, if an ED (identified by its own ID) belongs to the ISA-3, all its measurements are related to that SG. Then, the GUI in FCC associates the information received by each ED with the position registered for its SG.

Specific restrictions on the duty cycle of the EDs, transmission power and their antenna' gain have been imposed regarding the legal requirements. In this case, the transmissions are limited by the time of use of the eight channels of the physical layer (duty cycle equal to 0.1% of use time), thus with a relatively slow transmission frequency.

The gas ED (ID=14) of the ISA-7 group has been located far away and with a bad LoS from the CNs, to test the behavior of LoRa technology in harsh conditions. Moreover the dynamics of gas sensors is the slowest, being able to prepare a data between two and five minutes. Thus, gas sensor nodes have been configured to report every 10 minutes, complying with duty cycle and gas dynamics, which saves on battery power. In these cases, it is not necessary to have hard real-time information. The distance between these EDs and the CN located next to the FCC was 230 m, measured in a straight line.

A robust dataset was collected and processed in soft real-time, showcasing the adaptability of the proposed LoRa H-WSN under two configurations: edge mode (operating

Table 3.6 LoRa Data from the EDs Deployed in SGs

Ending of DevEUI (ID)	SG (packets)	Data (length in bytes)	SF (data rate in bit/sec)	Device Add. (port)
E1F45 (1)	ISA-1 (16)	NO2 (50)	9 (1760)	3011 (93)
E6A93 (2)	ISA-1 (13)	CO2 (50)	10 (980)	3012 (106)
FA4FB (3)	ISA-1 (35)	GPS (50)	12 (250)	3013 (25)
F547B (4)	ISA-2 (44)	T, AP, HUMA, GPS (50)	9 (1760)	3001 (12)
F1782 (5)	ISA-2 (41)	T, AP, HUMA, GPS (50)	9 (1760)	3002 (15)
E1783 (6)	ISA-2 (79)	T, HUMA (50)	12 (250)	3003 (18)
F1784 (7)	ISA-3 (11)	T, AP, HUMA, GPS, BAT (50)	8 (3125)	3004 (21)
E33FF (8)	ISA-3 (79)	CO2, NO2, T, HUMA, p (50)	8 (3125)	3005 (24)
F1787 (9)	ISA-4 (38)	T, AP, HUMA, GPS (50)	9 (1760)	3006 (88)
BECE (10)	ISA-4 (12)	CO, O3 (50)	10 (980)	3008 (101)
F1789 (11)	ISA-5 (112)	T, AP, GPS (124)	9 (1760)	3009 (156)
BAAB (12)	ISA-6 (38)	GPS (50)	11 (440)	3010 (81)
F1794 (13)	ISA-6 (38)	T, AP, H, GPS (50)	9 (1760)	3014 (26)
E5048 (14)	ISA-7 (17)	CO2, NO2 (50)	12 (250)	3016 (100)
F19E1 (15)	Rambler robot (45)	T, AP, HUMA, GPS (50)	9 (1760)	3007 (90)
F1795 (16)	Argo vehicle (48)	T, AP, HUMA, GPS (50)	9 (1760)	3015 (94)

without WLAN/WAN connection) and cloud mode (connected to the WAN and centralized application server). Moreover, data transmission was achieved via two methods: (1) MySQL synchronization from local to centralized databases at the FCC, and (2) MQTT topics hosted on edge brokers, allowing real-time data access.

The experiment demonstrated that CNs operating under these configurations effectively received LoRa packets from mobile SGs operating at significant distances. Figure 3.22 shows Rambler embarking the LoRa SG and CN, acting as a data source and sink while assisting to a victim.

RSSI values ranging from -96 to -113 dBm were observed across all LoRa packets received by the CNs in LAENTIEC, including the mobile CN onboard Rambler and the static CN positioned on the turret. This reliable performance, combined with the dynamic configuration of SFs, allows for mapping RSSI and SNR within the operational area, providing SAR teams with valuable insights into communication conditions.

3.4.1.3 Evaluation

The described scenario highlights the critical importance of acquiring reliable and timely data in environments devoid of conventional communication infrastructures. Traditional solutions, such as mesh or tree topologies, fail when physical barriers, environmental hazards, or operational risks prevent the deployment of intermediate nodes or data MULEs. Under these conditions, the introduction of LPWAN-based solutions, combined with mobile platforms acting as both data sources and sinks, demonstrates a viable alternative. By integrating

Figure 3.22 Rambler embarking LoRa EDs and CN in a ROUD scenario

mobile CNs and SGs into a star-of-stars topology, data can be collected and disseminated without dependence on continuous Internet connectivity. This approach enables the system to gather environmental readings, victim-related data, and other crucial information from remote or obstructed areas, offering improved resilience and adaptability compared to traditional network setups.

Key insights included:

- Dynamic LoRa channel allocation: 675 packets were transmitted using 8 EU 863-870 MHz channels, with downlink communication utilizing an additional channel (869.525 MHz). Most transmissions concentrated in the first three channels due to sub-band limitations.
- Optimized sensor integration: static SGs combined fast and slow dynamics for diverse data rates, enabling comprehensive monitoring. Depending on the type of ED, different SFs have been set within the same SG.
- Reliable packet delivery: High SF values ensured successful transmission from EDs, even out of LoS (e.g., ISA-7), with data update rates between 6 and 9 seconds. Using high SFs in the EDs that make up the mobile SGs allows more information to be received from the mobile sink, something that has already been tested in urban environments [56].

The proposed LoRa/LoRaWAN system has proved the viability operating in disconnected edge mode or centralized cloud mode, providing soft real-time data flow and robust communication for SAR teams in SAR scenarios.

Despite the promise of LPWAN-based H-WSNs, several challenges must be addressed to ensure robust operation in ROUD scenarios:

- **Intermittent Connectivity:** Ensuring seamless transitions between offline (edge-only) and online (cloud-connected) modes requires advanced caching, data synchronization strategies, and autonomous decision-making at the network's edge. The proposed multicasting technique, based on ABP mode and a combination of local and remote MQTT brokers allows to extract data for extended periods without WLAN and WAN access through mobile sinks.

- **Energy Constraints:** Battery-powered EDs and CNs must minimize energy consumption while maintaining reliable communications. Adjusting transmission rates, employing energy-efficient protocols, and optimizing hardware configurations are crucial to prolonging operational lifespans in the field.

- **Scalability and Interference:** Increasing the number of EDs or CNs can lead to channel congestion and higher packet loss. Techniques for efficient channel allocation, adaptive spreading factors [161], and dynamic duty-cycle adjustments must be employed to maintain acceptable Quality of Service (QoS) as the network scales.

- **Data Security and Integrity:** Secure data transfer in challenging, disconnected environments requires robust encryption and authentication, with keys and credentials accessible to multiple CNs without dependence on centralized key management.

- **Autonomous Operation:** Mobile CNs, potentially operating without human supervision, must handle unpredictable movement, variable LoS conditions, and evolving mission objectives. Incorporating AI-driven strategies for path planning, resource allocation, and fault detection can enhance autonomous resilience.

- Georeferencing the data obtained by the SGs can enhance autonomous navigation strategies to points of interest.

- It is essential to find a responsible balance with the use of the ISM band, avoiding traffic congestion, while using high SF values when the mobile agent is moving through areas that are difficult to explore. One line of research in this direction would be the study of coverage maps in ROUD scenarios by SGs operating with adaptative data rates.

3.5 Discussion and conclusions

Gathering information from a ROUD environment is essential for first responders, who must arrive at the scene fully aware of the challenges they will face: the surface area to cover, hidden zones (e.g., due to vegetation, debris, or indoor areas), cellular network coverage, GNSS reception quality, and environmental data (e.g., toxic gases, temperature, radiation). Additionally, once human resources are deployed in the operational area, it is crucial to monitor their status (e.g., identification, biometric parameters, position, task, current surroundings) and to support them in searching for and detecting potential victims, who may carry wireless devices transmitting signals in the environment.

However, robotic agents such as UGVs and UAVs are typically deployed before human responders arrive and later continue to cooperate in the operational scenario. It is vital for these mobile platforms to explore the area to establish a communication infrastructure that the SAR team can subsequently use, both on the ground and at the command and control center (CCCA).

This chapter has demonstrated how to co-design, implement, and deploy low-power, long-range networks that complement each other's properties to support SAR agents, for long period of times, in ROUD scenarios, where access is challenging, mobility is required, and communication infrastructure is scarce.

Different PAN and LPWAN networks and topologies are proposed to assemble different H-WSNs to support environment sensing and victim detection in ROUD scenarios, where mobile agents can wear nodes acting as mobile data sources and sinks.

In these isolated scenarios, it is assumed that Internet access is limited or non-existent. A first contribution in this regard is the use of star-connected Zigbee networks (several ZEDs connected to a ZC), where both the ZC and the ZEDs can be mobile and store information in their internal memories. Information travels through the mobile CNs, with mobile robots acting as data MULEs, collecting information from isolated ZEDs located far from the initially deployed coverage area. These mobile ZCs, along with others that can be static in pre-deployed zones, are interconnected to a WLAN established at the CCCA, which may have Internet access. In this way, the local databases of the mobile CNs synchronize with the database hosted at the FCC, which is remotely accessible and whose information can be monitored via a SCADA system.

The described Zigbee-based system was successfully validated in a realistic scenario. Operating the network without ZRs allowed for a thorough evaluation of Zigbee technology's performance in ROUD scenarios, particularly under the influence of moving nodes. This process involved designing and pre-determining the feasibility of using a PAN technology to construct H-WSNs that support SAR operations. The mobile ZCs can return to areas with

cellular network or WiFi coverage to offload its data, effectively extending coverage and acting as a data MULE capable of expanding the network without relays.

However, the main drawbacks for outdoor SAR operations are still penetrability and range. Thereby, based on the contributions with Zigbee, ZEDs are only recommended to cover events at medium distance from SAR agents, i.e. half a kilometer of radius maximum, and only if the LoS is partially good. In addition, we conclude that Zigbee can be helpful but it is not decisive for the detection of semi-buried victims, as tests indicate that the ZC, in charge of discovering the lost nodes, needs to be too close (at least 20 meters) to perceive its presence.

Nevertheless, the proposed Zigbee-based H-WSN enables the creation of Zigbee coverage zones, with nodes initially distributed in a star pattern. Coverage maps can be established, and ZEDs in intermediate positions could later be configured as ZRs remotely via the mobile ZCs when they approach and receive the data. This approach allows the network to dynamically adapt its topology, transitioning between configurations to meet the evolving needs of different phases in the emergency response.

Other PAN technologies can be helpful in other directions, such as detection (BLE) or localization (UWB). In this thesis, a BLE-based sensory system has been proposed to detect potential victims (PVs). Experimentation showed that all the deployed BLE devices were detected by the robotic platforms (UAV and UGV) as they moved through the application scenario, even those that were under the cover of dense dust, light debris or vegetation. Such nodes would not be visible from the aerial view of the area and may not be sufficiently conspicuous to promote the search by other more exhaustive methods. In fact, the UAV and UGV systems complement each other, the UGV can provide more detections in order to estimate the distance to the victim, while the UAV allows faster preliminary terrain reconnaissance. This allows covering wider areas, being able to detect even the semi-buried devices. For instance, the BH node located inside the tunnel was only detected by the UAV (Figure 3.9), since its height improved the LoS towards the device, thus demonstrating the complementary effect of a dual ground and airborne approach.

The BLE nodes were successfully detected even in conditions where dense dust, light debris, or vegetation obscured visibility from the air, making traditional search methods less effective. The UAV's BLE scanner, with a direct LoS to the ground, demonstrated higher sensitivity to transmissions compared to the UGV's scanner, which is more affected by interference and obstacles.

However, the UAV's speed proved critical for effective detection and synchronization. It must be moderate—neither too fast, to allow adequate scanning of BLE signals given

their periodicity, nor too slow, to avoid battery limitations requiring frequent landings for replacement.

In addition, the system does not rely on specific applications or hardware for detection, requiring only compatibility and activation of BLE versions, thus broadening the range of detectable devices.

However, an architecture capable of synchronizing detections made by each robot in real-time is required. Synchronization currently performed at the CCC could be implemented at the edge, integrating detections locally into ROS on each robotic platform. This approach would avoid MySQL queries, which introduce latencies into the system. Furthermore, the data collected locally could be processed to enable the robot to operate autonomously and reactively. Simultaneously, when a robot has Internet connectivity, it should be able to send its accumulated data to the CCC for analysis.

On the other hand, an alternative solution is offered with LoRa, an LPWAN technology that enables the coverage of large areas (on the order of kilometers) while transmitting data packets at regular intervals for extended periods of autonomy. The contribution in this regard is the ability to extract information from a catastrophic scenario using infrastructures located tens of kilometers away, where Internet connectivity is available, thereby leveraging remote resources in areas that are otherwise inaccessible due to a lack of Internet access. This has been achieved by modifying the LoRaWAN protocol at the node activation level, using the ABP mode, so that any CN in the vicinity of the deployed EDs can receive environmental information, regardless of whether it has Internet access.

The deployed LoRa H-WSN forms a *star of stars* configuration, unlike the Zigbee network, which is a set of independent stars that later aggregate the collected information. As a result, the Zigbee network provides information more slowly, whereas the LoRa network can deliver soft real-time information. In both cases, georeferenced sensory data are processed, enabling real-time tracking of agents carrying sensor nodes. A specific case was analyzed involving a SAR dog carrying Zigbee sensor nodes, demonstrating how the proposed offline/online configuration allows data to be obtained both in areas without coverage and after coverage is lost and subsequently regained.

In all cases presented, the concentrator nodes (ZCs in case of Zigbee, BLE scanners in case of Bluetooth, and gateways in the case of LoRa) must have access to a WLAN network, so that they can transmit the information to remote control centers.

WiFi (in 2.4 and 5 GHz bands) is an omnipresent wireless technology in all onboard robotic systems. The importance of including WiFi-based mesh networks (IEEE 802.11s) has been validated in order to cover remote areas in ROUD scenarios. An implementation of mobile infraestructure has been presented: a mesh network composed of four WiFi mesh

routers, that were Multi-User Multiple-Input Multiple-Output (MU-MIMO) nodes (two were static, and two were mobile). Both static relay nodes had to be placed at the south entrance to guarantee LoS into a dual-tunnel.

The mobile mesh nodes were placed on two UGVs: Rambler (repeater-UGV) and Rover J8 (scout-UGV). Both UGVs are integrated as robotic agents in a management platform for distributed robotic systems in emergency missions, including path planning [162]. In addition, two human agents (from now on, HA1 and HA2) operated two smartphones (UE1 and UE2, respectively) while moving through the dual-tunnel. They streamed video to the FCC where images were monitored in real-time, and KPIs were processed and stored.

The experiment involved testing a WiFi mesh network against a commercial 5G in 184-meter-long shallow buried tunnels of concrete. Two human agents (HAs) streamed video to the Internet through the mesh. The configuration of the presented mesh is novel as the extension of the network is done with two uncrewed ground vehicles (UGVs) with different roles: a repeater-UGV, with good coverage, which carries the mesh master node, and a scout-UGV, which explores the area until it reaches the tunnel, providing coverage to the users inside the tunnel. Thus, the network is deployed from an outdoor area to an indoor area, thanks to the mobile relay node on-board the scout-UGV, getting the repeater-UGV to maintain LoS with the scout-UGV.

Based on the results, the following lessons have been drawn: i) the designed setup proves to be effective in enhancing the KPIs of the static mesh network, providing coverage that allows smooth video streaming from two HAs in disaster-stricken areas where the 5G network is not operational; ii) Video transmission at a resolution of 640×480 proceeded without interruptions to a remote control station, with the mesh network consistently maintaining a latency of less than 200 ms throughout the dual-tunnel scenario; iii) The WiFi mesh network opens up possibilities for future remote tele-operation of the scout-UGV, enabling real-time video visualization from its onboard cameras; iv) H.264 was ultimately used for video streaming, offering significantly better compression efficiency compared to MJPEG or MPEG-4, thus enabling stable transmission at lower bandwidth without overwhelming the mesh network, while maintaining broad device compatibility and acceptable computational requirements.

Beyond the valuable role that ISM technologies such as BLE, Zigbee, LoRa, and WiFi play in enabling rapid, flexible, and energy-efficient information gathering in ROUD scenarios, it becomes necessary to complement them with more resilient infrastructures based on licensed and military frequency bands. ISM solutions are essential for deploying local sensing networks and extracting initial data even without existing infrastructure. However, in large-scale or congested disaster areas, licensed sub-1 GHz bands (e.g., VHF, UHF, LTE/5G

PPDR around 700–800 MHz) provide better coverage, improved propagation, and lower interference. Additionally, military/public protection bands and satellite links enhance robustness, security, and scalability. Future work should hybridize ISM-based H-WSNs with licensed and reserved-spectrum technologies, creating multi-band, multi-mode networks that adapt to mission requirements and environmental conditions.

In this context, each group of first responders — firefighters, police, military, and civil protection — should carry a standardized, ruggedized device integrating multiple communication technologies (both ISM and licensed). This would eliminate the need for multiple radios while enabling seamless inter-agency communication. A smartphone with external sensors could serve as a unified platform, supporting local protocols like WiFi and Bluetooth, and long-range standards such as LoRa, LTE/5G, and public safety systems like TETRA — a proven solution for secure, group-oriented voice and data transmission over licensed spectrum (typically 380–430 MHz). Integrating TETRA, DMR, LTE/5G PPDR, and dynamic spectrum technologies into a multi-frequency platform would standardize communication and improve operational effectiveness in critical scenarios.

Finally, a distributed architecture is required to share information between edge devices and cloud servers, allowing remote application deployment and location-based agent activation.

Chapter 4

Fostering Cooperation between Agents through MEC and Mobile Devices

4.1 Introduction

The previous chapter addressed the perception of ROUD environments by different mobile agents acting as mobile nodes of various hybrid networks based on low-power and low-data-rate technologies. The goal was to extract information from scenarios that do not have a fixed or stable communication infrastructure, enabling data distribution through different hosts (FCC and robots' PCs) without requiring high bandwidth or hard real-time performance. For this purpose, the MQTT protocol, designed for lightweight data transmission, was used.

However, in ROUD scenarios, it is also necessary to communicate heavier data types, requiring the integration of high-bandwidth sensors with low-latency needs, which calls for co-designing communication and processing systems capable of distributing the information the agents perceive. The challenge lies in achieving this while reducing latencies and ensuring efficient responses from the robots deployed in ROUD scenarios. How can all the gathered information be securely distributed to both mobile robots and human operators in remote locations, and how can this be applied to SAR use cases?

This chapter presents the second contribution of this thesis: the design and implementation of an Edge-Cloud architecture for integrating a diverse Internet of Robotic Things (IoRT). IoRT platforms are equipped with various sensors, ranging from high-throughput devices such as cameras and LiDARs to low-rate sensors capable of transmitting lightweight data, such as environmental information. Additionally, all data must be linked to the position provided by GNSS RTK modules on board the IoRT platforms. The system incorporates heterogeneous H-WSNs and 5G sensors, optimizing low-latency and adaptive bandwidth

requirements through network slicing. Key performance indicators (KPIs), such as latency, throughput, and quality of service, are regulated to ensure efficient distributed processing in resource-constrained scenarios.

The rest of the chapter is organized as follows: Section 4.2 introduces the Internet of Cooperative Agents (IoCA) architecture. Section 4.3 explains an implementation of the architecture in SAR missions, validated in realistic ROUD environments, including the details in Subsection 4.3.1. Subsections 4.3.2 and 4.3.3 present two use cases: remote SLAM via network slicing and a positioning system for potential victims (PV). Finally, Section 4.4 summarizes the contributions and presents the conclusions.

4.2 The Internet of Cooperative Agents architecture

The robots' onboard devices form a LAN managed by a mobile router. This router, provided by the Internet Service Provider (ISP) serves as the interface between the ISP's network and the robot's internal LAN. The router provides cable and WiFi connectivity and enables WAN access. Furthermore, routers can employ bonding, combining cellular connectivity from multiple ISPs to optimize the robot's coverage.

Co-designing architectures across sensing, communication, and distributed processing levels is critical to supporting real-time decision-making within the strict response times required for SAR operations. Key Performance Indicators (KPIs) such as latency, throughput, packet loss, coverage, quality of service, and processing load (both near and distant from data sources) must be carefully regulated. They can be measured in the ISP network, but also using the routers that interconnect all points in the IoRT.

This section introduces the development and implementation of the Internet of Cooperative Agents (IoCA) architecture [163], which addresses how the collected information can be securely distributed in real time to mobile robots and remote human operators. It examines the design, deployment, and continuous improvement of an Edge-Cloud architecture that integrates a diverse IoRT, including heterogeneous WSNs and ad-hoc 5G sensors with low latency and adaptive bandwidth requirements via network slicing. The proposed IoCA architecture consists of a sensory and communication framework for field applications requiring cooperation between human and robot teams. Additionally, this research explores the integration of smartphones in robotics, analyzing their benefits and limitations.

The architecture is based on two fundamental entities with distributed computational resources: an IoRT, which is the source of sensor data, and a Feedback Information System (FIS) for data processing, monitoring and storing.

The IoRT integrates hybrid and heterogeneous wireless sensor and communication (i.e., transceivers) networks (H2WTN), including PAN, WLAN, and LPWAN technologies (see Chapter 3), and a ROS-based 5G sensor network. This IoRT is connected to the FIS through MEC centers that are located between the edge and the cloud, namely the fog, where network slicing is included to offer enhanced mobile broadband (eMBB) services, according to the agents' location.

Location awareness [13] can be considered a subset of situational awareness, which also considers the identification of objects, the dynamics of the environment, and the interpretation of how these factors interact and may change over time (covered in Chapter 5). The challenge lies in achieving an situational awareness that accumulates the comprehension of the agents [29], enabling them to accomplish their tasks more efficiently.

As an analogy, in photography, a camera is often said to be *in focus* or *out of focus*. However, *focus* is something the camera *has* when its lens is correctly aligned with the subject, enabling it to *perceive* the subject *more* or *less* clearly. In this sense, focus can be seen as an internal (though *temporary*) property of the camera (something it acquires under correct optical conditions), allowing it to perceive the subject more clearly.

Similarly, in Fog Robotics, an agent (usually moving through its operational environment) can be said to *have more* or *less* location awareness. This location awareness reflects how well the agent can *perceive* that it has the necessary capacity to process information from a particular Point of Interest (PoI) in the application scenario (i.e., a data source) and act on it efficiently. Just as a camera's focus improves with proper alignment, an agent's location awareness improves with closer proximity to the PoI and better communication and processing resources. When resources are equivalent, the agent that is physically closer to an event *has more* location awareness (*focus*) relative to other agents. In this sense, location awareness can be seen as an internal (though *temporary*) property of the agent (something it acquires under correct communication and processing conditions), allowing it to perceive the operational environment more clearly.

This concept emphasizes that location awareness is a *temporary* characteristic of the agent rather than a collective product of multiple agents. By framing location awareness as something the agent *has*, it becomes distinct from situational awareness, which emerges from cooperation among various agents.

The IoCA architecture lays the foundations for a framework in which agents work together to complete tasks. These agents not only move toward points of interest to explore specific locations but also cooperate based on factors such as their poses, computational and communication capabilities, or whether they are robotic or not, giving rise to a group of cooperative agents with shared situational awareness.

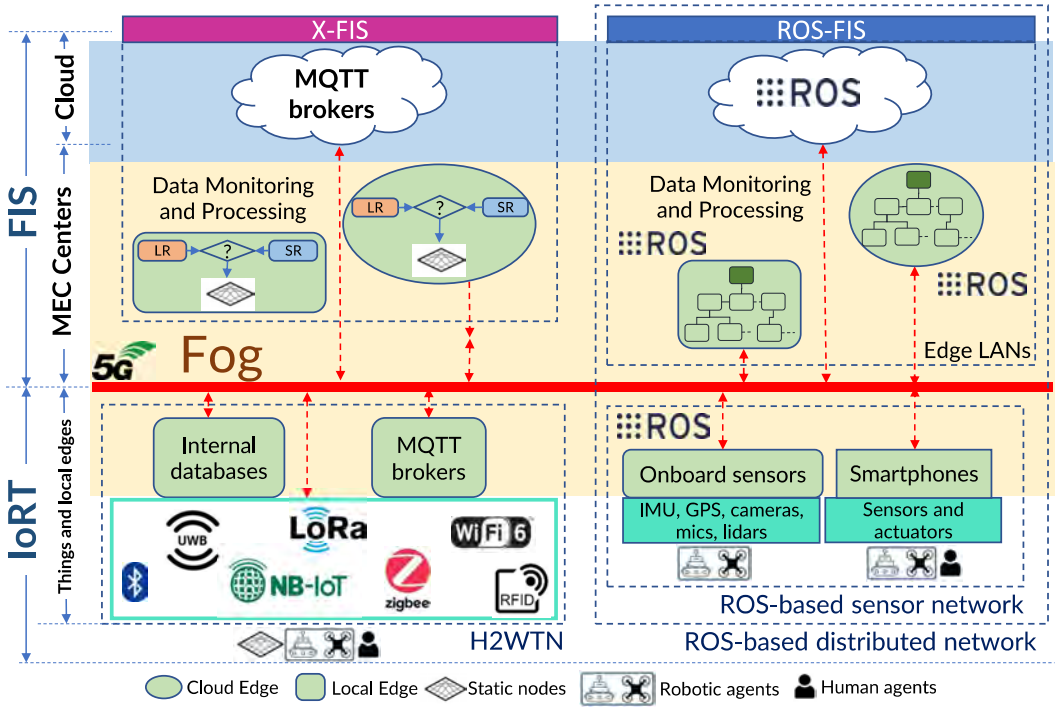


Figure 4.1 The Internet of Cooperative Agents architecture (X-IoCA).

Figure 4.1 shows the computational fog, which includes all elements except EDs (i.e., sensors and actuators) and the cloud. Thus, the elements in the fog consist of Local-Edge devices (depicted as green rounded rectangles) and Cloud-Edge devices (green ellipsoids), as well as switches and routers to route data traffic between their respective PCs or hosts (depicted as incoming red arrowheads in Figure 4.1). The thick red line represents the 5G communication system, which, at the time of this thesis, is the most advanced generation of mobile communications.

Local- and Cloud-Edge devices comprise computing devices used within the IoRT or the FIS, respectively, to operate services. Moreover, for communication, they have one or more User Equipent (UE) devices (e.g., a 5G router or a 5G smartphone). Local-Edge devices correspond to local hosts in the agents' operational environments, while Cloud-Edge devices are remote MEC centers. The MEC centers inside the FIS include two elements: X-FIS and ROS-FIS, each dedicated to a specific type of information flow coming from the two parts of the IoRT: the H2WTN and the ROS-based sensor network, respectively.

Using these data, Local-Edge devices act as an Infrastructure as a Service (IaaS), providing and receiving services at the edge, operating close to the operational environments and connecting to robots via WLAN/LAN networks. Similarly, Cloud-Edge devices enable

broader reach and integration with Internet-connected resources, acting as a Platform as a Service (PaaS).

Not all elements of the IoRT need to be physically located within the operational environments where data originate. Some agents and Local-Edge devices (e.g., CNs) may reside in remote areas, which are still considered part of the broader application scenario where data acquisition occurs. In the SAR context, the application scenario comprises the three different operational environments where an agent may operate: wide-are expanses, local areas near structures, and interiors of buildings [134]. Therefore, cooperative agents in any operational environment must remain connected through the Internet-enabled *things* in their payloads, which are linked to the FIS, i.e., the IoRT platform.

A given agent can use a resource that is more or less distant from it, which requires complying or not with a series of requirements, such as latency, bandwidth, storage capacity, fault tolerance, real-time capabilities, and response time to attend an event [114]. The IoCA architecture aims to promote agents' cooperation by sharing knowledge of the available assets at any given time. All cooperative agents operate in the same network, where they have access permissions to each other's resources. This allows a robot to rely on the onboard processing of another agent (which could be non-robotic) to run algorithms working on its data.

Finally, the X-IoCA architecture is implemented, where *X* denotes the system application. This chapter addresses SAR-IoCA, an implementation of the IoCA architecture for SAR missions, where shared management of edge and cloud resources can help improve emergency response times. In the SAR context, agents can be humans, different types of mobile robots and vehicles (e.g., UGVs, humanoids, quadrupeds, UAVs, or simply ambulances), and even sensorized dogs that carry EDs. The SAR-FIS must integrate all the information about the SAR scenarios to support robots to take actions through situational awareness (see Chapter 5), offering a smart link between perception and low-latency and high-bandwidth robotic services.

4.2.1 The Internet of Robotic Things

The IoRT is the source of information divided into two types of networks: i) a heterogeneous H-WSN for low-power EDs, including a WLAN for communications, giving rise to the H2WTN, and ii) a ROS-based distributed network for more significant bandwidth EDs. Therefore, the agents carry part of the IoRT, and the rest of the architecture supports them cooperating, resulting in the IoCA.

The Local-Edge devices are dedicated servers, including MQTT brokers, internal DB, and processing units, that must be available on the agents and in remote places. The information they produce must be available in the FIS.

4.2.1.1 The H2WTN for searching and providing coverage

The H2WTN comprises low-power networks, such as those presented in chapter 3, i.e., PAN, WLAN, and LPWAN networks. Therefore, LoRa, Zigbee, BLE, IEEE 802.11s, and other networks (UWB, Sigfox, etc) can be introduced to cover different applications. This H2WTN can perceive the environment without relying on sophisticated infrastructures and operate without the Internet when WLAN or WAN coverage is unavailable.

The H2WTN can detect sensor events, such as environmental issues at short (BLE, UWB, Zigbee, etc) or long distances (LoRa, Sigfox, NB-IoT, etc). This short-range (SR) and long-range (LR) information can be used by the motion planners at the FIS level (see Figure 4.1). After generating a trajectory to be followed by a given mobile robotic agent, it is ultimately up to the mission operators to decide whether the robot will be teleoperated or will move autonomously to the PoI, i.e., human operators choose strategies for robot motion control.

The H2WTN's EDs serve to detect and locate *things*, such as agents, while transmitting environmental information about their surroundings. The challenge is to connect all the data to the FIS through the Internet.

It is proposed to provide Internet to all concentrator nodes (whatever the technology) so that, if they have the network coverage, information can be transmitted and brought together in the same interface (in Chapter 3, each technology had its interface). MQTT brokers distributed among the Local-Edge devices will be used to process all the information in the FIS. This information flow is separated from the ROS-based distributed network and is managed by the X-FIS.

4.2.1.2 Development of ad-hoc 5G sensors

The goal is to extend the functionalities of the robotic and human agents by increasing their location awareness through 5G ad hoc sensors. Cooperative agents must embark similar sets of EDs, including IMU and GNSS modules, cameras, microphones, and LiDAR systems, among others. This payload is integrated into a ROS-based distributed network to obtain real-time information from the agents' environments in the FIS. However, on certain occasions, ad hoc devices are required to provide processing, communications, storage, and sensing while ensuring low weight and ease of placement. Using a smartphone as a payload for any agent boosts new applications in field robotics and within Edge-Cloud architectures, since

the smartphone can be used as an IoRT device. The challenge is to design and integrate it as a ROS node.

Smartphones serve as a two-way link for the agent, both with the FIS and with the operational environment, to inform and to request actions. Thus, a smartphone becomes an interesting IoRT device. The efforts have focused on transmitting, in real time, internal and external sensory information from the smartphone, trying to take advantage of its intrinsic sensing, processing, and communication properties. Several Android apps have been developed and shared with the community for a wide range of APIs, including ROS nodes to integrate smartphones in the ROS-based part of the IoRT.

The IoCA architecture must order the outgoing information of all the agents, by means of a ROS namespace specific to each agent, which helps to organize the agents from the FIS, where the location awareness of each agent is known. The goal is to use the ROS-FIS to modularize the agents' software, including that of smartphones.

ROS 1 integration in Android

Several ROS 1 smartphone applications have been designed and tested so that the smartphone can communicate with other ROS nodes distributed through the architecture. It is necessary to know the public/virtual socket (IP address and port) on which the ROS master node is hosted. Moreover, other smartphones can also act as Local-Edge devices, hosting subscriber ROS nodes. Local-Edge devices process and monitor the published information (especially, the audiovisual one) within the IoRT.

UMA-ROS-Android [164] is an open-source application developed to transmit information about the internal sensors in the smartphone. Figure 4.2 shows the provided data, including images from the cameras, pose from the internal IMU and GNSS modules, and audio from the smartphone's microphone. In addition, the application allows the agent's audio to be stored in the smartphone's internal memory.

ROS 2 integration in Android

Regarding ROS 2 applications, smartphones only need to have connectivity to the rest of ROS hosts through a VPN and share the same value of the *ROS_DOMAIN_ID* environment variable, whose default value in ROS systems is zero. This applies to Linux-based operating systems such as Ubuntu and Android. Modifying the *ROS_DOMAIN_ID* environment variable persistently on Android is not viable because it requires root access, which compromises security, limits app compatibility, and violates Google Play Store policies. Keeping the

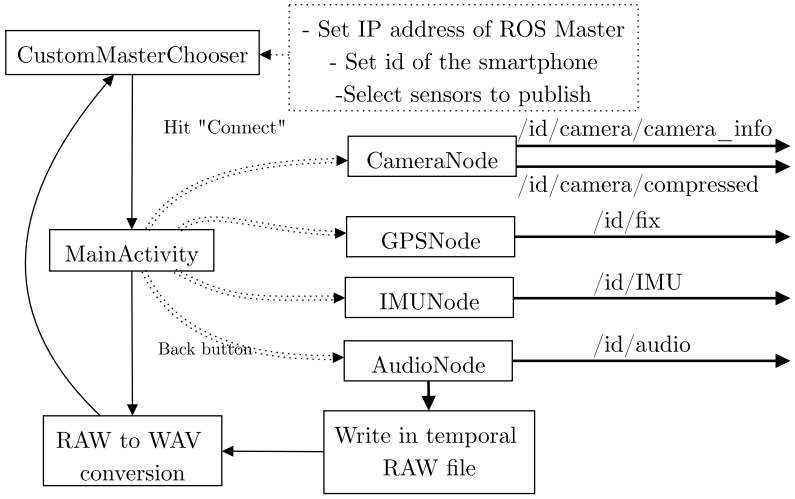


Figure 4.2 UMA-ROS-Android communication through ROS 1 nodes

default value (zero) requires all other hosts to set the same value for proper Data Distribution Service (DDS) communication; otherwise, all devices must use a matching domain ID.

The UMA-ROS2-Android (UR2A) application [165] can also transmit the sensory information of the smartphone, such as its position and orientation in space, or its percentage of available battery. In addition, a remote operator can send commands to the smartphone to switch among the different smartphone cameras and resolutions. It is available in a public repository: https://github.com/Robotics-Mechatronics-UMA/UMA-ROS2-Android/. Figure 4.3 shows the communication between the different ROS nodes through topics.

External devices can be connected and powered via a Type-C cable using the smartphone's battery, enabling the development of ad-hoc sensors that leverage the smartphone's processing, storage, communication capabilities, and power supply.

For example, equipping a UAV with an external GNSS module connected via Type-C to a smartphone and capable of receiving RTK corrections is valuable for ensuring data redundancy and achieving a higher data rate, allowing the agent's position to be determined more quickly.

This approach can also be extended to other agents, such as SAR dogs or human operators. By using the external module, the agent's position can be calculated and transmitted with significantly greater precision than what the smartphone's internal module could provide, with the data being published through the ROS-Android applications.

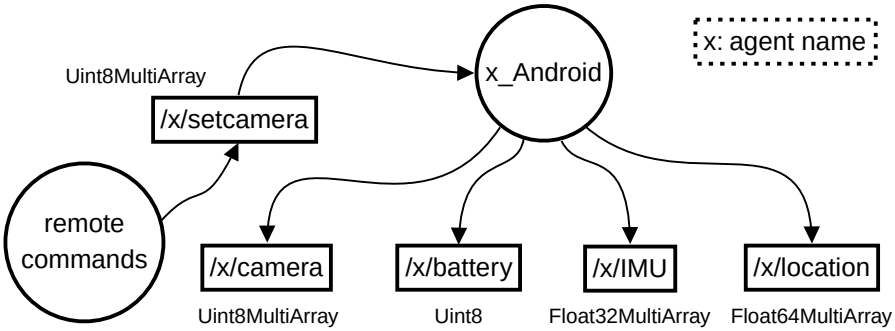


Figure 4.3 UR2A communication through ROS 2 nodes

4.2.2 The Feedback Information System

The FIS is distributed between MEC centers and the cloud, running specific algorithms on virtual machines with ROS. Conceived as a supervision and control system, the FIS is operated by both automated algorithms and human operators. It is divided into two components for data monitoring and processing: X-FIS and ROS-FIS.

This work extends the concept of location awareness. Beyond considering the location of a cooperative agent, it also factors in the computing edges's position and communication capacity, regardless of whether Local- and Cloud-Edge devices reside in the FIS, on an agent within the application scenario, in a remote location within the operational environment (e.g. a remote CN), or in the cloud. A particular Local-Edge does not inherently have less location awareness just because it is part of an MEC center. While Local-Edge devices in the operational environment may benefit from a favorable geographic position, their processing or communication capabilities might be lower than Cloud-Edge devices in the FIS.

As a result, location awareness depends on communication quality, processing capacity, and geographic placement within the fog computing space. Typically, devices closer to the cloud offer more computational abstraction (through virtualization and containerization), higher throughput, and greater processing power, but may also experience higher latency, thus resulting in lower location awareness compared to devices directly connected to the robots' LANs on-site.

When multiple robotic agents share the same MEC, distinguishing their location awareness becomes essential. Considering factors such as location, locomotion method, and communication capabilities, the agent most capable of minimizing response time at a PoI will be assigned a higher location awareness, making it the preferred choice to manage the event.

The X-FIS is dedicated to get data from the non-ROS things of the IoRT in order to run control algorithms on the cooperative robots (e.g. UGVs and UAVs). The events detected by

the H2WTN can trigger alarms to be attended by the cooperative robots, which can be planned thanks to the integration of a global path planner hosted on the computing edge devices, bringing redundancy to the system at different levels of location awareness. The information from the H2WTN is received in the cloud where an MQTT broker can communicate the important data to the planner in order to make decisions for every agent.

ROS-FIS is made up of Local- and Cloud-Edge devices designed to process and monitor information published by ROS nodes on the IoRT platforms (ROS-based sensor network). Each Local- or Cloud-edge device in the ROS-FIS is associated with a main computer (MPC), as illustrated in Figure 4.1, where the MPC is depicted at the top of a hierarchical tree structure. The MPC connects to multiple secondary computers (SPCs) within the same local network. Collectively, these PCs form a single computing edge, which connects to the Internet via a router with a known public IP address, ideally static. This is particularly critical for ROS 1, where bypassing Network Address Translation (NAT) is necessary to enable communication through the LAN/WAN interfaces for each Local- or Cloud-Edge device.

In this setup, the host directory (`/etc/hosts`) of each PC (whether MPC or SPC) within any LAN (on both the IoRT and FIS sides) must include the public IP addresses of all other routers. Each WAN IP must be paired with the hostname of the MPC for the corresponding LAN downstream of each router. This configuration enables bidirectional communication between any two MPCs within the ROS network, spanning the IoRT and ROS-FIS domains. Over the course of this thesis, the architecture has progressively migrated to ROS 2. At a high level, depending on the version used, the network design must meet certain considerations.

- ROS 1: ROS-FIS is linked to the cloud where the *roscore* node runs and manages all the connections between the distributed ROS nodes. In this thesis, the Cloud operated in the 5G core of a pilot 5G network (i.e. non-commercial), in a virtual machine, which must know the public IP addresses of the MPCs chosen within the fog. However, it is recommended to host the *roscore* node in a MEC center, since the 5G core is usually managed by an ISP, making it more challenging to get port-forwarding permissions.

  The demilitarized zone (DMZ) must be enabled on the routers, with the local IP address of each MPC being the one assigned to have the DMZ. Thus, for example, through RVIZ (a toolkit for real domain data visualization [166]) it is possible to visualize the compressed images published from the robot cameras, or listening to the audio captured by their microphones in the MEC centers. There are other port-forwarding options [167], and it is also possible to set up a VPN between the hosts involved in the application. The final choice will depend on access and security requirements.

- ROS 2: ROS-FIS is linked to the cloud where soft real-time processing is run. Communications are more democratic than in ROS 1, being able to isolate certain devices even if they are in the same LAN, thanks to the DDS protocol, being able to group devices by using the same value for the *ROS_DOMAIN_ID* environment variable.

4.3 Implementation in SAR Missions: SAR-IoCA

4.3.1 System architecture

Returning to the context of this thesis, the SAR-IoCA implementation is presented for disaster scenarios, where SAR missions require real-time, high bandwidth, and low latencies [168].

SAR-IoCA aims to support SAR teams in emergency situations, providing relevant information to MEC centers, and enabling tasks for the different SAR agents, which can be human (first responders, firefighters, military units, and police), robotic (UGVs, UAVs, and mobile manipulators), or canine rescue team (dog and handler). For the sake of clarity, the following is a representative implementation of the SAR-IoCA architecture (based on ROS 1) during some realistic experiments carried out in a the LAENTIEC operational area where, without the loss of generality, the heterogeneous robotic team (see Figure 4.4) is composed of three UGVs, one UAV, and one human agent, i.e. a foot soldier. The elements presented respect the shapes and colors adopted in the X-IoCA architecture (Figure 4.1).

The MEC centers are made up of a Forward Control Center (FCC) and a Base Control Center (BCC). The FCC acts as a Local-Edge within the FIS due to it is installed within the application scenario, while the BCC acts as a Cloud-Edge, since it has less location awareness than the FCC and agents, as it is in a MEC located in a remote laboratory, and has higher processing capabilities but a worse communication infrastructure. Moreover, the BCC includes a replica of the SAR-FIS in FCC to provide redundancy.

In addition, Figure 4.4 illustrates the five Local-Edge devices distributed across the SAR scenario. These devices include embedded computers (such as PCs or smartphones) carried by SAR agents, comprising three UGVs, one UAV, and one military personnel. The human agent carried a 5G smartphone with which he could communicate data bidirectionally with the MEC centers (FCC and BCC) and robots using ROS nodes integrated in his smartphone (via ROS-Android applications). Moreover, the embedded computational hosts on board the UGVs and the UAV were used for data processing. All these robotic agents have, at least, a GNSS receiver with RTK differential corrections. In addition to smartphones, other equipment on board robots, such as different types of cameras, LiDARs, or GNSS modules,

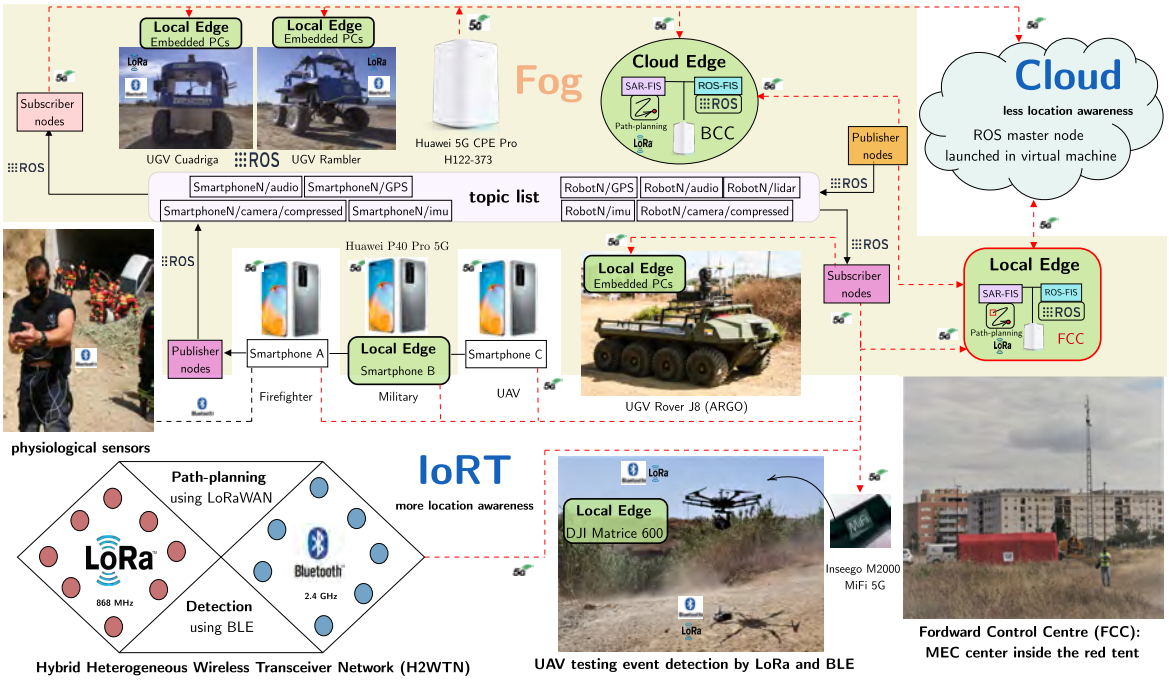


Figure 4.4 SAR-IoCA architecture (a ROS 1 implementation)

are used to communicate other sensory information to the ROS-FIS, dedicated to create datasets [169].

The H2WTN's EDs serve to detect and locate *agents* and *victims*, while transmitting environmental information about their surroundings. The challenge is to connect all the data to the SAR-FIS through the Internet.

The H2WTN, composed of static and mobile nodes based on LoRa, Zigbee, and BLE sensors, provides both MEC centers (FCC and BCC) with relevant information about the surroundings of the SAR agents (see Chapter 3).

In particular, the LoRa H-WSN (Figure 4.5) provides the MEC centers with information for identifying risks in the area of operations. For example, the internal accelerometer of an ED could detect an earthquake; temperature and humidity sensors could detect a fire; rain gauges could detect possible floods; or noise and gases sensors could detect risks related to terrorist attacks. To respond to this diversity of scenarios, event activation has been implemented through the use of these LoRa EDs (see Chapter 5), which are grouped together, forming sensory groups (SG), in different locations in the terrain, either static or carried by a mobile agent (rescue dog, UGV or UAV [64]). The information captured by the H2WTN is managed at the MEC centers. The difference with what was presented in Chapter 3 is that now all the deployed transceiver networks are monitored by the same tool (from now on the

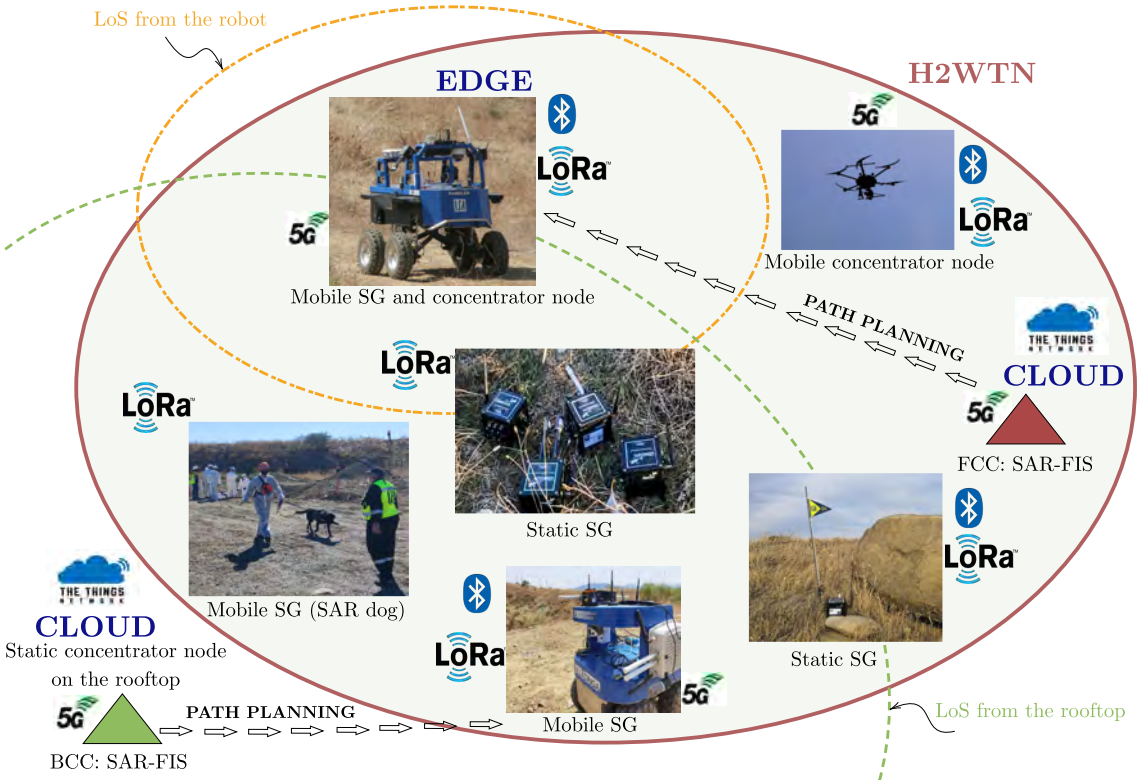


Figure 4.5 H2WTN scheme for the SAR experiments

SARFIS tool) that is also available in more than one MEC center, being able to receive and command information from different points inside and outside the application scenario.

The *SARFIS tool* is an ad hoc MATLAB application designed and implemented by the UMA Robotics and Mechatronics Group for the global planning of UGV paths over natural terrain and the monitoring of sensory devices deployed over the application scenario and distributed both on the operational environments and in remote areas to it. The IoRT integrated into *SARFIS tool* includes IP cameras, LoRa and BLE networks, IEEE 802.11mc devices, and other high-bandwidth sensors within the ROS network. This software has been gradually updated by the UMA Robotics and Mechatronics Group to the SAR-IoCA architecture requirements throughout the developments of this thesis.

The *SARFIS tool* distinguishes four basic elements: (i) agents: UGVs with the ability to move through the environment and to embark elements of the sensor network; (ii) sensory targets: static LoRa-based SGs; (iii) points of interest: locations in the environment that create areas of interest for path planning; and (iv) environment: area of operations, i.e. the application scenario, defined using a DEM and an orthophoto (zenith aerial photograph), where the multi-sensor network and agents are deployed.

The *SARFIS tool* integrates a strategic global planner, i.e. a global path planning system for a set of UGVs towards a set of target points with different priorities. The tool generates the sequence of waypoints in the environment that an agent must travel from its current position to reach a target point.

Each agent publishes the geolocation coming from the onboard GNSS receiver through MQTT topics to let its current position be known. To do this, each agent hosts an MQTT client with the ability to connect to the MQTT broker hosted in the SAR-FIS Cloud-Edge. In case of no Internet access, the agents can make use of the MQTT broker hosted on the LoRa CNs deployed in the application scenario, to which they have access thanks to the WLAN deployment (explained in chapter 3).

The sensory targets and PoIs constitute the possible target points of the path planner. The path-planning process can be enabled by a human operator or triggered by a sensory event (see chapter 5). The environment map shows an aerial orthophoto of the area of operations where symbols are displayed to locate each agent, sensory target, and point of interest defined in the *SARFIS tool*.

Finally, the lowest level of location awareness is the cloud, where a computing center hosted in a virtual machine (at the core of the 5G pilot network) performs processing (cloud computing) and hosts the master node of the ROS 1 network, which is distributed along the fog.

Figure 4.6 shows some cooperative agents (see Appendix A) carrying the proposed IoRT integrated in the IoCA architecture, including smartphones, H2WTN nodes, LiDAR systems, cameras, microphones, and GNSS and IMU modules.

However, the SAR-IoCA implementation contains a complete FIS composed of two elements: SAR-FIS and ROS-FIS.

The architecture is intended to distribute the processing and make the information available in the application scenario in as many computing centers (MEC) as necessary without affecting the performance of the systems.

The SAR-FIS includes the *SARFIS tool*, but it is not limited to this software (see Figure 4.7-a). In this case, the *SARFIS tool* can be replicated at both MEC centers (BCC and FCC) and integrates the LoRa H-WSN and a global path planner to send UGVs to a new PoI created through some event detected, for instance, by the H2WTN (see chapter 5). Each Local- or Cloud-Edge device in the FIS can host the *SARFIS tool* and other dedicated computers performing other tasks, such as running algorithms, monitoring cameras, managing data from MQTT brokers, etc.

The ROS-FIS consists of dedicated PCs or hosts (physically located in the MEC centers) which monitor and process, in real time, the information published on topics from the hosts

(a) Cuadriga being tele-operated while carrying a LoRa SG and a BLE scanner

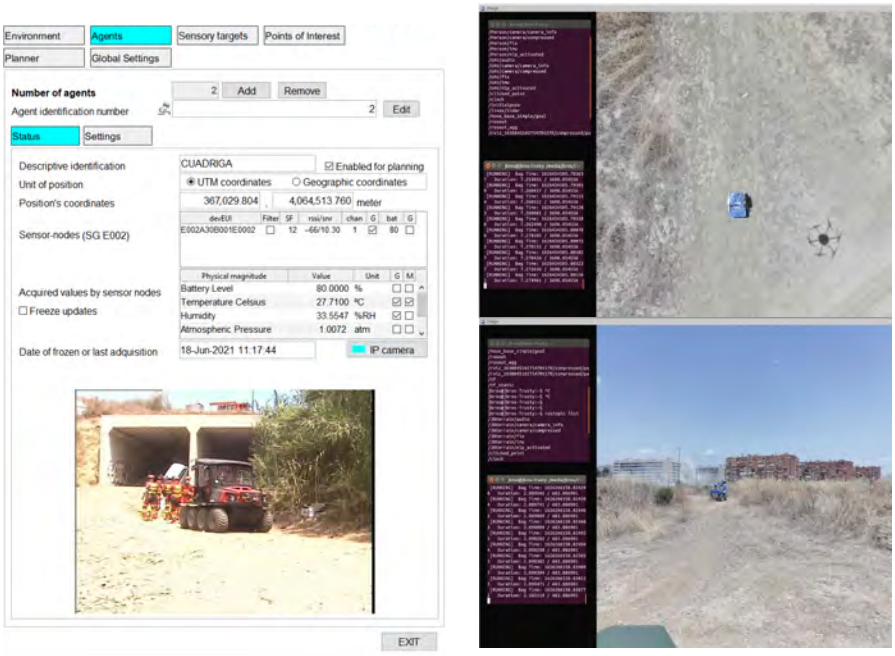
(b) Rambler close to a victim after performing a path-planning with a LoRa CN embarked

(c) Rover J8 in follow-me mode with LoRa SGs and a smartphone as part of its payload

(d) DJI Matrice 600 with a smartphone as a payload on its gimbal mechanism

Figure 4.6 Cooperative robots embarking H2WTN and ad-hoc 5G sensor nodes

on the IoRT platforms (see Figure 4.7-b), being interconnected and linked to the ROS master node, hosted in the cloud. In addition, the cloud host (a virtual machine directly connected to a Pilot 5G core) could perform processing tasks. Connections between remote hosts are made over the 5G network, prioritizing low latency and high throughput requirements. ROS-FIS separates the computational load on remote equipment (between the fog and the cloud), taking advantage of the 5G network functionalities, such as network slicing. In addition, ROS-FIS manages the flow of both outgoing and incoming data across devices on the 5G network. Each subscription to a ROS topic prompts a new uplink of that information at its source (i.e., the publishing nodes), as the request corresponds to a provided service. This setup makes it possible to control data traffic among the 5G EDs and even terminate ROS nodes on various agents if network performance degrades.



(a) (b)

Figure 4.7 Examples of camera images visualization in the MEC centers. (a) The *SARFIS tool* showing information from a LoRa SG onboard Cuadriga, whose IP camera is shown. (b) ROS-FIS showing 5G smartphones' cameras on UAV and Rover J8

4.3.2 Real-time perception through network slicing

4.3.2.1 Design and implementation

5G networks have the potential to enable the distribution and execution of remote algorithms, such as SLAM. However, as of the time of the IoCA architecture (2021)[163], no documented studies in the literature had explored the feasibility of relying on remote processing to execute such algorithms in real time. Even by late 2024, only a few proof-of-concept experiments with 5G and 6G have been conducted [170, 171].

Prior to the IoCA architecture, UGV on-board routers were constrained by asymmetrical 4G networks, limiting communication with the CCC to 100 Mbps in downlink and 20 Mbps in uplink under optimal outdoor conditions. These conditions relied on strong coverage and minimal network congestion, atypical in ROUD scenarios.

To address these limitations, network slicing has been integrated into the IoCA architecture using the pilot 5G network, ensuring that no other UEs were connected to the cell apart from the cooperative agents [172]. Specifically, this work presents a comparison between edge and cloud Computing for processing large volumes of data. In particular, three robotic platforms (one UAV and two UGVs) and two MEC centers (FCC and BCC) are used to

Figure 4.8 Rambler view from the UAV, which holds a smartphone.

validate the SAR-IoCA implementation with network slicing, sharing the available bandwidth between the FIS and the cooperative agents to perform various real-time tasks:

- To perform tessellation in the Cloud using aerial images captured by a UAV-mounted smartphone (see Figure 4.8), aiding in mapping prior to the deployment of the remaining cooperative agents.
- To conduct an exploration flight for victim detection, streaming 4K video to be analyzed in real time at the FCC.
- To deploy two scout UGVs, transmitting audiovisual information of their immediate surroundings while navigating, either autonomously or via teleoperation.
- To perform SLAM remotely via the non-commercial 5G network using a UGV, escorting a human team to an area of interest (see Figure 4.9).

4.3.2.2 Experimental results

Network slicing for bandwidth allocation was tested in a realistic exercise during JEMERG XXI, conducted at the LAENTIEC operational area on two separate occasions: 18 June 2021 and 30 September 2021. The objective was to evaluate the application of network slicing within a pilot 5G network.

To accurately measure the behavior of devices within each group, all devices associated with prioritized SIMs were integrated into a ROS 1 network as part of the IoCA architecture (see Figure 4.10).

Figure 4.9 Rover J8 and human agents responding to a request for evacuation

This setup allowed for end-to-end traffic control and was designed to manage large data transfers, such as high-resolution video and LiDAR data streams, between robots and MEC centers. The SIM cards available for the 5G pilot network were grouped into two sets. The first set was prioritized and could occupy up to 80% of the cell resources. The other set had less capacity associated with it (the remaining 20%). In other words, UEs operated in two virtual networks or slices (with different usage priorities) within the same 5G network infrastructure. All UEs were placed in the 20% slice except for the router onboard Rover J8, which was allocated 80% of the cell bandwidth.

To monitor the activity within each slice into which the network was divided, a ROS master node was hosted on a PC in each slice. Subnetwork A was governed by a ROS master node located in a remote virtual machine directly connected to the 5G core. Similarly, for the group of devices included in the 20% slice (subnetwork B), another ROS master was deployed, allowing internal monitoring of the bandwidth and transmission frequency of each host in both subnetworks. The implemented design was useful as a contrast to the traces recorded by the ISP. In addition, the throughput at the UE level was monitored for the router managing LAN1, which was associated with the BCC control center.

Regarding the IoRT, three smartphones (SP) and one 3D LiDAR were used to generate big throughputs. SP1 and SP2 were embarked on a DJI Matrice 600, and on Rover J8, respectively, publishing audiovisual information of the agents' environment through the UMA-ROS-Android application. In addition, a third smartphone (SP3) was used by a human agent to get information (through ROS subscribers) from the agents environment. On the other hand, Rover J8 also embarked a 3D LiDAR (Velodyne HDL-32) for SLAM.

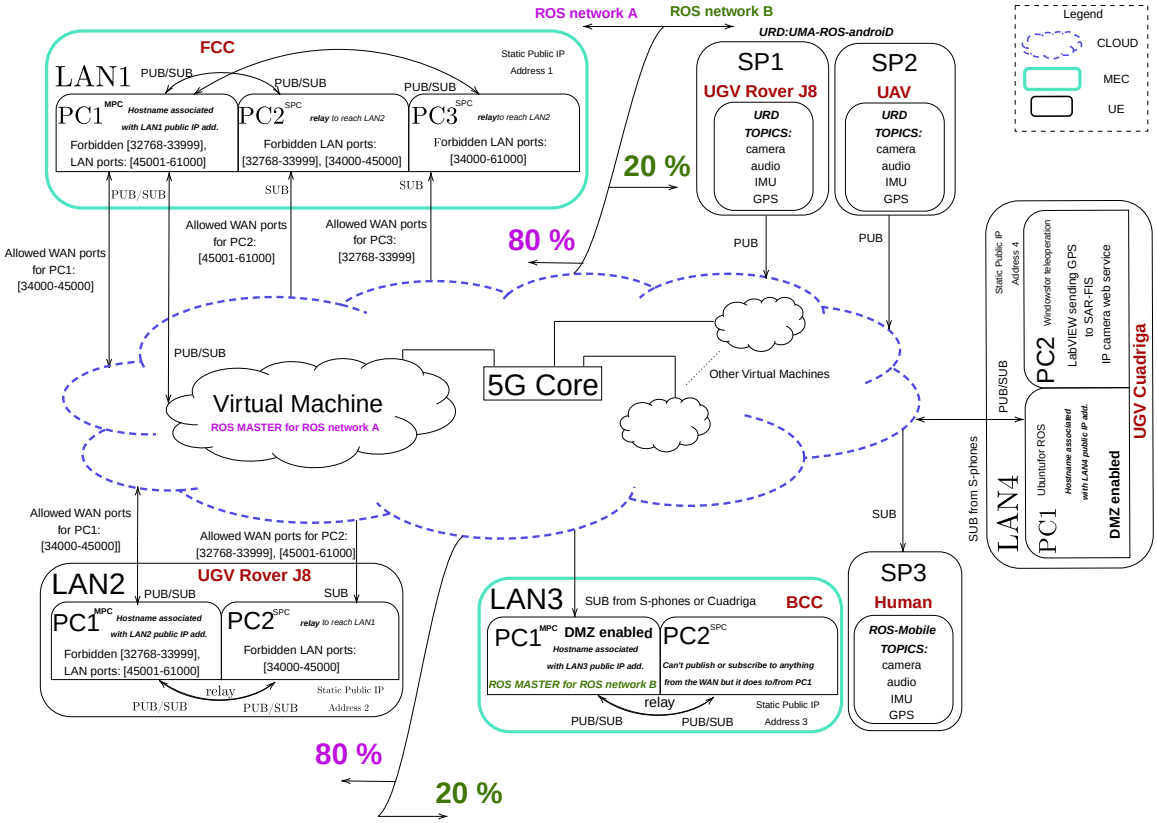


Figure 4.10 ROS 1 subnetworks for two slices in a non-commerical 5G Network

The experiments (see Tab. 4.1) aim to verify whether it is possible or not to rely on remote processing to execute SLAM in real time through 5G at the same time as other high data rate information is transmitted in the same ROS network. The generated map has to be available at the MEC centers.

Therefore, the challenge is to process in real time the raw data from the 3D LiDAR collected during the movement of Rover J8 in a ROUD scenario, with the computation falling on a virtual machine connected to the 5G core (Cloud topology), both for the construction of the point cloud and the SLAM itself.

The implemented algorithm is a Hierarchical Dynamic Layers (HDL) Graph SLAM [173]. Specifically, we use the hdl_graph_SLAM [1], an open source ROS package for 6 DOF SLAM which is valid for a real-time mapping of our experimental area. This algorithm was chosen since it is heavy for mid-range and low-end hardware, so cloud Computing is an interesting option. The algorithm joined the raw data coming from the GNSS module and the LiDAR on Rover J8.

[1] https://github.com/koide3/hdl_graph_slam

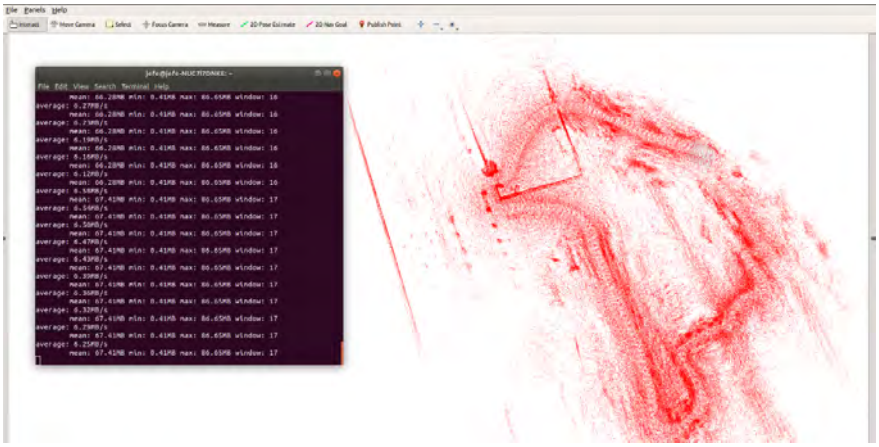


Figure 4.11 Throughput while creating the last map chunks: 6.58 MB/s

The path traveled is correctly mapped (see Figure 4.11), and the rectangular area where the Forward Command Posts are (including the FCC) is easily identified. The topic where the map chunks are published shows a data rate of 6.58 MB/s (≃53 Mbps), which is an affordable data upload for a 5G router connected to a non-commercial cell. This throughput would be impossible to achieve without 5G, and what is even more important, it wouldn't be possible to share the 5G network resources with other IoRT connected to the network without the network slicing applied to both ROS 1 subnetworks. As the map grows, the size of the message to be transmitted becomes larger. At this point, latency, which in 5G stand-alone is below 19 ms for WiFi connections, plays an important role, since with LTE or 4G networks this application would be unfeasible in real time. Network Slicing allows resources to be consumed fairly in another part of the network, i.e. without exceeding the limit allocated to each UE within a slice. In this way, it is possible for Rover J8's router SIM card to generate an even greater upload, streaming images with its on-board camera, or transmitting its GNSS position at high frequency.

4.3.2.3 Evaluation

Edge vs Cloud Robotics Focus

Raw data obtained by the 3D LiDAR (Velodyne HDL-32) during the movement of Rover J8 might be computed in a dedicated processor on board the robot edge or in a virtual machine in the cloud (provided by the ISP), depending on the chosen topology.

In the cloud version, which saves energy consumption and space on board the robot, it is possible to separate the map construction and transmission of the sensory data from the execution of the SLAM algorithm itself (hdl_graph_slam), between LAN2 and virtual

Table 4.1 Timeline of Events During the Experiment

Time	Event / Description
09:33	The planned sequence of experiments begins, starting with the display of Cuadriga's IP camera at the BCC. Uplink traffic is only generated when a client requests the visualization.
09:33–09:36	Cuadriga is teleoperated using video streaming. One subscription to the Rover J8 camera and one to the UAV camera are active. Before the Rover J8 moves, visualization tests are performed at the MEC to verify communication.
09:55	Teleoperation of Rover J8 begins.
09:56	The UAV takes off.
09:57	Four subscriptions are active (two from SP3 and two from FCC). Non-prioritized SIMs can consume 100% of the bandwidth if prioritized SIMs (80%) do not generate throughput. The difference between UL and DL is due to Cuadriga's IP camera.
10:00	SLAM is launched while Rover J8 is stationary. The green (UL) and red (DL) graphs diverge due to requests from the MEC. The blue graph (SLAM on LAN1) remains at zero since SLAM is not yet displayed on LAN1. PC1 (LAN1) subscribes to the map (in the Cloud) and LiDAR data.
10:03	Smartphone images (20%) start slowing down due to saturation. With the 80% slice using 37 Mbps UL, total DL is 115.5 Mbps, so four smartphones produce 78 Mbps. At 10:04, two SP3 subscriptions are stopped. The UAV's smartphone (SP2) is saturated and lost permanently. Only one Rover J8 smartphone subscription remains (20%) and SLAM is active (80%). By 10:05, all systems run smoothly again.
10:08	SLAM bandwidth: 7.2 Mbps. By 10:12, the UAV lands due to low battery. SLAM bandwidth increases up to 10:19, reaching 37.6 Mbps for the map plus LiDAR points, totaling 95 Mbps on the LAN1 router.
10:20	SLAM subscription is cut and relaunched at 10:21, generating a new map from scratch. The blue graph (SLAM) grows gradually.
10:23	Double SLAM subscription is set to visualize on PC1 (LAN1) while recording the map. One smartphone still transmits on the 20% slice.
10:24	Rover J8 starts moving in "follow me" mode. The second map is generated and recorded on a fast SSD.
10:27	SLAM (map) bandwidth: 820 KB/s. Two minutes later, it reaches 1.40 MB/s (11.2 Mbps). Traffic is duplicated (visualization + recording). To prevent issues, visualization is stopped, keeping only one SLAM subscription (recording).
10:29	SLAM (points and map) subscription is maintained from PC1 (LAN1). PC1 (LAN3) is subscribed to SP1 data. Stable smartphone data flow. After 10 minutes of SLAM operation, a 1.5 GB map is generated.
10:33	SLAM (map) bandwidth: 3.32 MB/s (26.56 Mbps). SLAM is stopped while the Rover J8 camera remains active without latency issues. The smartphone runs steadily at 17.91 Mbps.
10:34–10:45	SLAM remains stopped (experiment ends at 10:45). The final map reached 6.25 MB/s (excluding LiDAR), and the recorded .BAG file size is 2.1 GB.

machine (see Figure 4.10). In this case, the 5G router on board the robot must upload all data (GNSS position and LiDAR point cloud) to the cloud where the map is generated and stored, being able to share it with other cooperative agents, boosting the idea of cooperative SLAM.

Figure 4.12 shows the SLAM performed with the support of cloud processing, as seen in the RVIZ interface. Rover J8 began and ended its journey alongside the FCC location (inside the rectangular area in the northernmost corner, as shown in the image).

For the edge version, the Jetson AGX Xavier (NVIDIA) is chosen since it has plenty of computing power for this type of application, being a good HW option to use when there is no possibility of using a cloud resource. In this case, a ROS-bag file can be created in the robot LAN, storing the RAW RTK GNSS signals together with LiDAR data throughout the experiment, without consuming 5G network resources. The map is constructed using these data, running the SLAM algorithm on the robot LAN. This map can then be transmitted to the cloud to be shared with other cooperative agents that can use it for other applications. In this case, Rover J8 began and ended its journey at different points, but close to the FCC location. In addition, a log-file was recorded using jetson_stats, so that the use of HW resources can be studied. Although sometimes all the cores of this HW are in 100% usage, the tendency has been to use up around 45% of the CPU resources. Therefore, this HW can take over processing, at the cost of increasing the power consumption on board the robot. Thus, the CPU does not represent a bottleneck.

Key Performance Indicators from the Network Slicing

The data traffic generated by each UE and the most interesting KPIs of this experiment are listed below. Only if someone subscribes to information from a ROS publisher, traffic will be generated at the UE ends. The overall framework of this experiment is illustrated in Figure 4.13, with the details of the algorithm's execution shown in Figure 4.14.

- Smartphones SP1 and SP2 sent information from their internal sensors (camera, microphone, IMU, and GNSS modules) via the URD app, with the Full HD images being the heaviest data in the network, i.e., those requiring the highest bandwidth. The resolution of the smartphone cameras was decided and pre-set based on various studies carried out in previous trials to avoid saturating the 20% slice.

- The BCC hosted the ROS master of the ROS network B. The PCs in this network (LAN3) could request, in real time, to view Cuadriga's IP camera (AXIS P213), which generated UL data traffic on this robot's router. The resolution of the images was 704x576, producing a throughput of approximately 5 Mbps.

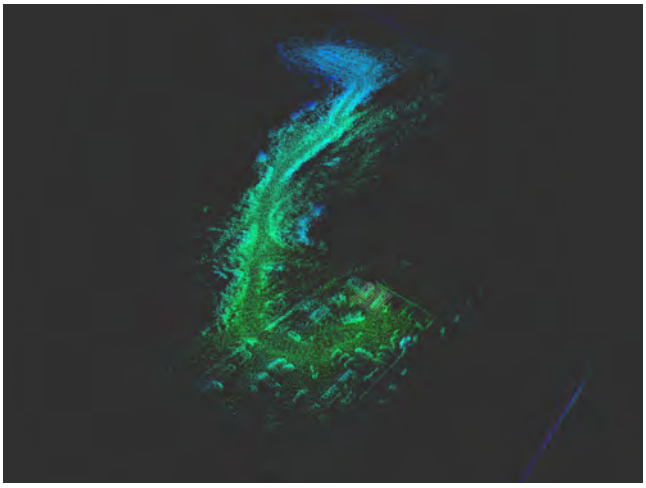


Figure 4.12 Rover J8 SLAM (cloud computing)

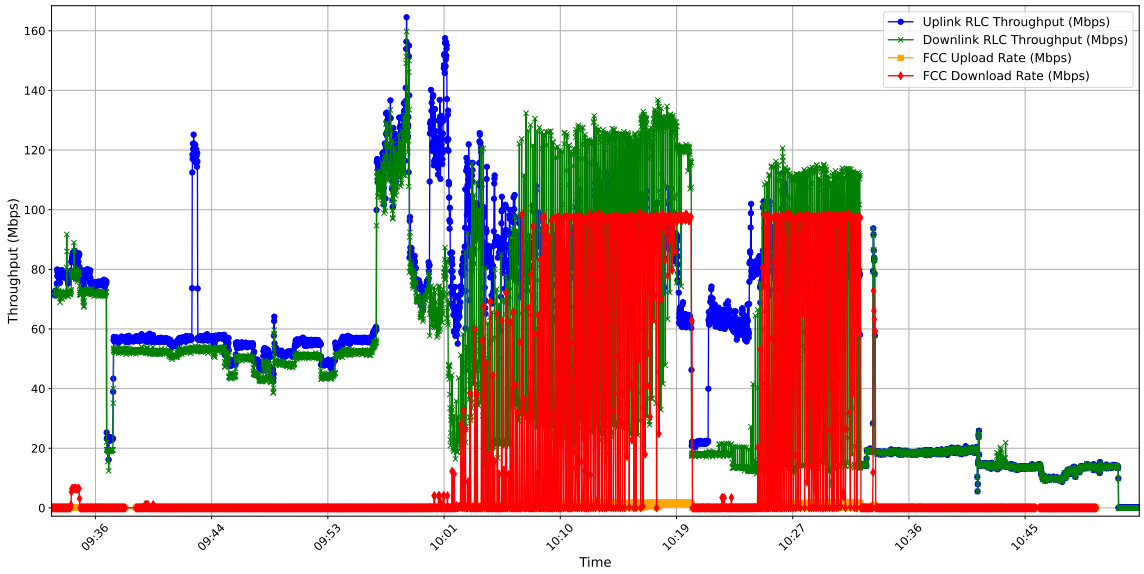


Figure 4.13 Throughput within the 5G cell during the experiment. Data passing through the UE in FCC (the router of LAN 1) is shown in red.

4.3.3 Real-Time FTM-based positioning systems

In this thesis, environmental sensing in ROUD scenarios has been supported by networks of low-power, medium- and long-range sensors, complemented by cameras and LiDAR—key tools in locating victims. Nonetheless, SAR teams require more advanced detection systems capable of accurately mapping victim positions and transmitting data in real time to the CCCA, enabling rapid evacuation and immediate assistance, especially since victims often remain immobile. A major challenge is ensuring that the emergency coordinator—typically based at a remote command center—receives this information quickly enough for it to remain

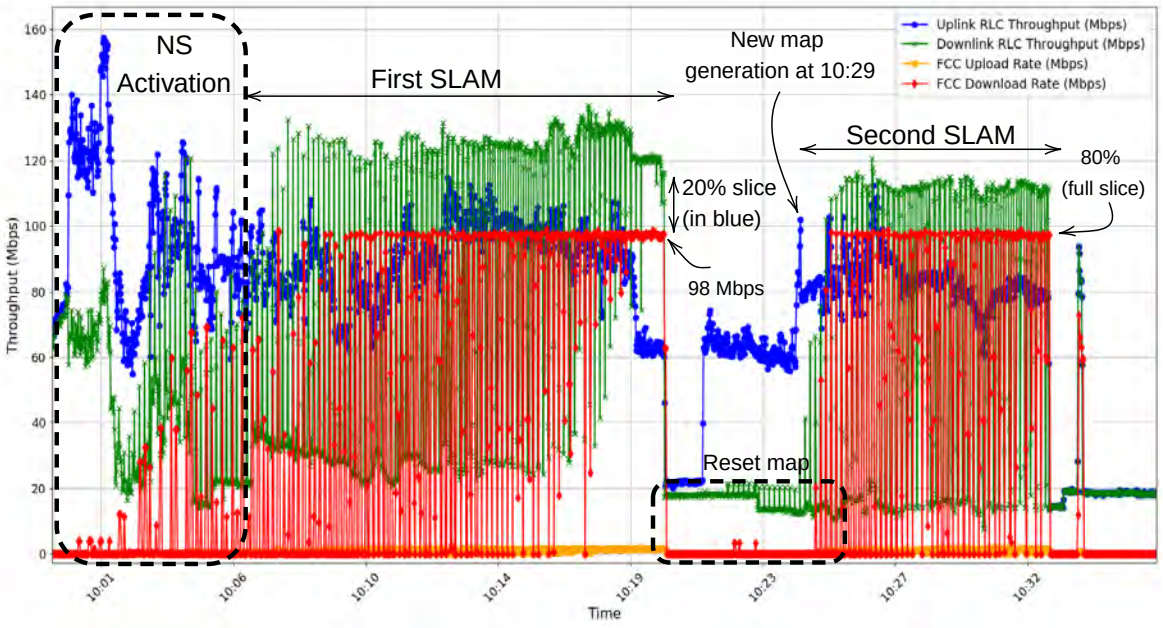


Figure 4.14 Throughput within the SAR-IoCA architecture in both SLAMs

useful for decision-making, enabling timely responses and dynamic adjustments to SAR operations as field conditions evolve.

Positioning methods determine global coordinates using techniques like trilateration using a GNSS receiver or triangulation through mobile phone base stations. However, these approaches depend on specific infrastructures or signals that are often unavailable in ROUD scenarios. In these scenarios, PVs can be unconscious or injured, being isolated, trapped, semi-buried, or visually hidden.

This work introduces an innovative real-time positioning system that leverages Fine Time Measurement (FTM), a feature of the IEEE 802.11mc amendment, to detect WiFi-enabled devices typically carried by potential victims. WiFi-based positioning using the FTM protocol relies on precise Round Trip Time (RTT) exchanges to estimate distances. Operating in the 5 GHz band with bandwidths ranging from 20 MHz to 160 MHz, WiFi FTM can achieve sub-meter accuracy under favorable conditions, with typical operational ranges between 50 m and 100 m. Although WiFi FTM is primarily intended for indoor use, its broad adoption in commercial devices and greater resilience compared to BLE and ZigBee highlight its potential for victim localization in ROUD scenarios. Given the lack of systematic outdoor studies, further evaluation of WiFi FTM in realistic SAR conditions is needed.

This section introduces two strategies for victim positioning in ROUD scenarios, integrated into the SAR-IoCA implementation. Both approaches assume the victim is equipped with a WiFi-enabled smartphone serving as a wearable transmitting device.

Strategy A: an FTM-based H-WSN to gather RTT packets

4.3.3.1 Design and implementation

This strategy involves deploying a hybrid network of FTM anchors distributed across the operational environment (see Figure 4.15). Each anchor exchanges RTT signals with a transmitter, potentially attached to the PV. If three or more FTM anchors detect the same RTT signal, multilateration can be applied to estimate the victim's position. This approach accommodates the possibility of a moving PV, assuming its movement is slow, as would likely be the case if the victim is injured or disoriented.

The proposed system architecture consists of three fundamental components, as illustrated in Figure 4.16: (i) a H-WSN composed of FTM-based anchors designed to detect RTT signals from WiFi devices that could be PVs within the operational environment; (ii) a fleet of mobile robotic platforms transporting mobile anchors, each equipped with GNSS receivers for continuous and accurate location tracking; and (iii) a Feedback Information System (FIS) [163] capable of remotely executing a multilateration algorithm using data collected from the agents' operational environment. The FIS provides graphical representations of localization data within the CCC, facilitating informed decision-making by the SAR coordinator.

In ROS 1, nodes must communicate directly with each other using TCP connections for topics, not just through the Master. Each node runs an XMLRPC server on a TCP port to handle registration and setup. To exchange messages across different networks (e.g., behind different routers), nodes must have direct IP reachability, the correct `ROS_MASTER_URI` configuration, and properly set `ROS_IP` or `ROS_HOSTNAME` to advertise accessible addresses.

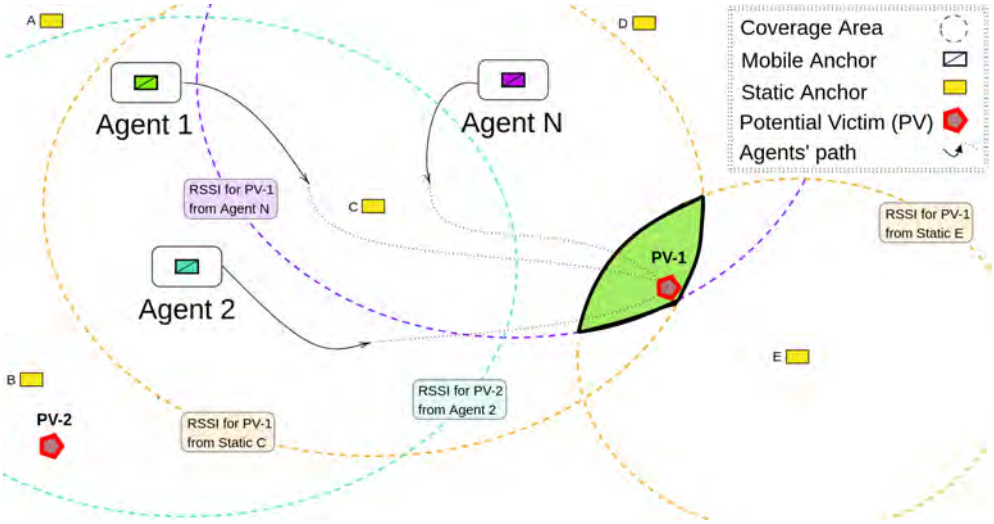


Figure 4.15 FTM-based H-WSN for detecting victims in ROUD scenarios.

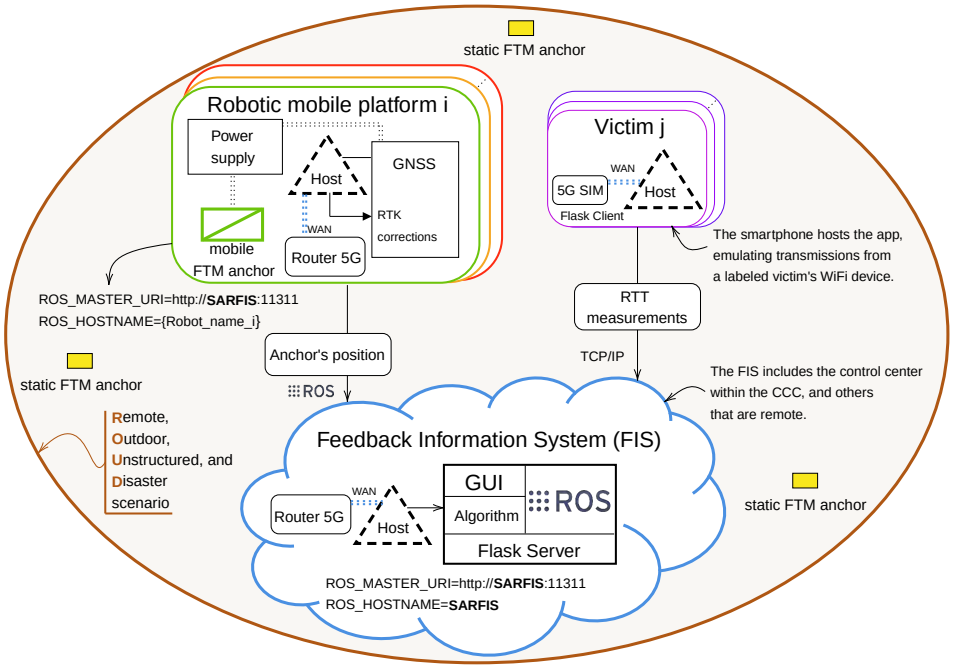


Figure 4.16 Fundamental components of the localization system.

Without this, even if nodes can register with the Master, they won't be able to establish peer-to-peer connections for topics. Common solutions include using a VPN to create a virtual LAN, setting up port forwarding on routers, or requesting static public IP addresses from the ISP to allow direct access.

VPNs provide an effective way to connect distributed ROS 1 nodes across different physical networks by creating a secure, private IP-level communication environment. This allows nodes to establish the necessary peer-to-peer connections required by ROS 1 for topic and service exchange. However, using a VPN can introduce additional latency compared to a direct LAN connection, and packet encapsulation overhead may lead to Maximum Transmission Unit (MTU) issues, potentially causing message fragmentation or connection instability if not properly addressed. Despite these challenges, VPNs remain a reliable solution to enable robust and flexible ROS 1 deployments over heterogeneous network infrastructures.

Among available VPN solutions, ZeroTier stands out for its simplicity, automatic network management, and its ability to optimize direct peer-to-peer connections whenever possible. It offers encrypted communication with minimal configuration effort, making it particularly attractive for ROS 1 systems that require rapid and stable network setups across diverse and dynamic environments.

Thus, all PCs onboard the robotic platforms and those within the FIS are interconnected through a VPN, which also enables communication with the victims' smartphones. The system uses ZeroTier, a peer-to-peer VPN that includes end-to-end encryption and is specifically suitable for robotic applications with ROS due to its low-latency architecture that does not route traffic through central servers. We opted for ZeroTier over similar solutions such as Husarnet—which is also based on ZeroTier—because of our prior experience and the simpler, more scalable network management it offers. Each agent is assigned a unique hostname and a virtual IP within the private VPN, which is centrally configured and monitored from the CCC center.

The main challenge lies in managing and processing WiFi signals from lost UEs, such as smartphones, within the deployed H-WSN. To enable efficient experimentation, we use an Android application [174] installed on the victims' smartphones. While deploying a mobile application on victims' devices may seem unrealistic in real-world scenarios, this approach allows us to: (i) facilitate communication between the operational environment and the CCC; (ii) support testing of localization algorithms in ROUD scenarios while reducing costs; and (iii) minimize system latency. This strategy enables testing different multilateration algorithms in real time without relying on local configurations, effectively demonstrating the practical functionality of our system.

Our purpose is to demonstrate the implementation of the proposed real-time localization system using the Iterative Weighted Least Squares (IWLS) algorithm, which generalizes Ordinary Least Squares (OLS) by incorporating dynamic weighting schemes to handle noisy data or outliers [175]. Additionally, we provide, along with a raw dataset, detailed guidance on applying other algorithms and/or filters.

The localization procedure begins by receiving WiFi RTT distance measurements and extracting the known positions of the anchors. These GPS positions are transformed into Earth-Centered, Earth-Fixed (ECEF) Cartesian coordinates to operate within a uniform metric space. An initial position estimate is selected near the anchor reporting the shortest measured distance to the device, and an iterative refinement process is applied. In each iteration, estimated distances are computed, the Jacobian matrix is built, and the position is updated using weighted least squares. The key aspect of IWLS is that observation weights are dynamically updated based on residual errors, following the selected strategy (OLS, Huber, Trimmed, or Tukey). Once convergence is reached, the estimated position is converted back from ECEF to GPS coordinates to provide a geodetically meaningful output.

The mobile application sends RTT information in real time to the remote FIS, indicating which anchors (identified by their MAC addresses) have detected the smartphone and their respective distances, as shown in Figure 4.17. We assume smartphones have an active SIM

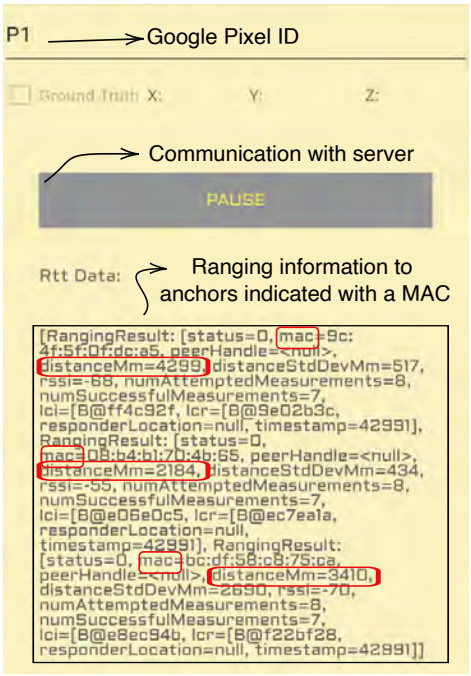


Figure 4.17 Android application interface.

card; however, in ROUD scenarios, coverage loss or degradation is common. The proposed system only requires an Internet connection to join the VPN; functionality is automatically restored when the connection is reestablished. RTT packet readings are associated with the GNSS positions of the anchors, whose measurements achieve RTK-level accuracy depending on the available differential corrections, thus impacting overall system accuracy. The H-WSN nodes (FTM anchors) are based on Google WiFi Mesh devices [176].

Two hardware versions have been employed: model GL0102, operating at 5 V and 3 A (15 W) via USB-C, and model GJ2CQ, operating at 14 V and 1.1 A (15.4 W) via a barrel connector. Each anchor is powered by a 20000 mA h external DC battery. The GL0102 model achieves up to 6.7 h of continuous operation, whereas the GJ2CQ model reaches approximately 5.5 h due to conversion losses. GL0102 units are preferred for static anchors and UAV-mounted nodes, where autonomy is critical, while GJ2CQ units are deployed mainly on UGVs, which can supply continuous power. The combined weight of a static anchor and its power source is approximately 500 g, facilitating rapid deployment without hindering SAR operations.

Smartphones emit probe requests over both 2.4 GHz and 5 GHz bands when WiFi is enabled, even if not connected, provided the device supports dual-band operation. While Google WiFi can detect beacons in both bands, FTM sessions—and thus effective localiza-

tion—occur only if the smartphone operates in the 5 GHz band, due to hardware limitations favoring this less congested frequency with better temporal resolution.

Mobile anchors are integrated into autonomous robotic platforms. UGVs can operate for several days without recharging, while UAVs are deployed for flight sessions of approximately 25 min, with their onboard anchors independently powered. This setup ensures that the system meets the energy and operational demands of time-sensitive SAR scenarios while maintaining deployment flexibility.

The Google WiFi device uses only the non-DFS (Dynamic Frequency Selection) 5 GHz channels (36 , 40 , 44 , 48) for WiFi FTM, as it avoids DFS channels (52 to 144) to ensure quick startup and stable operation without radar scanning delays. This restriction limits the available spectrum for wide channels (e.g., 80 MHz), which are essential for higher temporal resolution in FTM-based ranging. While using these lower channels reduces complexity and improves compatibility, it can also lead to increased multipath interference in dense indoor environments, as the limited frequency diversity may cause reflected signals to be indistinguishable from direct paths. This can degrade positioning accuracy. However, in low-interference or open environments, the use of clean 80 MHz non-DFS channels can still provide reliable sub-meter accuracy.

The victim's smartphone forwards RTT measurements to a Flask server [177] connected to the multilateration algorithm, which is integrated within the ROS 1 architecture. The mobile application allows the selection of the socket (IP and port) for RTT data transmission, enabling a direct connection to the server in the FIS, where the ROS master is hosted

The RTT information is then integrated into ROS and processed remotely together with the GNSS position of the anchors, which is published by the PCs on board the robots in ROS topics. Once the FIS collects all the necessary information, two dedicated computers evaluate the effectiveness of the positioning system in real time. Two different PCs are used to distribute the computational load, as each has to process other information from the application scenario. However, the whole system can run on one PC and even in a virtual container (as presented in the open repository) to launch via a portable image. Using more than one PC, they can be in a different physical location if they operate on the same VPN, which is useful for providing visual information to SAR team coordinators at different locations.

The FIS is composed of three running processes:

- The Flask server, which receives the RTT and RSSI information through the smartphones carried by the victims, including their labels to register them.

- The multilateration algorithm, in which different ROS nodes subscribe and publish information of interest. For instance, the algorithm must be able to get the RTK GNSS measurements via ROS subscription, while it has to publish the estimated coordinates for the PVs, including those WiFi devices in the operational environment that are not true victims.
- The *SARFIS tool* which displays, in real time, the positions of mobile and static anchors, as well as other visual information of interest. At this point, the *SARFIS tool* can receive information through ROS.

On the one hand, one PC hosts the Flask server, the ROS master node, and runs the multilateration algorithm, producing the results it publishes in different ROS topics. This information is available on any PC with the proper configuration in the ROS network.

As it operates with ROS 1, all hosts on the network must know the master's virtual IP or, alternatively, its hostname (registered in the /etc/hosts file of each device). Figure 4.16 illustrates how to set environment variables, while the shared public repository provides more detailed implementation instructions. Lastly, the Flask server must run in a separate thread to synchronize the reception of WiFi data (from Flask clients hosted on victims' smartphones) with the ROS nodes responsible for distributing information across the system. Although the Flask server does not run on ROS, it opens a port (5001 by default) to allow access to Flask clients. Meanwhile, ROS nodes utilize the ephemeral ports of each PC (running Ubuntu 20.04).

On the other hand, another PC runs the *SARFIS tool*. Whenever an anchor detection is generated, a circumference with a radius equal to the distance estimated by RTT will be displayed in real time. In this way, the CCC is alerted that a PV may have been detected in the vicinity of the anchor, whose position is the center of the circumference. In addition, the detected RSSI value is located at the edge of the circumference, which relates to the proximity to the transmitting device concerning the pose of the mobile robot carrying the anchor.

4.3.3.2 Experimental results

A realistic experiment was conducted in JEMERG XVII. Table 4.2 presents the open-source dataset, including input and output information to the FIS, useful for offline reproduction experiments. The explored terrain has a perimeter of 180 meters and an area of 2,000 square meters. Two fake victims were located on the ground, each with a smartphone in their

pockets. Both were hidden, although one of them was above ground (X3) and the other (P1) was covered with scrap metal. The actual distance (in space) between them is 3.5 meters.

Human SAR agents initially deployed four static FTM-based anchors, identified as 1F, AB, A5, and AA, in tactical positions considered safe zones with LoS to the catastrophic scenario. These anchors were equipped with portable external power supplies and strategically placed to cover a broad area for subsequent exploration by robotic agents.

To accurately register the positions of the static anchors, human agents used smartphones connected to RTK GNSS modules. When placing an anchor, the operator placed the smartphone statically on it and waited to send five fixed positions via ROS [165], ensuring the decimal values remained constant to guarantee a fixed position with an error margin of 2–10 cm. The geolocation of each anchor was transmitted on the ROS topic /Anchor{label}/position and stored in the FIS, where positions were displayed on the GUI with corresponding labels.

In addition to the static anchors, two mobile robots (Rover J8 and Atyges FV8) explored the area within the ROUD scenario, introducing two additional FTM-based anchors in the area of interest. Their detections were published on the topics {robot_name}/wifi_rtt_estimations.

The UGV and UAV were equipped with necessary sensors and communication systems to detect WiFi RTT signals from PVs. The UAV was teleoperated and took off from a point of interest, while the UGV followed a human agent using a LiDAR-based *follow me* mode and maintained its position upon reaching a designated point. Both mobile robots were exploring

Table 4.2 Number of ROS messages during the experiment.

ROS topic	**Nº of messages**
FTM anchors' positions (*sensor_msgs/NavSatFix*)	
/Anchor{*label*}/position	5
/FV8/mavros/global_position/raw/fix	885
/RoverJ8/gps0/fix	3539
Victims' detections from the H-WSN (*std_msgs/String*)	
/Anchor1F/wifi_rtt_estimation	687
/AnchorAA/wifi_rtt_estimation	6
/FV8/wifi_rtt_estimation	1427
/RoverJ8/wifi_rtt_estimation	1414
Algorithm outputs (*std_msgs/String*)	
/geo_multilateration	529
Offline reproducible data (*std_msgs/String*)	
/infoServer	1692
/inputServer	1682

the area for 7 minutes and 22 seconds. The UAV's pilot conducted the scan by performing circular traverses at different altitudes to capture possible victims with the UAV camera.

PVs were emulated using smartphones: P1, a Google Pixel 3, was buried and hidden, while X3, a Xiaomi Mi10 T Lite, was placed semi-buried on the surface. Their true positions were recorded before the exercise as ground truth.

Two static anchors (AB and A5) did not detect any RTT signals from the victims due to suboptimal positioning. However, this was not considered an error but a probabilistic outcome, as the human agents could not know the victims' locations beforehand.

The trajectories of the UGV (red) and UAV (blue) are illustrated in Figure 4.18. Upon detecting the two victims (P1 and X3), already being detected by the static anchors, the CCC instructed the UGV to scan the area, resulting in two mobile anchors operating during the final five minutes of the exercise. Figure 4.19 shows the UAV's camera view at 21 meters altitude, displaying the UGV following a human agent at approximately $1\,\mathrm{m\,s^{-1}}$.

The geo_multilateration topic contained all positions estimated by the multilateration algorithm, including any WiFi devices within H-WSN coverage. The smartphones emulating the victims transmitted RTT measurements with identification tags. Victim P1 was geolocated only four times due to being buried, whereas X3 was localized 289 times. The algorithm computed 236 geolocations for potential victims, intentionally including additional detections to verify the system's ability to identify wearable devices with WiFi in the environment.

RTT messages for P1 included 227 from the UGV and 179 from the UAV. For X3, there were 316 messages from the UGV and 406 from the UAV, indicating that the semi-buried victim (P1) was less frequently detected than the semi-exposed victim (X3) despite their proximity.

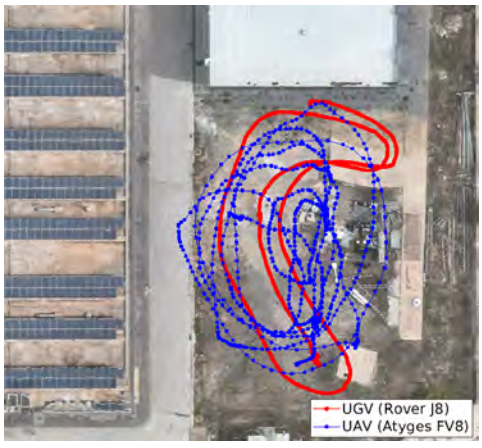


Figure 4.18 Trajectories followed by the mobile anchors on UGV (red) and UAV (blue).

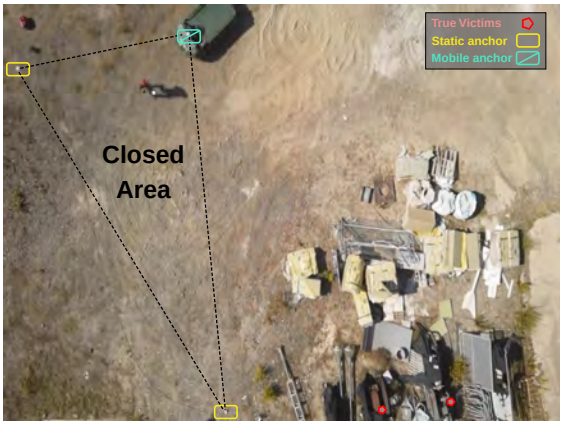


Figure 4.19 Field of view of the UAV when hovering 21 m above the UGV.

Figures 4.20 and 4.21 show the RSSI values of the detections made by the anchors onboard the robotic agents, along with the altitude of these along their scan. In addition, a comparison between the RTT-estimated distances to the victims and the real distance, taking as ground truth the GNSS receiver of each robot, with its differential corrections, is included. Each point represents a single detection of each mobile anchor, significantly outperforming the static H-WSN detections. For example, node AA detected a WiFi device only six times, nodes AB and A5 detected none, and node 1F contributed 687 detections to the algorithm.

In addition, the UGV remained stationary at the designated point, during which the RSSI of its detections stayed stable around –75 dBm. Two minutes and 32 seconds after the UAV's takeoff, the UGV began moving, allowing analysis of GNSS receiver error. The UGV's altitude fluctuated by up to 1.49 meters over 152 seconds despite differential corrections (Figure 4.20), i.e., the state of the measurements had a DGPS accuracy, primarily due to the ROUD environment's surrounding structures and buildings. These inherent GNSS errors suggest lower than expected accuracy in the multilateration algorithm's estimations due to unstable communications. Moreover, it is observed that the distance estimates by RTT are quite close to the ground truth, at least for the transmitter on the ground surface (X3).

Figure 4.22 shows the empirical cumulative distribution function (ECDF) of distance and horizontal positioning errors for both victims. The most occluded one (P1) was geolocated only 4 times, despite being detected around 200 times by each mobile agent. The only static anchor potentially useful for trilateration (AA) detected P1 just 6 times and never detected X3, making it insufficient to assess accuracy for buried victims.

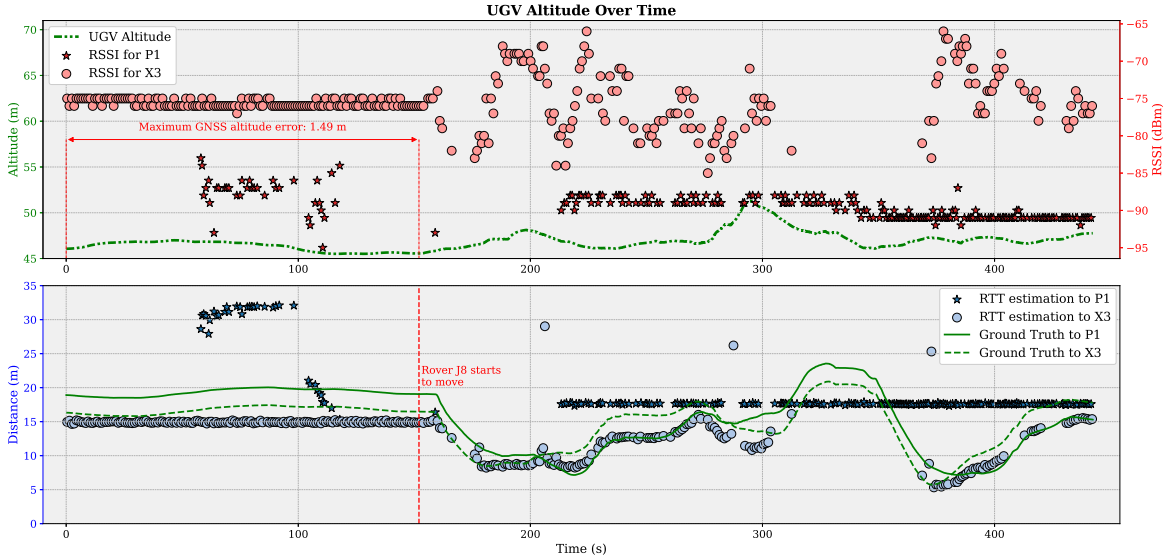


Figure 4.20 Victim detections and RTT distance estimated from the UGV's anchor.

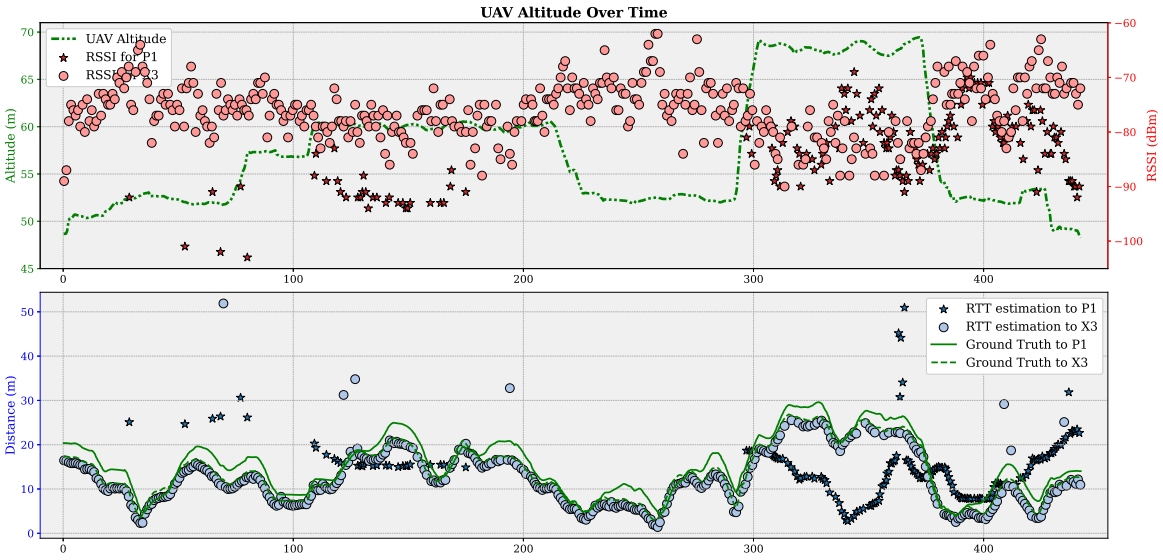


Figure 4.21 Victim detections and RTT distance estimated from the UAV's anchor.

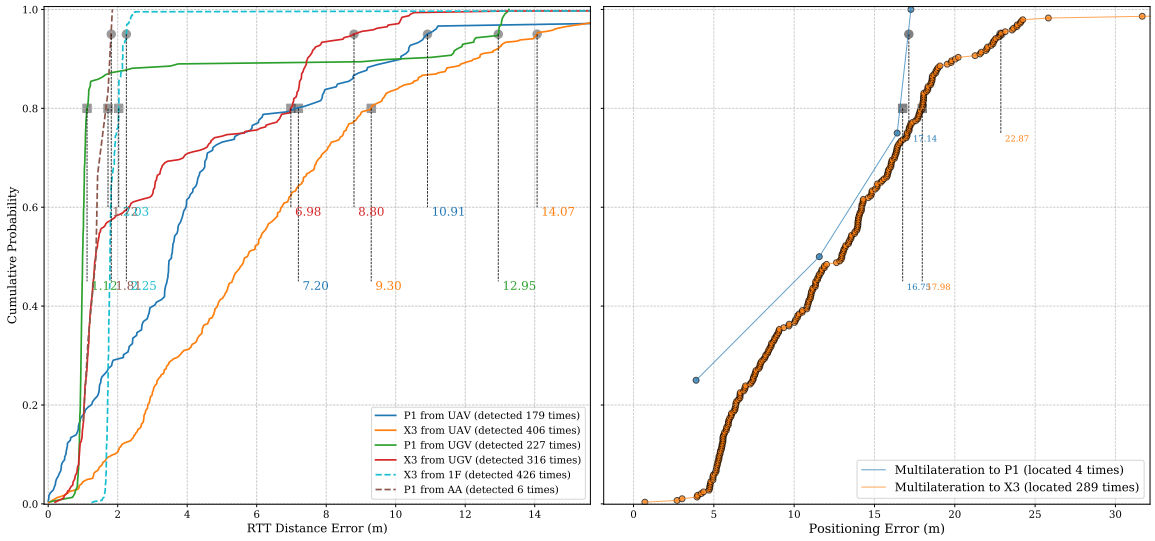


Figure 4.22 ECDF of distance and horizontal positioning error for both victims.

On the other hand algorithm converged 289 times for the semi-hidden victim, confirming the system's utility for SAR teams as it allows quick localization of wireless devices at disaster scenes, even if semi-hidden or partially buried. Percentiles have been superimposed for all graphs. The black squares represent 80%, while the black circles indicate 95%.

Table 4.3 presents the results of the multilateration process. The localization system achieved an estimated distance error of 12.97 m for X3 and 13.98 m for P1 concerning their absolute positions, considering the median of all geolocations. Additionally, the 80% and

95% sorted horizontal errors compared with the ground truth are shown. This table shows that three anchors viewed the X3 victim more than 300 times, resulting in 289 geolocations.

Error	80% Percentile [m]	95% Percentile [m]	Nº of detections
RTT (Round Trip Time)			
X3 from UGV	6.96	8.80	316
X3 from UAV	9.28	14.05	406
X3 from 1F	2.03	2.25	426
P1 from UGV	1.10	12.95	227
P1 from UAV	7.13	10.91	179
P1 from AA	1.72	1.81	6
Multilateration			
X3	17.98	22.87	289
P1	16.75	17.14	4

Table 4.3 Results for P1 (semi-buried) and X3 (hidden).

In the case of the UGV, its on-board anchor and detections are shown in red; the UAV's detections and GNSS RTK position appear in blue. Anchor altitude is represented in black, while RSSI values are displayed at the circles' edges to indicate signal strength (see Figure 4.23). When the algorithm converges on a position, it is marked on the orthophoto with a blue circle and the corresponding PV label. Other WiFi devices may also be detected—for instance, X2, a moving smartphone intentionally placed in the scenario, was identified by the UAV (as shown in one of the labeled red circles).

Figure 4.23 The *SARFIS tool* showing RTT and multilateration estimations in the CCC.

Strategy B: One Search Agent (OSA) to gather RTT packets

4.3.3.3 Design and implementation

This strategy utilizes one single mobile anchor to detect the victim's signal (see Figure 4.24). In this approach, multilateration becomes feasible starting from the third detection by the mobile anchor. However, this method is only suitable for static targets, as the relative motion of both the anchor and the target would compromise the reference needed for accurate flight time measurements. In this case, a static position is assumed for the victim.

The proposed system architecture consists of two fundamental components, as illustrated in Figure 4.25: (i): One Single Agent (OSA), preferably a UAV (as it provides a clearer LoS with the terrain compared to a UGV), designed to detect RTT signals from PVs while exploring the ROUD scenario; and (ii): a FIS integrated in a ROS 2 network and connected to the multilateration algorithm, which uses data collected from the OSA's operational environment.

The FIS is composed of the same three running processes implemented for the strategy A: the Flask Server, which receives the RTT packets from the PVs' smartphones, the multilateration algorithm, and the *SARFIS tool*. However, in this case, the IoCA architecture has been migrated to ROS 2.

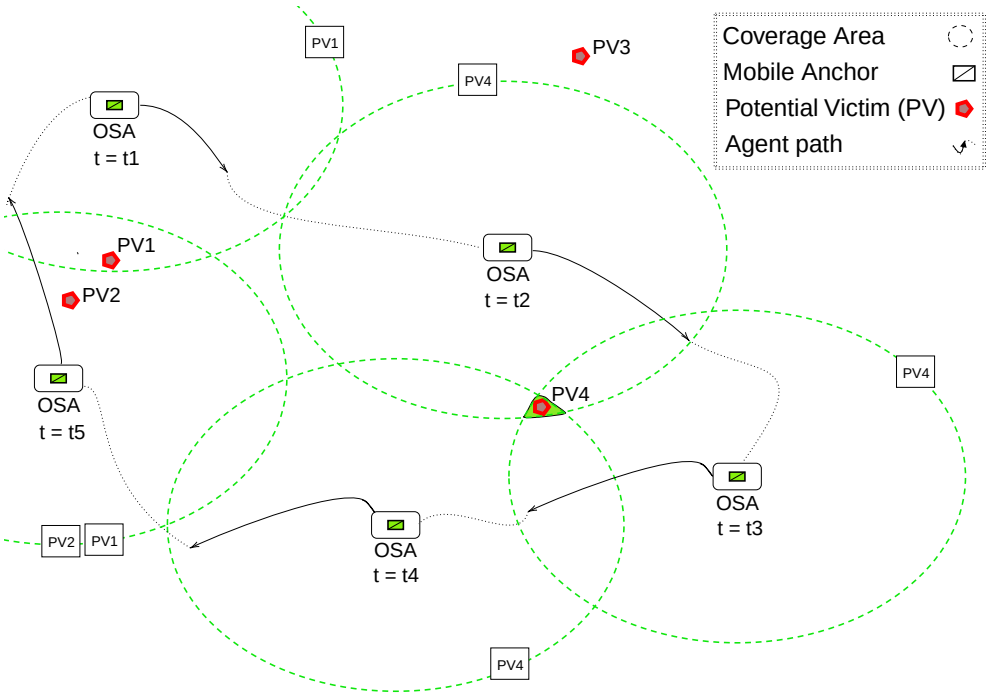


Figure 4.24 Strategy B: One Search Agent (OSA)

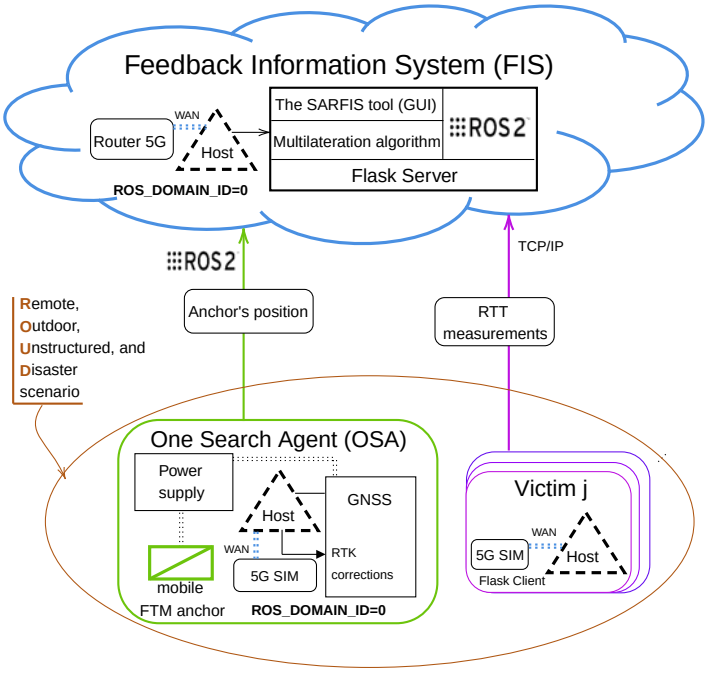


Figure 4.25 Architecture of OSA-based positioning system

For system validation, the developed 5G ad-hoc sensor is used as the host of the OSA, which in this case is an UAV. Given the limited UAV's payload (and the influence of it on their flight autonomy), a smartphone making use of the UR2A application has been mounted on it for the experiments. Using the developed UR2A application, the smartphone is included in the ROS 2 architecture. It avoids shipping a mini-PC and router to extract and transmit data from the own GNSS modules on the UAV, reducing the weight and cost of the system.

The UR2A application has been tested in two different smartphone models: Google Pixel 7 Pro and Huawei P40 Pro. The external GNSS module (zed-F9P RTK model by u-blox), connected via USB-C to the smartphone, receives differential corrections from the *MLGA 13460M001* station located in Málaga, thus acquiring RTK-Fix precision (1-3 cm error). Figure 4.26 shows the positioning error in the three Cartesian axes during a real test flight. The error measures the difference in positioning of the internal GNSS of the smartphone with respect to the same smartphone using the external RTK GNSS module.

In addition, zenithal images can be transmitted through ROS 2, publishing the images captured by its rear camera and the rest of the data from its internal sensors (camera, microphone, IMU, and GNSS module) by implementing various ROS nodes in the smartphone.

The application has two modes for publishing images: intermittent (0.5 Hz) or fluid (30 fps). The QoS was set to *best effort* on the image publisher, as the most current message

should be prioritized, regardless of the loss of older messages. For the rest of the information, *reliable* was used.

4.3.3.4 Experimental results

The experiments were conducted during the JEMERG XVIII experimental exercises in the operational area of LAENTIEC. Compared to the scenario presented in Strategy A, this application scenario was larger, more realistic, and irregular. The UAV took off from an altitude of 99.5 meters but only detected RTT signals after reaching an altitude of 103.8 meters. This highlights the significant impact of the mountainous terrain, with its valleys and hills, on maintaining a clear LoS.

Figure 4.27 shows the RTT detections for the two victims in the terrain: X4 (partially hidden) and P1 (semi-buried). The geolocations resulting from applying the multilateration algorithm for both victims (P1, red circles; and X4, green asterisks) indicate a higher accuracy for the less hidden smartphone (X4).

Figure 4.28 shows a snapshot of the exercise (the full video can be seen in the repository). In this frame, the UAV's position is shown in red, being the center of two circles, one for each PV detection, based on the detection of its WiFi FTM signals. The SAR-FIS interface shows the previously generated orthophoto of the operational environment. The UAV is flying over an area complex for humans to access to detect and then geolocate the victims. Each circle shows the RSSI of the detected signal, and its radius is equal to the estimated distance times the flight time, which is very informative for the SAR agents, who are supervised from the FCC. The anchor onboard the UAV is scanning these signals during its flight and sends raw data as an input to the multilateration algorithm, together with the device label. Then, the algorithm gradually converges to a geolocation, proving latitude and longitude for each PV.

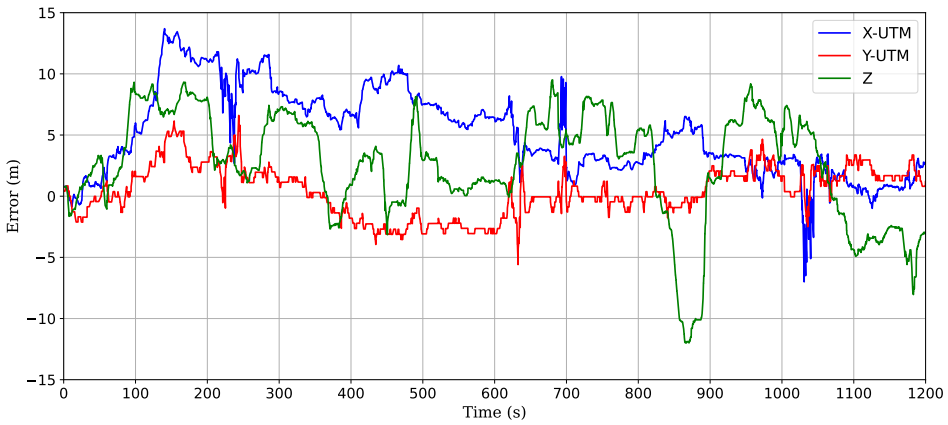


Figure 4.26 Position error between data from internal and external GNSS modules

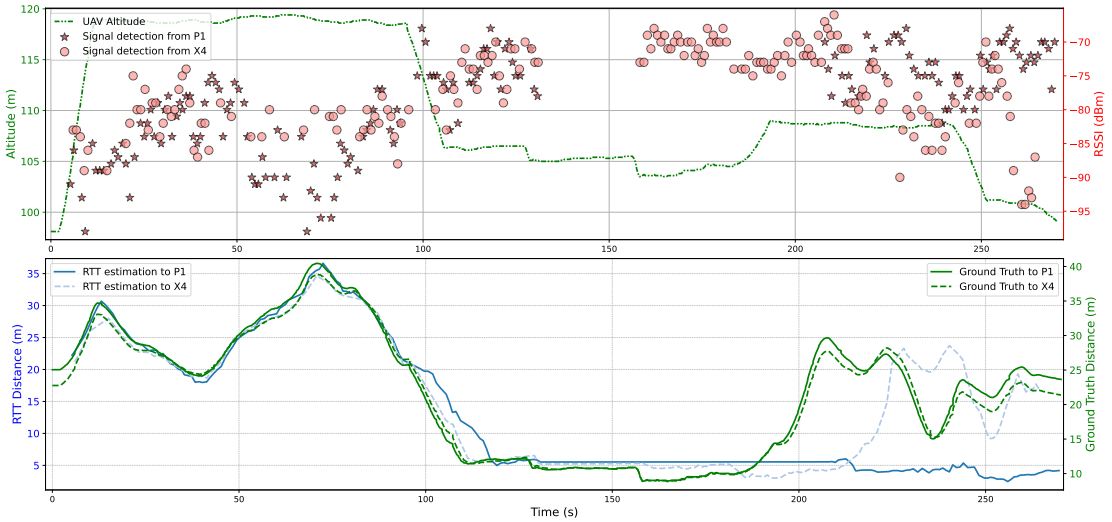


Figure 4.27 Visualization of the UAV's ground truth (in green), RTT, and RSSI data during a five-minute exploration flight

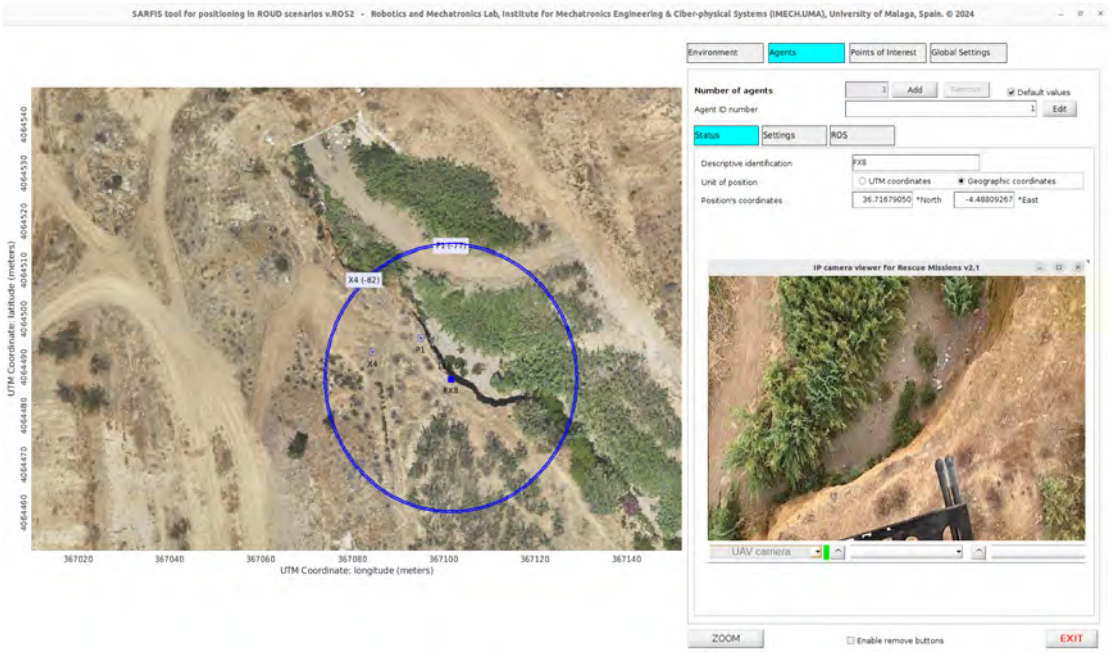


Figure 4.28 Screenshot of the *SARFIS tool* during an active search mission with both PVs being detected and geolocated by OSA

The real position of both devices has been taken before the flight to provide ground truth. Both the real positions (tiny colored points) and those estimated by the algorithm (small blue circles with the PV label) are displayed in real time.

4.3.3.5 Evaluation

Comparing the two strategies, it is observed that with Strategy A, a significant amount of information is lost because detections made by a single mobile anchor must coincide in time with detections from two other anchors. In contrast, the number of detections in Strategy B is fully utilized, as it does not depend on other anchors. Although the detections unused in Strategy A are leveraged by using Strategy B, Strategy A enables the geolocation of moving devices, as the receptions considered by the algorithm involve different references. In Strategy B, it is only possible to position static objects, such as a buried victim or one whose movement is restricted. Additionally, Strategy B has the disadvantage of not being fault-tolerant.

Strategy A

Implementing the proposed system in real-world scenarios revealed several technical and operational challenges that need addressing to enhance efficiency and practicality. A primary issue is the system's complexity, requiring multiple elements for successful deployment. The presence of various static and mobile anchors and the need for cooperative robots result in a slower deployment process than ideal.

Limitations of UGVs present another challenge, as they often cannot traverse rough or obstructed terrain, which is common in ROUD scenarios. While UGVs offer payload capacity and stability, their restricted mobility constrains their effectiveness. Integrating UAVs provides an alternative for detecting and positioning victims in inaccessible areas but introduces the issue of limited battery life, necessitating effective management strategies for continuous coverage. Extending UAV autonomy is crucial for enhancing system resilience in prolonged SAR missions.

UGVs typically used as casualty evacuation platforms, can remain static at points of interest while UAVs scan the area. Observations indicate that while the UGV remains stationary, it continues to detect signals from PVs. Analyzing RSSI patterns could model whether a detected PV is above ground, hidden, or semi-buried. Figure 4.20 shows that when the UGV was static, the surface victim (X3) maintained a regular RSSI value, while the semi-buried victim (P1) exhibited varying values. When the UGV moved, the opposite occurred, suggesting that obstacle presence around the connection ends affects signal stability.

Currently, RTT estimation relies on measurements taken by the UE (the victim's smartphone) instead of the AP (anchor) due to privacy concerns. A potential solution is to determine distance through the AP while anonymizing the UE's identity to prevent exposure to

personal information. This approach requires further research into secure, privacy-preserving methodologies for distance estimation in public safety applications.

Processing localization data on the edge, directly onboard UGVs and UAVs, would improve computation time for real-time localization. Thus, it is crucial to work on developing hybrid platforms where the architecture prioritizes computation on the IoRT. Moving part of the computation to the edge and leaving only the monitoring of the results to the Cloud may be a more robust solution. However, this implies the development of ad-hoc FTM anchors, which can serve as IoRT. In this way, the bottleneck in processing times and communication latencies can be reduced. These improvements would enhance a system with more accurate and immediate updates of estimated geolocations.

Strategy B

The UAV, acting as OSA, flew over an area that is difficult for ground mobile robots and humans to access when searching for victims. Each circumference in Figure 4.28 represents the RSSI of the detected signal, with its radius corresponding to the distance estimated by RTT. This information is highly valuable for SAR agents, who are supervised from the command and control center to facilitate rapid ground interventions.

The anchor onboard the UAV continuously scans wireless signals throughout its flight, transmitting raw data to the multilateration algorithm along with the corresponding device labels (P1 and X4 in this case). This iterative process enables the Cloud algorithm to progressively refine the geolocation estimate, ultimately providing latitude and longitude coordinates for each PV, which are displayed in real time on the map in the *SARFIS tool*. The algorithm's results are plotted as small blue circumferences with the victim identifier. This visualization provides first responders with immediate situational awareness, yet further improvements in localization accuracy and robustness are required to ensure reliability.

However, discrepancies between estimated and actual locations may arise due to the dynamic nature of the UAV's movement, environmental conditions affecting signal stability, and communication losses. Issues such as Internet disruptions, VPN failures, or the anchor's inability to process FTM packets in real time can introduce delays in the estimation process. This effect is particularly evident in the final phase of the flight (see Figure 4.27), where multilateration was applied using anchor positions that were not synchronized with RTT receptions, leading to noticeable localization offsets.

In the event of a communication loss or hardware failure, the system would need to be reset, which is not trivial while the UAV is in operation. This issue could be mitigated by migrating the proposed architecture to the edge, meaning that all the elements in the FIS should move to the UAV platform, while the GUI remains responsible for informing the

SAR coordinator in the CCC. To achieve this, the FTM initiator would need to be positioned onboard the UAV. However, this was incompatible with the current architecture, which relies on an Android app to transmit RTT estimations. Therefore, extending the OSA strategy to allow the UAV's onboard device to function as an FTM initiator is essential.

Moreover, the *SARFIS tool* also allows real-time monitoring of the cameras of the agents selected in the GUI. In this case the image of the zenithal camera of the UAV is observed, which is georeferenced in the orthophoto, being able to see its position's coordinates.

In addition, smartphones, acting as 5G ad-hoc perception devices, provide a flexible and cost-effective alternative for UAV-based sensing. Their ability to integrate with externally attached RTK GNSS modules enables precise georeferencing of images and accurate positioning of UAV detections. However, the effectiveness of this approach is influenced by factors such as network availability, GNSS signal quality, and the computational constraints of mobile devices. These limitations can impact the real-time performance and accuracy of the system, particularly in environments with poor connectivity or signal interference. Despite these challenges, the ability to integrate RTK GNSS modules with smartphones offers a promising alternative for UAV-based perception, reducing costs while maintaining high localization accuracy. A key consideration is balancing the trade-offs between hardware limitations and the benefits of a flexible, ad-hoc sensing approach.

Figure 4.29 shows the ECDF of distance and horizontal positioning error for both victims (P1 and X4, in this case). The results show a much higher number of geolocations compared to strategy A. In addition, the accuracy has improved considerably.

The contents of the dataset (available in the open-source repository) are shown in Table 4.5. On the other hand, Table 4.5 presents the estimates for both victims based on RTT and multilateration estimates, considering different minimum flight altitudes.

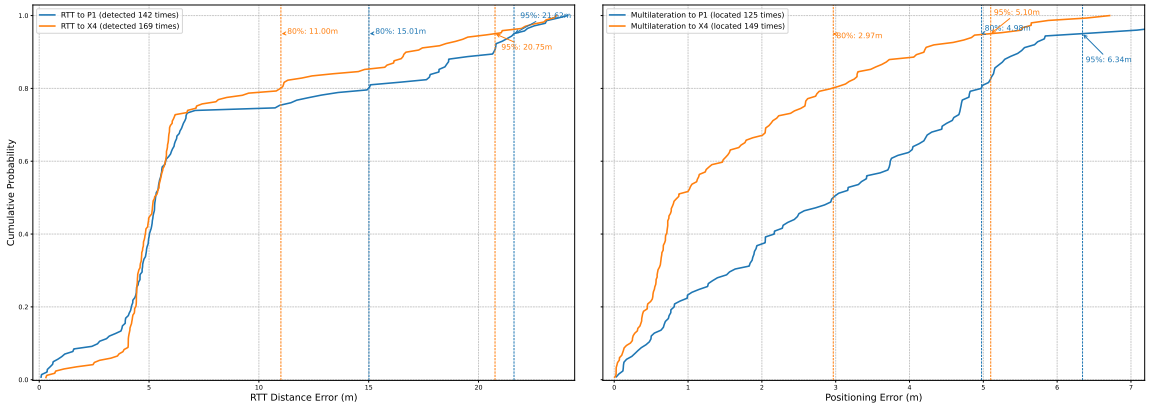


Figure 4.29 ECDF of distance and horizontal positioning error for both victims

Table 4.4 Dataset Collected in JEMERG 2024

ROS Topic	Description	Message Count	Details and Observations
/FX8/multilateration_all	Provides the outputs from the multilateration algorithm for all detected PVs.	335	A median-based filter is applied to remove some outliers.
/FX8/fix	RTK GNSS data.	1843	This data is obtained through a 5G ad-hoc sensor onboard the UAV (ground truth).
/Anchor1F/wifi_rtt_estimation	String data containing three values: WiFi RTT distance estimation, RSSI values, and PV label to identify the PV.	376	WiFi RTT provides distance estimates by measuring the time taken for a signal to travel from the anchor to the robot and back. RTT values are less precise than GNSS but useful in certain environments.

Table 4.5 Multilateration and RTT errors for a five-minute search flight: raw and median-based filtered data

From take off at 99.5 m (WGS84) to landing at 98.8 m (WGS84)	**X4 (RTT)** 186 detections	**X4 (Multilateration)**		**P1 (RTT)** 166 detections	**P1 (Multilateration)**	
		Raw	**Filtered**		**Raw**	**Filtered**
Mean	7.43 m	4.47 m	2.13 m	9.70 m	8.78 m	4.50 m
95th Percentile	20.12 m	13.59 m	6.50 m	21.92 m	18.72 m	15.69 m
When flying above 105 m (WGS84)	**X4 (RTT)** 140 detections	**X4 (Multilateration)**		**P1 (RTT)** 141 detections	**P1 (Multilateration)**	
		Raw	**Filtered**		**Raw**	**Filtered**
Mean	7.97 m	4.15 m	1.93 m	8.04 m	7.90 m	3.27 m
95th Percentile	21.62 m	9.91 m	5.51 m	21.12 m	11.80 m	6.41 m
When flying above 115 m (WGS84)	**X4 (RTT)** 49 detections	**X4 (Multilateration)**		**P1 (RTT)** 75 detections	**P1 (Multilateration)**	
		Raw	**Filtered**		**Raw**	**Filtered**
Mean	4.88 m	4.20 m	1.43 m	4.90 m	9.11 m	3.67 m
95th Percentile	5.71 m	10.86 m	4.32 m	6.17 m	44.46 m	10.55 m

Several additional tests were conducted, using both a mini-PC onboard the UAV with an edge architecture, and connecting the smartphone to the same LAN, i.e., transmitting images without using the WAN. Images in 4K were received using the intermittent mode at 0.5 Hz in the UR2A application. In fluid mode, images were received at 30 fps using the lowest resolution. Therefore, the combination of georeferenced images along with position estimation through multilateration using the OSA has represented a significant advancement compared to Strategy A. The images transmitted by the smartphone onboard the UAV serve to complement the estimated geolocations of the victims with visual information. In this way, the estimated positions of the victims on the ground are obtained, along with zenithal images from the UAV during its flight.

4.4 Discussion and Conclusions

The proposed Internet of Cooperative Agents (IoCA) architecture enables the efficient management of SAR agents through a H2WTN and ad-hoc 5G sensors. Each agent, whether robotic or human, can connect their sensors to an FIS where the part that requires more computational calculation can be distributed among different PCs and then return the re-

quired information to the agent. The IoCA architecture extends the location awareness and productivity of the robots, and support the increase of the situational awareness of all the agents, including the emergency coordinator, who is in charge of directing the agents in the application scenario. The *SARFIS tool* integrates ROS nodes, absorbing part of the ROS-FIS.

This IoRT platform has been tested in ROUD environments under challenging conditions and evaluated in both edge and cloud computing configurations. Remote SLAM was performed using network slicing techniques, integrating the system into a pilot 5G network (without commercial clients), enabling the end-to-end architecture to be tested under bandwidth saturation conditions. A map of the operational environment of a ground robot was successfully constructed between the edge and the cloud. The separation of low and high power/data production networks into slices has enabled task prioritization by allocating 80% of the network resources to the router managing the communications of the robot as it navigates and localizes itself in the environment. This demonstrates a clear application of location awareness, allowing the exchange of access priorities to the resources of a 5G stand-alone cell, operating independently of LTE infrastructure. The throughput sharing is dynamic but prioritized for the robot executing the SLAM. The MEC center allowed to free up resources on board the robots, offering more computational capacity, remote task management, access to ad-hoc services, and data monitoring and storage. Therefore, the implementation of network slicing allows the creation of ad-hoc services, improving reliability and quality of service for different use cases.

However, challenges remain, such as optimizing latency in critical tasks and ensuring integration of additional agents and sensors into the network. Further enhancements in fault tolerance and dynamic resource allocation are necessary to fully exploit the potential of the IoRT platform in highly dynamic and resource-constrained environments.

Additionally, the architecture proved functional for positioning targets in the application scenario, integrating computationally intensive algorithms operating in a Cloud architecture. In this case, the system shows room for improvement due to the impact of communication latencies inherent in using an Android application to transmit measurements of the operational environment, which must be sent and processed by the algorithm hosted in the FIS.

Nevertheless, in the Cloud configuration, the designed positioning system could locate a victim with an error of less than 20 meters in an area of 2000 square meters within 5 minutes using Strategy A, which relies on an H-WSN composed of FTM anchors. The architecture migration to ROS 2, with Strategy B, has improved communication management and provided greater control. Strategy B, which involves using One Search Agent (OSA), has enhanced positioning accuracy, achieving an error of less than 5 meters with search times similar to Strategy A, using only a UAV.

The importance of integrating smartphones into field robotics has been demonstrated. In particular, the light weight, computational power, and communication capabilities of modern smartphones enable the development of mobile applications that integrate with ROS, combining the onboard sensory systems of various agents with the internal and external sensors of the smartphone. This is particularly useful for UAVs, as the smartphone, together with an externally attached RTK GNSS module, allows for georeferencing images (from its cameras) or associating precise UAV positions with detections from its anchor. Finally, a series of tests have been conducted using the UR2A app, which facilitates the detection of fiducial markers across various camera resolutions. The app also includes the ability to receive remote commands, allowing users to switch between cameras published in ROS, adjust resolutions, control zoom, or modify viewing angles. This flexibility not only supports current operations but also lays the groundwork for diverse future applications, such as adaptive vision systems and dynamic task configurations in isolated operational environments.

All in all, the architecture needs to evolve by focusing on the situational awareness of agents to interact with their environment more intelligently, leading to potentially more exciting applications in the context of emergencies, such as planned movements for robotic vehicles based on their perception or teleoperations for healthcare purposes in remote or isolated environments. Therefore, the efficient distribution of the data sensed by the agents can lead to an architecture based on situational awareness rather than location awareness.

Chapter 5

Situational Awareness for Control of Heterogeneous Robots

5.1 Introduction

While location awareness, as discussed in Chapter 4, focuses on determining where an agent is, situational awareness addresses understanding what is happening around the agent and anticipating how the environment may change. Does situational awareness belong inherently to an agent, or is it something the agent receives and manages? How can cooperative agents be remotely controlled as a team based on the information they share from their operational environments? A close approach can be found in nature [178], where the importance of team communication to achieve a solid cooperation is demonstrated. Similarly, in robotics, agents must share their status, influencing collective situational awareness.

This chapter aims to address these questions by exploring the control of heterogeneous robotic agents as they interact with their application scenario, leveraging an evolved version of the IoCA architecture. Two cases are presented: the autonomous navigation of a UGV triggered by sensory events and smartphone requests, and the tele-operated control of a robot manipulator with its Digital Twin (DT) integrated via a Cloud-based MQTT broker.

Therefore, this chapter addresses the third contribution of this thesis: enhancing the situational awareness of heterogeneous agents for performance in demanding application scenarios requiring soft real-time capabilities, such as SAR and telemedicine. Thus, this chapter contributes to the elements that must be necessarily included in the IoCA architecture to control robots and effectively achieve the action in the definition of robot: an intelligent link between perception and action.

Following this introduction, the chapter is structured as follows: Section 5.2 discusses the control of UGVs, which is achieved either through sensory targets triggered by LoRa EDs or via ROS services commanded from a smartphone. Section 5.3 presents the IoRT-in-hand for connecting human and robot hands, highlighting the importance of DTs in isolated scenarios where tele-medicine is needed. Finally, Section 5.4 presents the contributions and outlines the conclusions.

5.2 Triggering path-planning through Local-Edge devices

This section proposes a novel implementation of the SAR-IoCA architecture, fully integrating ROS in the FIS so that all MEC centers can be connected to the Cloud. Communications are based on 5G commercial networks. The system has been tested for remote operation and planning of a UGV in urban scenarios, including remote operation with over 400 km from the control center to the operation area [179].

Four alternative navigation modes have been developed for the UGV: i) planned navigation to a SG-defined PoI; ii) planned navigation to a user-defined PoI; iii) tele-operation; and iv) planned navigation to a PoI triggered by a request for a robot from a smartphone.

5.2.1 Design and implementation

Navigation activated by sensory targets

The geolocation of each static SG is registered by human agents, who deploy EDs, by means of an RTK receiver connected to the agent's smartphone using the UR2A application. A sensory event is triggered when a physical quantity acquired by an SG does not belong to a predefined range of values. When the sensory event is detected, the SAR-FIS tool automatically plans a path for a free UGV to travel to the position of the SG involved.

A planned navigation to an SG-defined PoI allows an agent to respond to possible risks efficiently. All UGVs are connected to the H2WTN through MQTT brokers linked to the *SARFIS tool*, which integrates an MQTT client capable of connecting to any remote MQTT broker. Thus, this team of UGVs can be planned when a magnitude associated to a risk is detected by the H2WTN, which includes the LoRa H-WSN for long range detections, and the Zigbee and BLE H-WSN for short range. The UGV with the lowest or most optimal location awareness (considering time, battery cost, computational, and communication resources) will receive the event assignment, unless the FIS operator overrides the path-planning to assign a different task to that robot. This property has been enabled for security, so that a human agent handling the *SARFIS tool* accepts or rejects the agent's autonomous movements.

Figure 5.1 shows the information flow to SAR-FIS, such as the IoRT messages coming through both edge and cloud MQTT brokers (see chapter 3), position coordinates of each agent, and images from the their IP cameras. Edge brokers are on board the UGVs. Depending on the UGV, it can host a Windows-based or an Ubuntu-based system.

For example, in an initial version, Rambler and Cuadriga tele-operation systems were managed via a LabVIEW project (Figure 5.2), where an MQTT Client and UDP communications were included to extract real-time data, such as the current of motors, wheel speed, or other data from the controller (Roboteq). However, there comes a time when the project's scalability becomes too complex and unsustainable, and its use by an remote operator and the onboard systems of the robots must be migrated to ROS.

One option is to use a ROS partition on the robot host and work on it directly, but due to software requirements, these robots have maintained their operability in LabVIEW. Therefore, integrating an MQTT client in the LabVIEW virtual instrument allows lighter information to be transmitted to a remote socket, where MQTT topics can be transformed into ROS topics using a bridge.

Thus, all the light data from the IoRT in these robots, such as GNSS data or IoT messages, can be transmitted through MQTT. Both Rambler and Cuadriga wear a LoRa CN, i.e. an

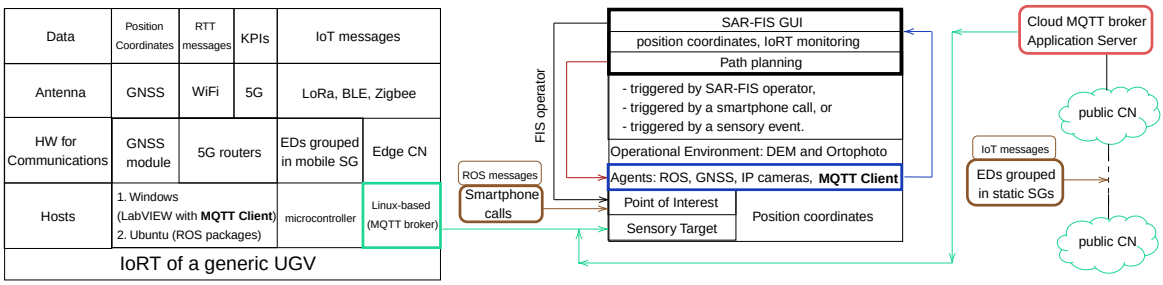


Figure 5.1 Integration of a H-WSN with a path planner in SAR-FIS.

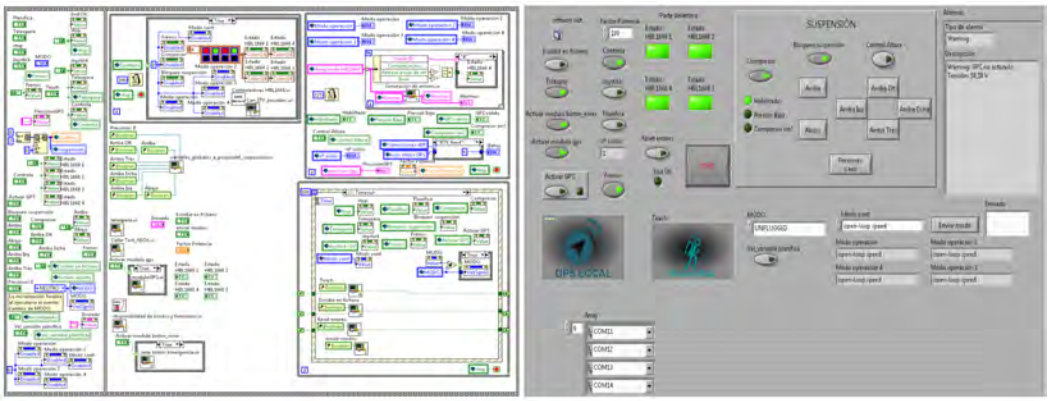


Figure 5.2 LabVIEW project in Rambler and Cuadriga

edge MQTT broker. In case, these UGVs lose their Internet coverage, their LoRa CNs maintain the data received from the LoRa EDs, including its position coordinates, thanks to the contribution about multicasting in LoRa CNs with and without Internet connectivity (see chapter 3).

SAR-FIS is used for remote monitoring of the agents (e.g., a human with a smartphone and the UGV) and path planning for UGVs in response of a request from the human agent in the operational area. In particular, the SAR-FIS tool uses PoIs as target positions for the robots. A PoI can either be defined by the operator as GNSS coordinates or selected DEM positions, or triggered by geo-referenced events.

The LoRa CNs can operate in either Edge mode or Cloud mode, which means that they are connected to an MQTT broker in the Edge or in the Cloud. Edge brokers operate in private hosts, i.e. deployed on the uncrewed vehicles (mobile CN) or in thir operational environments (static CNs), while a cloud broker can link several public gateways (static CN) with an application. Thus, agents can now interact with data through all the CNs in the application sceanrio.

As an example, if an increase in the concentration of toxic gases is detected in a particular area controlled by an SG, a UGV, whose location awareness is optimal, should inspect the surroundings of that area by executing a path planning commanded from SAR-FIS. Since the SG position is registered in SAR-FIS when deployed, the distance, slope, and path that each UGV should follow among all operatives is always known. It is then up to the SAR team leader to confirm by walkie-talkie that this resource (the most appropriate UGV for the sensory event) can be used. Based on this information, the FIS operator can allow the planned navigation. This protocol is done for safety, although our systems are fully automated so that the robot can be allowed to come to the SG position immediately when the anomaly is detected.

SAR-FIS automatically registers each SG when it receives information related to a LoRa ED. In this way, SAR-FIS updates, in real time, the sensory information coming from the LoRa H-WSN. These sensory data are monitored on the SAR-FIS GUI, together with the images captured by the onboard IP cameras.

The GUI in SAR-FIS is divided into two main areas (see Figure 5.3): (i) the operational environment map with selective monitoring of agents, sensory targets, and points of interest, and (ii) the status or the configuration parameters of environment, agents, sensory targets, points of interest, path planner, or general system.

The FIS operator can selectively enable the monitoring of physical magnitudes acquired by both static and mobile SGs. For instance, the temperature (27.7 °C) in Cuadriga's

immediate surroundings is displayed on the environment map by accesing to the data from its embarked SG.

Figure 5.3 also illustrates the status of a specific sensory target, showing information such as:

- The SG geolocation in geographic or Universal Transverse Mercator (UTM) coordinates.
- The list of sensor nodes (EDs) of each SG, showing, for each LoRa ED, the current values of: spread factor (SF), received signal strength indicator (RSSI) and signal to noise ratio (SNR), radio frequency channel used, and battery percentage.
- The list of acquired physical magnitudes, showing the current values received from the sensor network.
- The date and time of the last data received from the SG, and a time evolution chart.

The FIS operator can configure what information is shown on the SAR-FIS GUI. For instance, Figure 5.3 shows the time evolution chart related to RSSI and radio frequency channel used by sensor-nodes belonging to SG *A3FF*. SAR-FIS displays the sensory information from this SG.

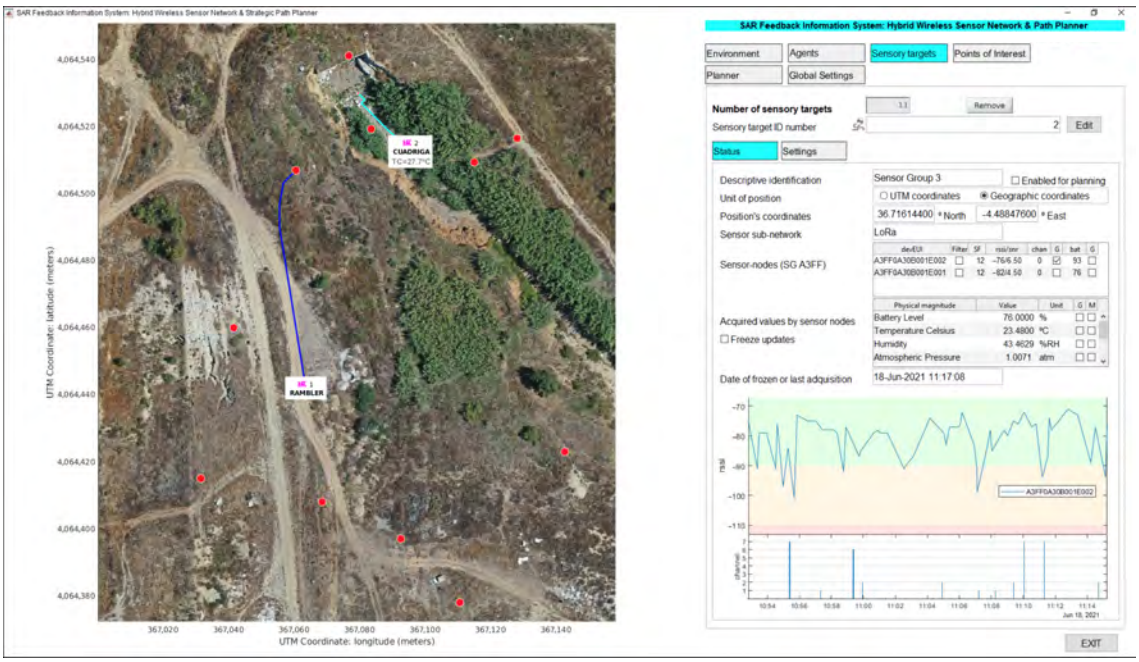


Figure 5.3 Rambler performed a path-planning triggered by this SG A3FF

To facilitate the data analysis related to an experiment, SAR-FIS saves in a data structure the information received from the LoRa H-WSN and from the GNSS devices on board the agents. This data structure is saved in a file, constituting a real dataset. The file with the generated dataset can be restored in the SAR-FIS tool for further data analysis. In addition, the ROS-FIS saves the ROS bags for every experiment.

Smartphone requests to UGVs

Now, SAR-FIS is connected to all the robotic and non-robotic agent's smartphones through ROS. Figure 5.4 shows two human agents preparing a call to Rover J8 with a 5G smartphone. In addition, Rover J8 and a DJI Matrice 600 embark a smartphone, as a 5G ad hoc sensor node, to send information about its internal and external sensors (IMU, GNSS, cameras and microphone) through the UMA-ROS-Android application (see Figure 5.5). This application allows to:

- Expand the ROS network, adding new sensor-nodes to the IoRT ecosystem, either through a LAN or via WAN, by means of smartphones.
- Acquire information from the agents' surroundings, and encourage cooperation between them.
- Interact with the operational environment, being able to call an UGV through SAR-FIS to go to an specific position, established by the GNSS module on the smartphone, at the moment the human agent requested the robot's support.

The app has two activities or screens for the user interface: the setup and the connection activities. A dark, high contrast colour gamut has been chosen for proper outdoor viewing and so that devices with OLED displays can benefit from reduced battery consumption. Two user inputs are requested: the ROS master socket (local, public, or virtual IP address and port) and the SAR agent identity, which is associated with the namespace of the ROS topics where the desired information will be published. Finally, the connect button activates the smartphone in the ROS network, being all the topics visible in any host of the architecture: Cloud, MEC centers and IoRT.

The second screen is the connection activity, where the human agent can select one or more switches to call one or more of the robots (UGVs) available in the IoRT. The list of UGVs presented is static, and includes three UGVs from our Robotics and Mechatronics Group: Cuadriga, Rambler and Rover J8 (see Appendix A). All of them can be called from the app, thanks to the total integration of SAR-IoCA in a ROS network. Figure 5.5 shows

Figure 5.4 Various agents wearing a 5G smartphone integrated in ROS

how the agent García has performed a request for the Rover J8 to be planned to the location of the smartphone, which is taken by SAR-FIS in the ROS topic /Garcia/fix. Finally, the bottom of the connection activity shows what information is transmitted from this smartphone and through which ROS topics.

Therefore, a smartphone running the application UMA-ROS-Android allows a human agent on the operational environment to request an UGV. The app turns the smartphone into a ROS node with the capability to publish information about its sensors, but also to call an UGV through SAR-FIS, where the FIS operator must take (or not) the call, depending on the SAR Team leader instructions. Moreover, the smartphone is subscribed to the status of the robots, i.e. the human agent can know which robot is available or busy to execute a path planning.

The smartphone sends position and request status through the ROS network by the ROS publishers associated to their respective topics. SAR-FIS receives this information through subscriber ROS nodes associated to the same topics. Upon detecting a change of state in the planning request of a particular UGV, SAR-FIS enables or disables that agent for path planning. In case of enablement, SAR-FIS also creates a new temporary PoI associated with the location of the smartphone at the instant of the request. Analogously to what is previously described, after executing the path planning, a list of waypoints from the initial position of

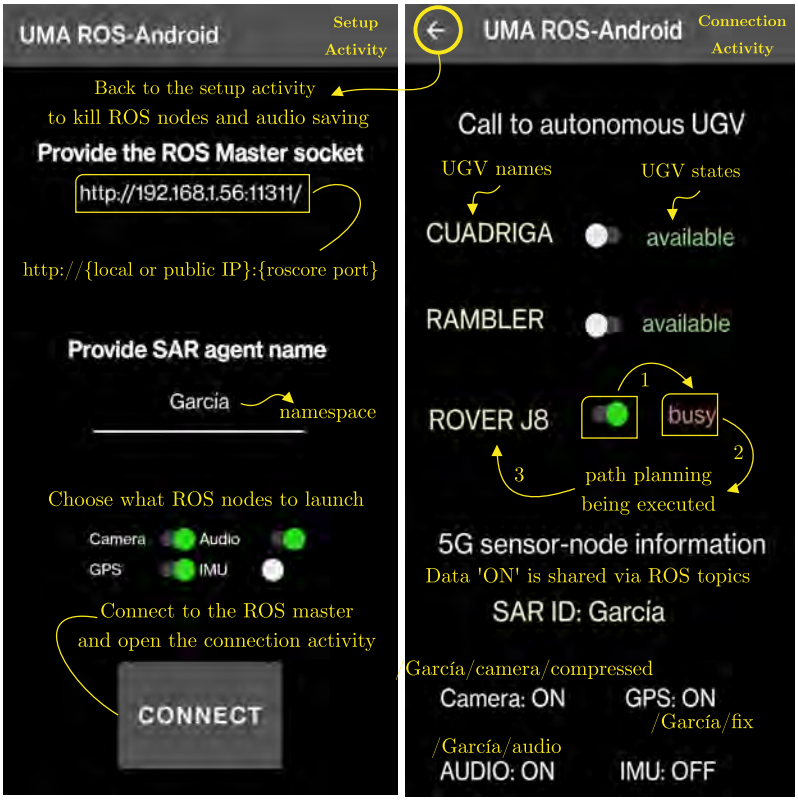


Figure 5.5 UMA-ROS-Android

the UGV to the location of the temporary PoI is obtained. Then, the path-following task is executed by the UGV.

Communications through the Internet

A generic SAR-IoCA system can have M robotic agents, N human agents, and P LANs in MEC centers, each with its own static and public IP address (SP-IPA). Thus, the maximum number of SP-IPA is $2M+N+P$. SAR teams usually work with more than one control center ($P>1$). During this thesis, we have been operating in two control centers ($P=2$): one remote to the operational environment (BCC) and another one close to it (FCC).

Thus, if we consider to use the BCC and the FCC, apart from all the agents' smartphones, the tuple (M,N,P) is $(M,N,2)$. Thus, for instance, in a typical case, we could use two UGVs, one UAV and two human agents to deploy EDs in SGs. In addition, each agent wears a smartphone. In this case, considering each agent has its own SIM card to have Internet connectivity, the tuple values are $(3,2,2)$, resulting in seven SP-IPAs, which is an affordable number for any ISP, especially for research projects.

However, it is easy for the number of SP-IPAs to grow, depending on the needs of the use case. In that case, it is advisable to use VPN and port reconfiguration, as explained in chapter 4. This is not a problem in ROS 2, but in ROS 1 the distributed communications are more complicated.

5.2.2 Experimental results

In this section, we present the ROS architecture implemented to link the IoRT with SAR-FIS in a real use-case. The experimentation part took place in Avila (Castile and Leon, Spain), at the facilities of the Spanish National Police, 440 km away from the BCC, which was again at the facilities of University of Malaga. An FCC was installed at visual range of the operational area of the robot, allowing also planning, monitoring and tele-operation of the robot. In this case, communications between the robot, the BCC and the FCC used a commercial non-stand alone (NSA) 5G network. The goal was to move the robot according to a path planned remotely from the BCC, focusing in the behaviour of the architecture. First, in order to plan paths for the UGV, aerial mapping is performed to obtain a Digital Elevation Model (DEM) and an orthophoto of the operational environment. The chosen area of operations was fairly flat for security reasons. In addition, the area was fairly clear of obstacles for the robot to take a simple trajectory when called by the smartphone. In this case, the idea was to visually demonstrate the error caused by the smartphone's internal GNSS module.

In this case, the path was planned to answer a request for the robot from the UMA-ROS-Android app. Figure 5.6 shows the different ROS 1 nodes connected between the IoRT and MEC centers for remote planning and operation of a UGV through the evolution of the IoCA architecture.

Four prerequisites have to be fulfilled: (i) end-to-end data traffic between two mobile cells towers far apart, (ii) the integration of the UGV in a ROS network, (ii) the generation and processing of a DEM of the area of interest to be explored by the robot, (iv) and the test of the system in a real scenario.

To integrate the UGV in the SAR-IoCA architecture, some hardware elements have to be added to the robot. These include an Intel NUC8i7BEH minicomputer with Ubuntu 18.04 and ROS [180], which communicates with both GNSS receivers through different Universal Serial Bus (USB) ports, and is connected to the Internet through a 5G router providing Ethernet connection.

Thus, it is possible to get differential correction data via the standard protocol Networked Transport of RTCM via Internet Protocol (NTRIP) through regional public positioning networks [181]. The Arduino Leonardo is connected to the robot's PC, and simulates an Xbox controller through the Arduino Xinput library.

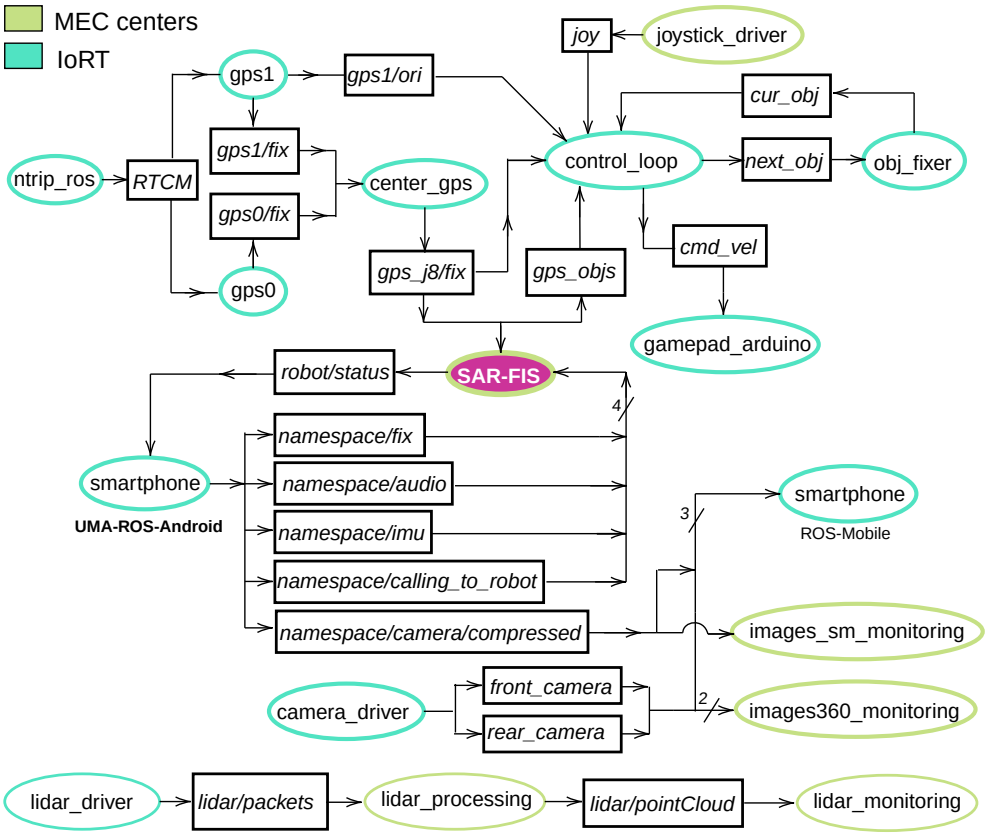


Figure 5.6 ROS 1 network connected to SAR-FIS

The integration of the Rover J8 inside the SAR system has been implemented through a series of ROS nodes, such as drivers to publish the information gathered by the onboard sensors (e.g. cameras, LiDAR and GNSS) or to allow a human agent to tele-operate the UGV from a control center.

5.2.3 Evaluation

A demonstration of the evolution of the IoCA architecture is presented, where a single robotic agent (Rover J8), a human agent, including two smartphones, and two control centers (FCC and BCC) were used. Thus, the SP-IPA value is 5. In this case, the UAV flight was performed offline to generate the DEM, before the real-time experiment between Málaga and Ávila.

The MEC centers (BCC and FCC, in our experimental setups) are made of different hosts, each one with various purposes related to the ROS nodes they host. Two LANs have been designed, with a pair of hosts, i.e. PCs, each one, intended to reduce the data flow coming from the IoRT and the cloud. Sharing tasks between available hosts in the MEC centers and in the cloud has improved the efficiency of the SAR-IoCA architecture (see Table 5.1).

Table 5.1 Sharing tasks between the Edge and the Cloud

LAN-1 (FCC)	**LAN-2 (FCC)**	**SAR-FIS PC (BCC)**
SAR-FIS replica	DEM generation	DEM processing
BCC monitoring	Control of UGV mode	ROS-SAR-FIS integration
UGV monitoring	UGV monitoring	Generation of paths
Creation of datasets (ROS bags)	UGV tele-operation	Dataset saving

The data rate and throughput via the ROS topics have to be studied and predicted before the case-study implementation to ensure a good performance. For this purpose, different port forwarding techniques have been implemented. A straightforward way to achieve this over the wide area network (WAN) with ROS is using a single PC per LAN and enabling demilitarized zone (DMZ). It is necessary to have an SP-IPA. This configuration has been prioritised for the single LAN in the cloud, which hosts the MATLAB-SAR-FIS ROS node, and also one of the LANs on the edge, which hosts the ROS master node.

Four 5G routers (Huawei 5G CPE Pro 2 H122-373) are distributed in the network: one for the UGV, one at the BCC and two at the FCC. Only one PC on each LAN is associated with the SP-IPA of its corresponding router. This PC is known as the main PC (MPC). The rest are the secondaries (SPC). In addition, two smartphones (Huawei P40 Pro 5G), with an SP-IPA, operate in the IoRT.

Each MPC is known for the devices in other LANs in the ROS network, registering their hostname in the rest of devices over the ROS network. Thus, active ROS nodes in SPCs cannot publish information for other devices of the WAN. Then, their corresponding MPCs have to republish this local information in another topic for the WAN.

The UMA-ROS-Android app has been integrated to give a human agent the option to call a robot [43, 182]. In response, SAR-FIS plans the robot path to the smartphone user's location. Thus, a smartphone can be used by the SAR robotic agent to come to the aid of a human agent.

A second smartphone is onboard the UGV to monitor the robot's environment with audio, video, location, and orientation. Finally, the IoRT (see Figure 5.7) is completed by the sensors onboard the robot, namely two 3D LiDAR systems (Velodyne VLP 16), two 180° field-of-view (FOV) cameras (rear and front, Axis F1035-E), two GNSS receivers (u-blox ANN-MB1) and antennas (Hetronic GK-442 TNC-Plug and two Southwest 1001-085), and the smartphone sensors.

Figure 5.8 shows the UGV last position after the call of a human agent through its smartphone (left) and the executed autonomous path from the *red square*. The global

planning was performed towards a temporary PoI by the request made from the smartphone carried by a human agent (*cyan square*).

The resulting path (*red dotted line*) represents the waypoint list to be crossed by the UGV until reaching the target point, thus avoiding static obstacles present in the environment, according to the information contained in the DEM of the operational area. The robot performed the task successfully, and repeatedly. However, the final position of the robot differs by 4 metres from the static position of the human agent, due to the positioning error of the smartphone's internal GNSS module. Finally, tele-operation was also performed from the FCC, transmitting commands and receiving video through 4G in FCC and BCC. LiDAR data was also available at both the FCC and the BCC.

Figure 5.7 UGV Rover J8 in the experimental urban area (left) and onboard sensors (right)

Figure 5.8 Path planning for a smartphone request (*cyan square*)

5.3 The IoRT-in-hand: tele-robotic echography and DTs on Cloud-Edge devices

This section presents a novel application for healthcare delivery to geographically isolated populations in ROUD scenarios. The proposed system remotely connects robot and specialist hands by means of an IoRT-in-hand, i.e., a smart end-effector, and a mobile device, e.g., a smartphone, respectively.

The IoRT-in-hand is introduced as a Cloud-Edge device within the IoCA architecture, designed to transmit information over the Internet to remote environments. Its goal is to provide a lightweight, easily attachable smart end-effector compatible with compact robotic arms using standard couplings. The IoRT-in-hand integrates a medical instrument, an RGB camera with dual servos, a force/torque sensor, and a mini-PC with Internet connectivity.

An open-source Android application is also proposed to tele-operate the IoRT-in-hand. This application incorporates ROS nodes and MQTT clients to enable remote manipulation tasks, providing operators with control and visualization capabilities in physical and virtual robot workspaces. The architecture includes a DT hosted in the FIS, allowing remote users to access 3D environmental data from the RGB camera and interact with the DT on a server.

The presented implementation comprises several interconnected products (see Figure 5.9), which have been developed and implemented to cover a specific tele-medicine operation: an ultrasound scanner.

5.3.1 Design and implementation

Requirements for the tele-operation system

The challenge is to touch a human safely with a robotic arm in non-engineered or even out-of-hospital environments, and particular specifications must be met in four areas: end-effector standardization, tele-control based on mobile devices, architecture for the Edge-Cloud continuum, and sensory feedback to the remote operator.

The instrumentation for the end-effector must be designed to be lightweight, portable, and standardized so that it can be carried to the application scenario to be attached to the robotic arm with agility. EN ISO 9409-1 [183] refers to a specific standard for mechanical interfaces of industrial robots, focusing on the size and type of coupling (flange) used to connect tools or end-effector to a robotic manipulator. Tele-medicine use cases require end-effectors suitable for the vast majority of Compact Robotic Arms (CRA), such as, among others, the UR3e (3 kg payload), UR5e (5 kg payload), UR10e (12.5 kg payload), and UR16e (16 kg payload) models from Universal Robots (Odense, Denmark), the LBR iiwa 7 R800 model

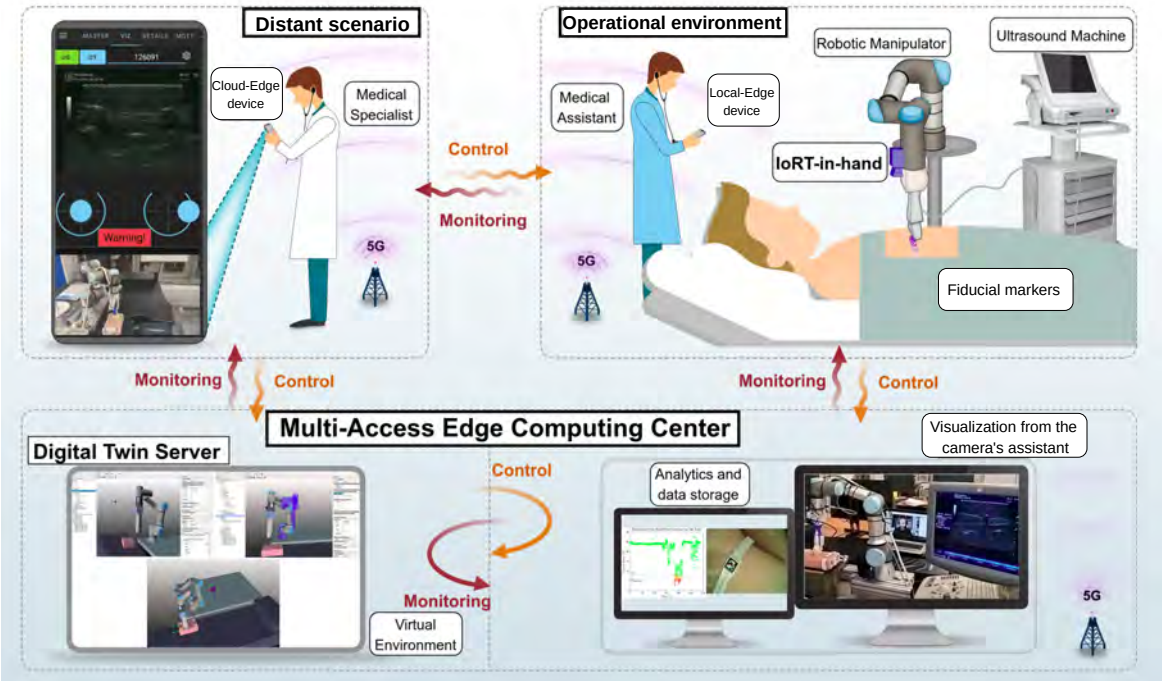


Figure 5.9 An IoRT-in-hand for a medical application scenario

(7 kg payload) from KUKA, the Motoman HC10 model (10 kg payload) from Yaskawa, or the TM5 model (7 kg payload) from Omrom. This kind of CRA can be helpful inside an operating room [184] or, in an outdoor disaster scenario, embarked on a large UGV [185], thus acting as a mobile manipulator.

The IoRT-in-hand is a smart device to be placed in the end-effector, which includes a Local-Edge device, i.e., an entry point for information from the operational environment, with processing capability and Internet connection, which can be embedded in a mechanism. As a detachable smart end-effector, the IoRT-in-hand has to be lightweight (3 kg maximum) to allow not only controlled movement of the CRA, but also to be portable. In addition, this end-device should access to the Internet quickly once it is attached to the CRA and switched on.

The IoRT-in-hand device consists of a 3D-printed housing designed to accommodate the necessary components for performing tele-robotic ultrasound examinations (see Figure 5.10). It includes a compact RGB camera equipped with a torch for subject illumination, two servos to control the camera's pitch and yaw angles, and an ultrasound probe for acquiring echographic images. Additionally, a force–torque sensor is integrated to measure the interaction between the robotic hand and the subject, while a Local-Edge computing unit (a mini-PC) handles image processing and transmission to the remote tele-operator.

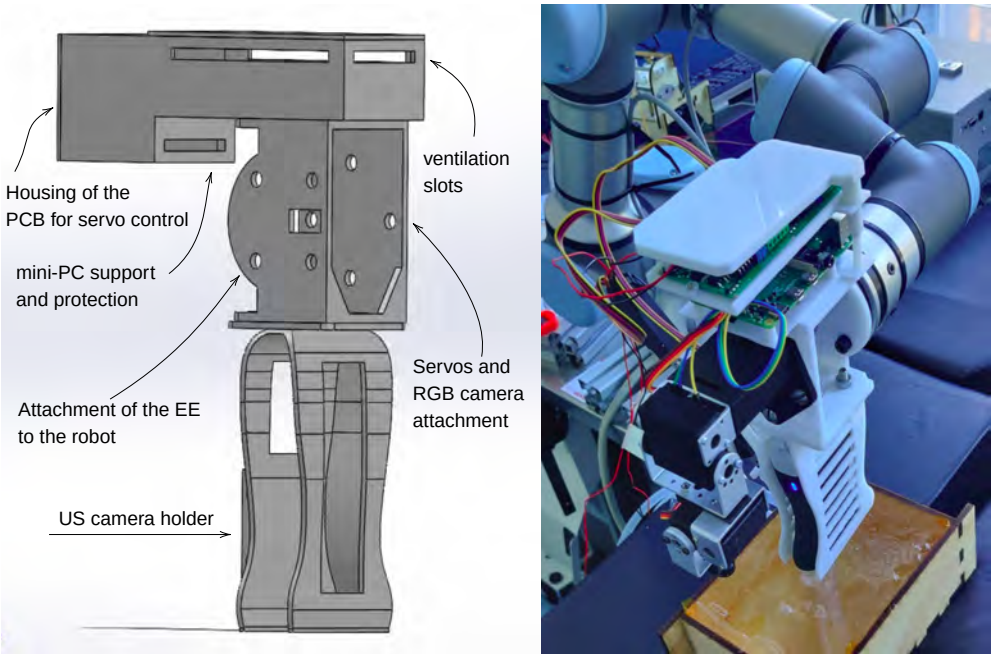


Figure 5.10 Smart end-effector design

The tele-control is performed from a Cloud-Edge device (e.g. an smartphone or tablet) without needing accessories, such as smart glasses, so that any remote physician can quickly control the robotic system without engineering means. Another need is stable tele-communications with low latency and a valid bandwidth to transmit images to the operator. Thus, both ends of the network must able to connect to a network with at least 5G capability, so that low latency and bandwidth that allows the transmission of medium resolution images, without cuts, are guaranteed.

In addition, depending on the type of the operational environment, other elements could be required in the robot workspace, interacting with the IoRT-in-hand. Thus, the visualization, including augmented reality and control, has to be integrated into an MEC center, accessible from the smartphone, allowing the operator depth perception and efficient control. In this work, ArUco markers [186] are included in the system as connected products since they are linked to Internet once they are detected by the IoRT-in-hand. These fiducial markers are helpful not only to detect and estimate the relative pose of the objects in the robot's environment, but also to apply augmented reality, linked to the DT [187]. Finally, the novelty presented here is to use this virtual information to enable the tele-operator to understand the robot's workspace three-dimensionally. The challenge is to display both the virtual and real environment in the smarphone screen in order to show to the tele-operator a 3D perspective of the robot workspace. Due to the low-latency needs, a single ROS 1 network, i.e., managed by a ROS master in the cloud, governs the architecture. Moreover, we include the possibility

to move the robot and DT through an MQTT broker, hosted also in the Cloud. Thus, the smartphone has access to the robot and DT through the Internet, via MQTT and ROS.

System architecture

Once all the end-devices are on a detachable mechanism following the EN ISO 9409-1-50-4-M6 standard, it is necessary to interconnect them with the Cloud-Edge device controlling the system remotely and with the MEC center hosting the DT server.

The system architecture (see Figure 5.11) favors the cooperation, in real-time, between the human and robotic hands, that are interconnected through ROS and MQTT. Therefore, the architecture enables the digitalization of healthcare procedures and remote healthcare solutions. Medical doctors are a valuable resource, and it is important to avoid putting them in catastrophic scenarios such as wars, terrorist attacks, or natural disasters. Therefore, the primary objective is to enable remote performance of medical procedures in scenarios where specialists are scarce, or cannot be physically present due to time constraints, or risks to their safety.

Feedback Information System

Among the services to be made available to the tele-operator, the following should be highlighted:

- Access to the visualization of the DT, in real time, including the visualization of virtual points associated with the ArUco markers, so that it is possible to map not only objects in the structured environment in the robot's workspace, but also parts of the body of the subject to be treated. For this purpose, the subject's own body can have some reference markers placed on it, either as bands, so that if the person moves, this information is visible in the virtual environment.
- Access to sensory information connected to the IoRT-in-hand, including RGB images or forces exerted by the end-effector.
- Ability to command Cartesian velocities to the end-effector.
- Ability to visualize the ultrasound images, in real time, to facilitate the remote diagnosis.
- Ability to return the robot to safety poses at any time, being able to visualize the trajectories previously in the virtual environment. To do this, the developed Android application allows the DT to be controlled by MQTT commands.

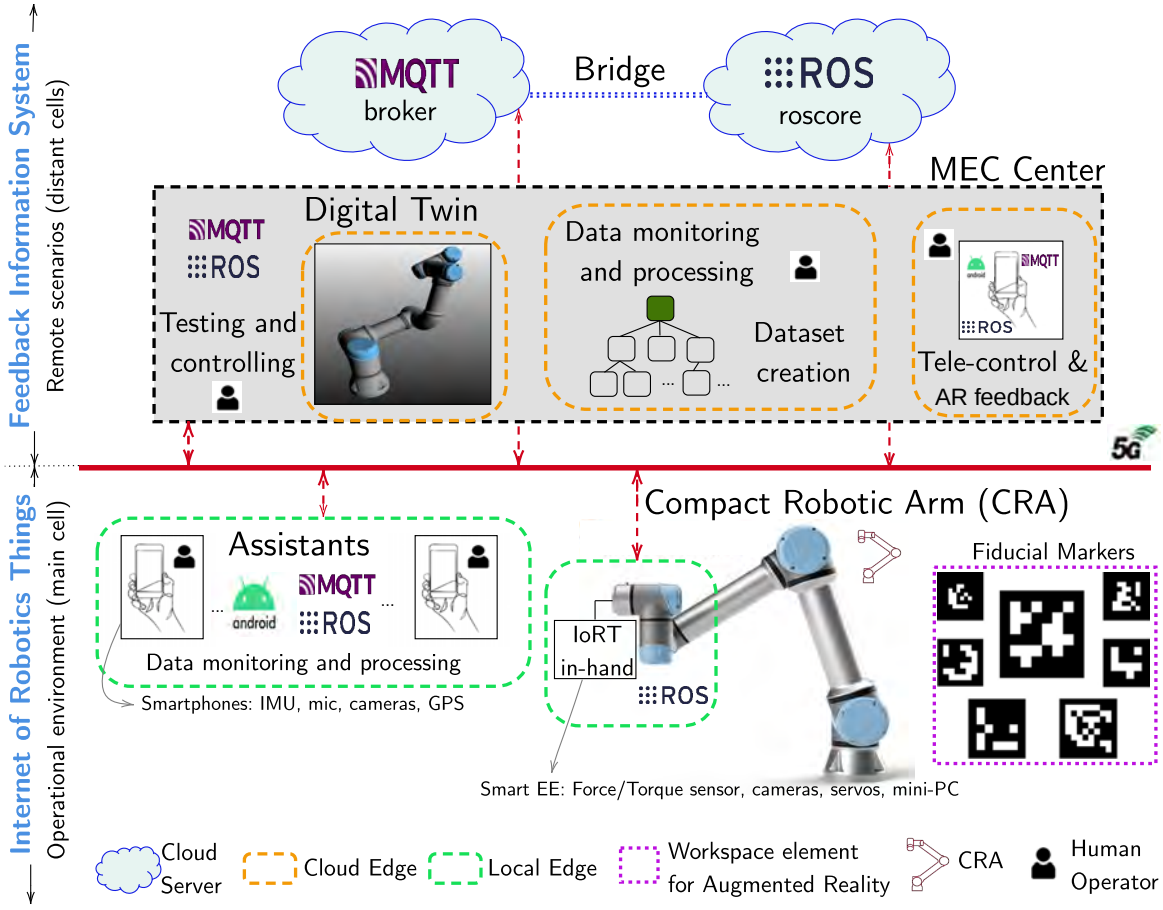


Figure 5.11 Edge-Cloud architecture for tele-robotic manipulation

All these services must be able to run in the FIS, i.e., in Cloud-Edge devices, that can be placed in remote scenarios (e.g. in another country).

The proposed architecture distributes the tasks among different hosts through the fog to be able to manage and control the IoRT-in-hand that must have a pre-established configuration, as the idea is that once it is in place, it should not have to be set up locally. Therefore, the ROS master node and the MQTT broker are hosted in the Cloud, regardless of their physical location. On the other hand, the DT server is hosted in the MEC center that can manage the entire network. Connections are made through a VPN managed from the MEC center itself. Testing, networking, control, and monitoring tasks are executed as a whole in the FIS. The location and Internet service for the Cloud-Edge device that will control the robot can be different from those of the DT server.

Thus, the FIS covers all tasks hosted on dedicated computers located on remote cells: data monitoring, processing, remote control of the CRA, and data-set creation. Among these Cloud-Edge devices, a DT server allows to emulate the movements of the CRA with its DT

in a virtual environment. Moreover, it is possible to synchronize both the CRA and the DT, whether the control is autonomous or tele-operated. Then, a remote human operator can see the movements of the CRA in real-time by getting the virtual environment images from the DT server.

The Internet of Robotic Things

Within the main mobile cell, where the CRA has to perform the medical application, there are two classes of edge devices: the IoRT-in-hand, and the smartphones carried by the medical assistants. Thus, smartphones can have different roles within the IoRT or the FIS, depending on whether they act as information providers or remote processors and task monitors, respectively. An Android app, based on the ROS-Mobile app [188], has been developed[1] to integrate the capabilities of MQTT clients and ROS nodes on a smartphone. Thus, the smartphone screen serves as a user interface for communication between the CRA and the DT. The screen displays both the images from the camera in the smart end-effector and the DT server, simultaneously providing two perspectives to the tele-operator. Specifically, the bridge in the cloud allows the tele-operator to synchronize and control, in real-time, either the DT or the CRA from the app. Thus, it is possible to move the DT without seeing the physical robot, which allows the tele-operator to improve their skills. Moreover, any assistant using their smartphone can access to the images from the virtual environment, i.e., the DT, and the onboard RGB camera (eye-in-hand).

The IoRT includes sensor networks distributed between the CRA, the assistants and their workspaces, such as cameras, microphones, physiological and industrial sensors, GNSS, as well as other kinds of elements to include augmented reality capabilities. Fiducial markers, such as, ArUco markers are tipically used to estimate the distances between the end-effector and the objects within an structured environment. Moreover, we use them to load extra virtual points around the DT, hosted on the MEC center.

All the devices distributed between the edge and the cloud have processing and storage capabilities, and are connected to the CRA controller and to the IoRT-in-hand through a VPN, being able to exchange information between the operational environment and the FIS [43]. The most important information from the operational environment is related to the CRA parameters, such as joint values, end-effector position, force and torque intensities on the tip, processed images from the cameras, and transformation matrices.

[1] https://github.com/Robotics-Mechatronics-UMA/Adhoc-ROS-Mobile

Interoperability

For the safe movement of the CRA in the human subject's environment, the telecommunications must be low latency and high bandwidth, characteristics that can be met using a 5G network. Thus, a 5G network is recommended to establish connections between distant edge and fog devices. The challenge is to move the robot from a distance of more than 2000 km and touch the human with the IoRT-in-hand, which requires smooth, lag-free control. In addition, the video stream from both cameras on the robot's hand cannot have a lag associated with the network and processing. In order to guarantee the lowest possible display latency, the detection and processing of the poses of the referenced objects is done at the IoRT-in-hand. The result is sent in real-time to the DT server, so that the tele-operator can see virtual points associated to the referenced objects in the robot workspace. Data communication, such as position, orientation, and transformation matrices, are computed in the edge-device, and sent to the DT server (VEROSIM) through MQTT, as it is a specific protocol for lightweight messaging and very extended in the IoT world.

In addition, the MQTT Extension for VEROSIM [189] allows the external client integrated in the tele-operator smartphone to interact with the DT, being possible to restrict access to other external clients for setting property values. This MQTT plugin is also used to exchange between the virtual environment and the physical world through the detection of the ArUco markers. Both the physical robot and the DT can be controlled through the Android app, which integrates external MQTT clients and ROS nodes that implement the ROS-MQTT bridge, making use of the timer-based and event-based publishers and the subscriber. This way, a mobile device, like a smartphone, can control both the DT and the CRA. The ROS master node and the MQTT broker are hosted in the cloud, being everything interconnected through a Virtual Private Network (VPN) based in Zerotier to disable the Network Address Translation (NAT). Thus, every host is assigned an IP address within the same virtual network, being accessible through ROS [179] and MQTT. In addition, the architecture requires the routers' ports for cameras and DT application to be open.

Figure 5.12 summarizes the followed interoperability scheme to run the tele-operation system. First, a medical assistant puts the IoRT-in-hand on the robot end-effector and switches it on. After that, they calibrate the RGB camera and robot [190] if necessary. Usually, it is only required for the first use of a specific workspace.

Then, the IoRT-in-hand gets information about the robot interactions with its surrounding, such as, RGB images, ultrasound images, forces exerted on the end-effector, or ArUco markers detections. Once the fiducial markers are detected on the objects within the robot workspace, the estimated poses for them are computed, sending them to the DT server

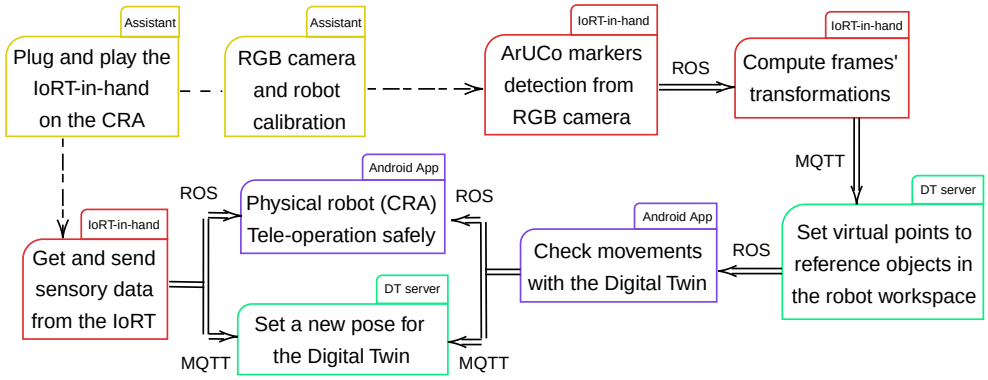


Figure 5.12 Interoperability

(running VEROSIM) to be represented as virtual points. Finally, it is necessary to check the movements with the DT, before tele-operating the CRA safely.

At the same time, sensors generating higher data traffic, such as cameras, are routed to ROS topics. The MQTT-ROS bridge [191] transfers the information between ROS and MQTT topics, which provides redundancy when storing data and also makes information accessible. Regarding the virtual environment image, this is sent through a streaming of the screen on which the DT is monitored locally, transmitting the RGB images processed locally via TCP/IP.

5.3.2 Robotic tele-medicine implementation

For illustration and validation, we present a case study in which a robotic manipulator is controlled in real-time to perform a remote ultrasound scan. Next, the three major parts of the architecture are described below: the operational environment within the main cell where the CRA is; a remote cell, where the smartphone used for tele-operation is considered a Fog-Edge device; and the FIS, where several fog-devices coexist by exchanging information with the application scenario (operational environment and Cloud).

Application scenario setup

First, it is necessary to prepare the subject on a stretcher, and perform the eye-in-hand (RGB camera) and robot calibrations. In principle, this only needs to be done the first time the IoRT-in-hand is placed on the end-effector of the CRA, unless environmental conditions change drastically, and re-calibration of the camera is necessary to ensure the accuracy of the estimates. Thus, camera calibration is needed to obtain its parameters, and then be able to apply AR, through the ArUco markers.

For camera calibration, the implemented methodology is based on chessboard calibration images, where it is necessary to obtain all the corner points inside the chessboard, discarding the outer ones. The process involves capturing, at least, 10 images of the entire chessboard, taken them in different planes to avoid degenerate case, from different angles and positions, detecting the corners of the images, and then using optimization algorithms to find the camera parameter values that best fit the observations. Then, for every corner point, a relationship between image and world points is achieved. The mentioned parameters allow the camera to map real-world $3D$ points to $2D$ points in the image.

For guiding the robot to the target body surface, two straps with ArUco markers are adjusted to the subject (see Figure 5.13a, where a phantom is used for testing). The rest of the workspace is mapped using a tip on the end-effector for the robot calibration. The virtual environment has been emulated by referencing objects in the robot workspace, such as the bed, the robot table and the relative distances to the objects. Every detail has been taken into account while creating the virtual model and DT in VEROSIM.

Once the IoRT-in-hand is switched on, it is connected to a wireless network whose credentials are saved in the edge-device configuration. After that, the ROS package is launched automatically, since it is registered in the bash file. For this prototype, a mobile network is the Internet source of the whole testing setup.

Regarding the rest of the IoRT, it includes the cameras worn by the medical assistants, being the video stream available in the smartphone of the tele-operator via ROS.

Each time an ArUco marker is detected, its pose is sent to VEROSIM, via MQTT, in order to establish a virtual point with the position and orientation referred to the object, with respect to the camera's reference system. In this way, the distances between the body to be scanned and the ultrasound camera are estimated. Moreover, the feedback provided by the force sensor gives the tele-operator an idea of whether or not he is touching the subject.

The selected CRA is the UR3 manipulator, which offers a work envelope of 500 mm. All joints exhibit a range of motion of $\pm 360^\circ$, with the exception of the final joint, which allows for unlimited rotation. The three wrist joints operate at an angular velocity of $360^\circ/\text{s}$, whereas the first three joints are limited to $180^\circ/\text{s}$.

Typical velocity requirements for robotic-assisted medical procedures fall within the range of 2–170 mm/s [192, 193]. Nonetheless, for teleoperated interventions, the maximum translational speed has been constrained to 30 mm/s to ensure operational safety.

The CRA is interfaced through a real-time data exchange protocol, implemented via the `ur_rtde` library[2], as described in [194]. Accurate spatial calibration of the CRA relative

[2] https://sdurobotics.gitlab.io/ur_rtde/

(a) The IoRT-in-hand (b) Smartphone app (c) Real-time estimation of poses

Figure 5.13 Testing setup: The tele-robotic echography is tested on a phantom, being the DT synchronized with the CRA. The transformation matrices are calculated on the edge-device, and poses estimated are sent via MQTT to the DT server to display the virtual target points for the CRA.

to the subject's bed is performed prior to operation, enabling the initialization and full structuring of the virtual environment within DT framework.

A custom-designed edge device is mounted at the CRA end-effector, integrating the following components: a six DOF force-torque sensor (OptoForce, Hungary), a local edge computing unit, a servo-actuated RGB camera, and an ultrasound (US) probe. The edge computing platform is realized using a Raspberry Pi 4B. The RGB camera captures images for local ArUco marker detection, which facilitates real-time estimation of the CRA-to-subject relative pose.

The servo stage provides active control of the camera's field of view. Concurrently, the edge device performs acquisition of ultrasound images and continuous force monitoring. Specifically, vertical force measurements are evaluated in real-time, and an alert is transmitted via MQTT protocol if the exerted force exceeds a predefined safety threshold.

In addition to the applied force, the relative position is also streaming to the FIS via MQTT in real-time, thus being able to estimate and visualize the position of the ArUco markers both in the RGB image and in the virtual environment (see Figure 5.13). Meanwhile, data generated by the physiological sensors on the subject are transmitted via MQTT if they are lightweight enough to be sent efficiently. Here, we use a wearable device based on a photoplethysmography (PPG) sensor and an ESP32 board to monitor heart rate and transmit this data in real time.

Regarding the data flow exchanged through subscriber/publisher ROS nodes, there are the images captured by cameras worn by the medical assistants, the values of the robot's joints and pose messages, as well as the warnings sent to the tele-operator based on the readings of the force sensor located at the tip of the robotic arm. In addition, the virtual joysticks (on the smartphone screen) with which the tele-operator interacts send the speed commands to the robot controller to move it in the Cartesian coordinate system.

The operational environment (see Figure 5.13) consists of a bed on which the subject would lie down. The local reference system of the ultrasound camera has been drawn in Figure 5.13-a, being the Y-Axis parallel to the bed. A table on which the CRA is placed is close to the bed. The region of interest for ultrasound scanning is defined by two straps with ArUco markers and is covered by the robot workspace.

Finally, the TCP/IP protocol is used to transmit images taken by the RGB and ultrasound cameras, and the virtual environment where the DT is hosted. The necessary ports are enabled on the routers of each network end.

Remote cell

An HA, usually an MD, tele-operates the CRA from a remote cell using a smartphone. For this purpose, the ad hoc ROS-Mobile is designed with two displays. The user can select to view video streaming hosted on the web, including IP camera streams and ROS images from other smartphones [179]. The top display can show either the ultrasound imaging or the DT scenario, determined by the two buttons on the left upper corner (see Figure 5.13-b). The bottom one displays real-time ROS images from any RGB camera, such as those on assistants' smartphones. By this means, we aim to facilitate the control of CRA for the tele-operator. In addition, the app allows the user to control the CRA movement through two virtual joysticks. A dead man's switch is enabled for both joysticks to automatically return to zero when the user stops touching the mobile screen.

Moreover, a new ROS subscriber node is created to warn the user about the pressing forces that the end-effector exerts on the subject. The pressing force is categorized into three levels and represented by different colors (green for safe, yellow for caution, and red for dangerous) to signify the potential discomfort that the subject may experience.

The local-Edge device onboard the CRA hosts a ROS publisher node which sends a value of type UInt8 (0, 1, or 2) depending on the measured forces on the ultrasound probe. Two limits are established according to the subject's pain. The color risk (green, yellow, and red) associated with the measured pressure on the tip is shown on the smartphone screen thanks to the new ROS subscriber (see Figure 5.13-b) running in the ad hoc ROS-Mobile app. In addition, the app includes a new tab associated with MQTT clients hosted on the

user's smartphone to allow information exchange with other MQTT clients in the tele-robotic system.

Digital Twins and Augmented Reality for the FIS

The FIS consists of three MEC centers: a health center, a data center, and a server running the DT.

In the health center, the HA is typically an MD whose role is to supervise the tele-operation and provide diagnosis suggestions by watching the stream-back ultrasound imaging and the subject's physiological signals.

The data center collects and saves the data from both the remote cell and the operational environment. Specifically, data generated during the experiments are saved in a rosbag. MQTT topics and images transmitted via TCP/IP[3] are converted to ROS topics before being saved.

The DT runs in VEROSIM, a virtual testbed implementation for 3D robot simulations [195]. It is constructed in a cloud/edge server where an HA (typically an engineer) is responsible for supervising the procedure. The DT receives MQTT messages from the remote cell for synchronizing the posture between the CRA and the DT. By this means, the DT enables the HA in the remote cell to assign multiple waypoints and conduct a dry run in the virtual environment. Also, an interface based on Deskreen[4] is designed for capturing the DT display and streaming it via TCP/IP. This allows the HA in the remote cell to visualize and gain more insights into the CRA movement from the virtual environment in a different perspective.

5.3.3 Experiments and results

Two types of tele-medicine exercise experiments were conducted for the medical implementation of the proposed tele-robotic system. The first type was designed to characterize the system based on a LAN, where multiple local tele-operations (LTs) were carried out to test the different parts of the system. In the second type, a series of remote tele-operations (RTs) were conducted between two laboratories located in Málaga (Spain) and Odense (Denmark), with a VPN established between the two endpoints of the network. The end in Málaga used a 5G network, while the end in Odense used a 4G-level network. Figure 5.13 illustrates the application scenario for both types of experiments, where the tele-operator was the same person. The aim of these experiments was to measure five key parameters to evaluate

[3] https://motion-project.github.io/

[4] https://github.com/pavlobu/deskreen

the feasibility of using our IoRT-in-hand in practical applications, including forces on the end-effector, the number of warnings given to the tele-operator, time to do the tele-medicine, time in contact with the body, and latency between the network ends.

Evaluation metrics

A well-designed tele-operation architecture should provide an intuitive user experience, reducing the occurrence of operational errors. Taking the complex anatomical geometry into account, controlling tele-operation remotely can be challenging for operators. The efficacy of a tele-operation can be measured by the ability to quickly locate the target without exerting excessive force on the subject's body during the scan, which can be quantified by the percentage of time spent performing the scan (τ) compared to the total time (t_t).

$$\psi = \tau / t_t \tag{5.1}$$

In addition, we define a penalty term Ω to include a weighted relevance on the number of warnings triggered due to the exceeded pressing force on the body during tele-robotic echography. Two warning thresholds are pre-set to indicate the maximum allowable force exerted without causing the subject discomfort (yellow alert) or pain (red alert). Specifically, Ω is defined as follows: a red warning counts four times higher risk than a yellow warning.

$$\Omega = 4 \times \omega_R + \omega_Y \tag{5.2}$$

The final performance index is defined as φ:

$$\varphi = \alpha \times \psi - \beta \times \Omega / (1 - \psi) \tag{5.3}$$

where α and β are the weights for balancing the operational efficacy and the penalty due to excessive pressure. After some practical considerations, the values of α and β are chosen to be 230 and 0.25, respectively. This is not intended as a precise mathematical expression but as a way of measuring the quality of tele-operations in percentage: $\varphi \in (0, 100)[\%]$. It is assumed that an MD performing a scan manually should achieve a 100%.

Experiment 1: characterization of a local implemented system

The aim of Experiment 1 was to test the tele-robotic system on the same LAN network, except for a dedicated computer representing the cloud and the smartphone used for LTs, which utilized a 4G network. Then, the agents and MEC centers were connected to a WiFi-enabled router, being the LAN connected through a VPN to the smartphone and cloud, which hosted

the ROS master and the MQTT broker. The visualization of the DT, as well as the cameras, worked smoothly, and the tele-operator was able to perform the LTs from a smartphone using its own mobile data service. The app was getting the information through the VPN. With this setup, the maximum end-effector velocity associated with the joystick fully depressed was 30 mm/s for the safe use of the RA.

Table 5.2 presents the results from 4 consecutive LT runs (LT1, LT2, LT3 and LT4). There was an improvement in the φ as the tele-operator gained experience with the app. LT1 and LT2 had a similar duration and φ. Subsequently, the same HA performed LT3 and LT4, reducing the t_t and significantly improving the φ, which was close to 100%. After these runs, the tele-operator felt confident working at speeds even higher than the operation limits set in the joystick app configuration.

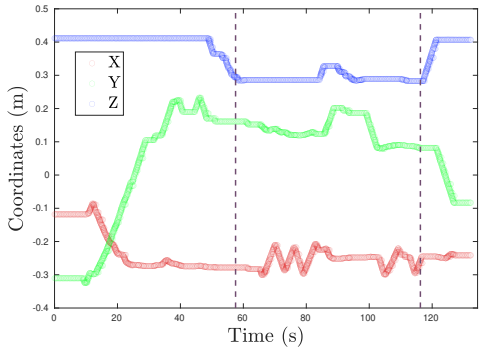
Figure 5.14 shows the end-effector's trajectories for the CRA during LT3. Before scan time, the end-effector lost height (in blue) as it approached the phantom. During the scan time, two zig-zag sweeps were performed (see Figure 5.14-a), as seen in the X-axis movement (in red). At the beginning and the end of these two scans (i.e., when contact started), some over-pressure occurred (see Figure 5.14-b), which triggered warnings on the smartphone's screen, as indicated in Figure 5.14-c. The start point in Figure 5.14-c and d represents the start of the RA's movement, while the endpoint indicates a safe landmark for the subject, being chosen by the tele-operator as a good moment in the return trajectory to stop the RA.

Experiment 2: Remote tele-operation on a volunteer

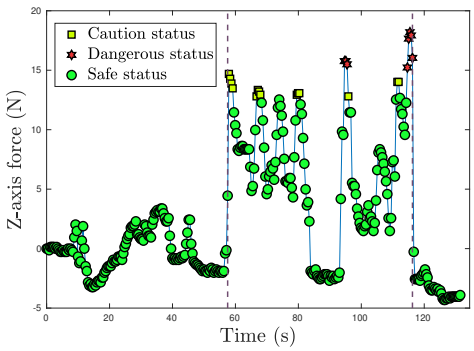
The second type of experiment was conducted between two distant locations to evaluate the tele-robotic system performance. In these experiments, the tele-operator and the cloud were in Málaga, while the RA, the DT, the subject and the assistants were in Odense, 2300 km away. A series of RTs were conducted aiming to find the femoral artery on the subject's thigh (see Figure 5.15). For RTs, the maximum end-effector velocity was limited to 8 mm/s because higher velocities gave rise to mistrust on the part of the tele-operator. The velocity commands

Table 5.2 Results for some LTs over the phantom.

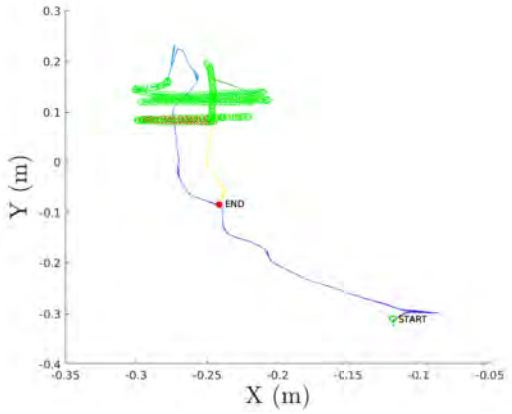
Magnitude	LT1	LT2	LT3	LT4
Tele-operation time (t_t) [s]	180.67	178.72	132.43	130.46
Body contact time (τ) [s]	55.88	44.51	45.87	51.22
Contact time ratio (ψ)	0.31	0.25	0.35	0.39
Dangerous warnings (w_R)	6	2	8	0
Caution warnings (w_Y)	15	6	14	4
Performance index (φ) [%]	57.02	52.62	62.07	88.66



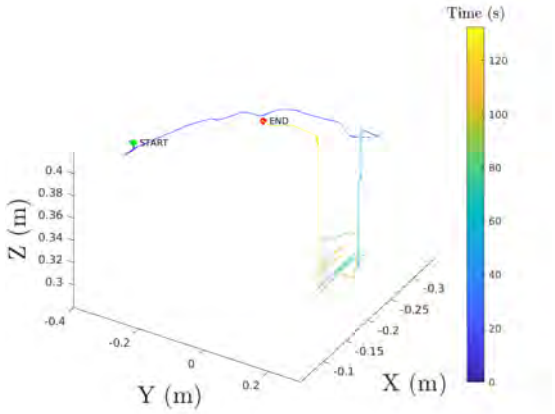
(a) Motion of end-effector along the axes of its local frame



(b) Risk levels for Z-axis forces are indicated by colors and dashed lines delimit the body contact



(c) 2D trajectory of end-effector with warnings



(d) 3D trajectory of end-effector

Figure 5.14 Results for LT3 (φ = 62.07%).

from the smartphone were always given at 60 Hz. However, the RA's behaviour was quite accurate up to 20 mm/s, at which point the ROS buffer tended to saturate, translating into discontinuous movement.

Table 5.3 presents the results from four consecutive RT runs (RT1, RT2, RT3 and RT4) carried out on the same day by the same tele-operator as in LTs. On previous days, up to fourteen remote tests were performed for operator training and developing data logging in ROS bags. The table indicates that the tele-operator had previous experience with the app interface and made fewer errors as the exercise was repeated. The noticeable decrease of φ in subsequent runs was due to the need for verbal interactions between the tele-operator and the assistant to record the dataset synchronously during RTs. In addition, each new RT run generated stress on both HAs, directly influencing φ.

Figure 5.16 shows the latency between the devices on each side of the network. The implemented code to control both the CRA and DT, and to get information from them can

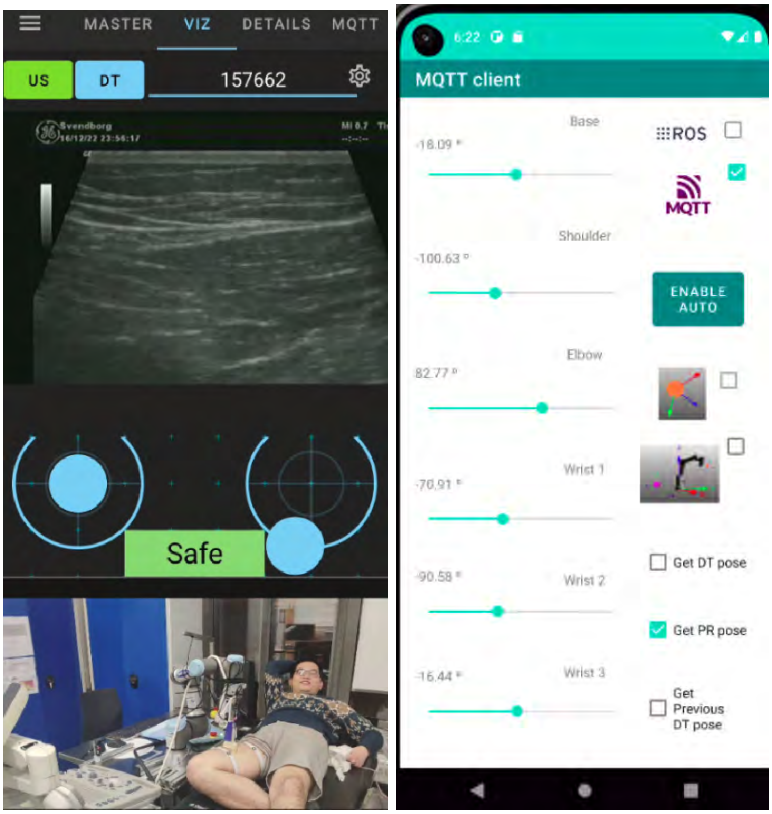


Figure 5.15 RT for robotic ultrasound scan on the subject's leg using the ad hoc ROS-Mobile. The integrated MQTT client synchronizes the movements of both DT and RA.

be run at either network endpoint, but the processing and latency were lower when code execution was done close to the RA. The maximum peak latency recorded was 318 ms (ping = 636 ms), while the minimum was 43 ms (ping = 86 ms), being 58 ms the average latency during the RTs.

5.3.4 Evaluation

This work presents an evolution of the IoCA architecture, focused on robotic manipulators and their implementation in robotic tele-medicine. It enables medical procedures to be carried out even when a physician with the required expertise is unavailable. The presented architecture is underpinned by location-aware services provided by/to agents, depending on their identity (robotic or human) and their physical situation in the application scenario and concerning it, enabling interconnection and data sharing between digital elements and agents with embedded MQTT clients and ROS nodes. The agents' cooperation is encouraged through the IoRT and the use of a DT, which bridge the virtual simulation and the real-world operation for safety consideration, allowing the control of the DT and the CRA either

Table 5.3 Results for some RTs over the subject's thigh.

Magnitude	RT1	RT2	RT3	RT4
Tele-operation Time (t) [s]	195.13	190.06	152.01	135.42
Body contact time (τ) [s]	82.22	66.02	52.39	42.26
Body time ratio (ψ)	0.42	0.35	0.34	0.31
Dangerous warnings (w_R)	2	1	0	0
Caution warnings (w_Y)	3	5	2	0
Performance index (φ) [%]	92.16	76.44	78.51	71.77

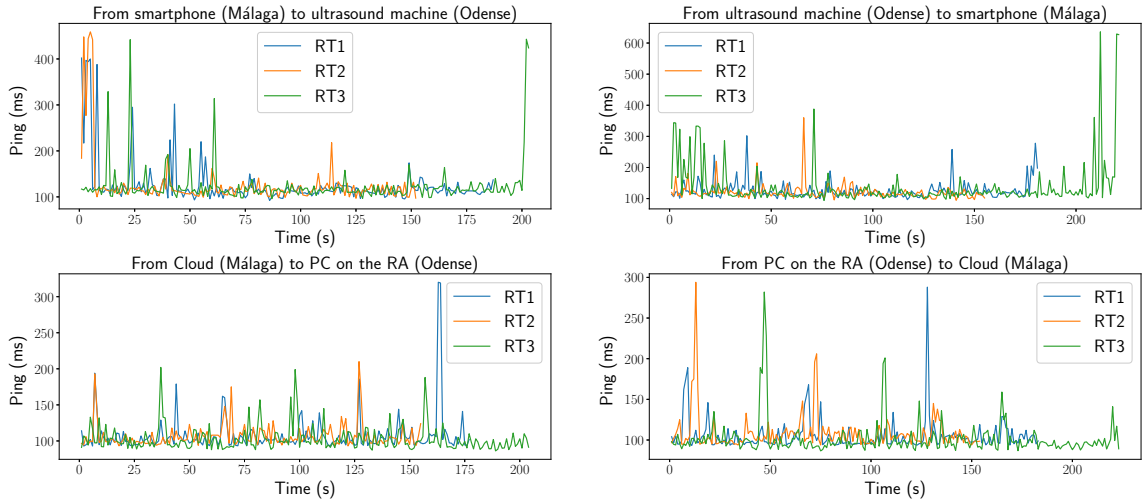


Figure 5.16 Ping values between the tele-operator and the operational environment during RT1, RT2 and RT3. The mean value was 116 ms, the distance being 2300 km.

separately or simultaneously. This allows the tele-operator to obtain visual feedback from the DT, thus facilitating his perception and operation. Finally, the new app links the IoRT information with the DT to give full visual input to the tele-operator.

For the work presented here, this device is detachable and usable on different types of robots to perform this particular medical operation. However, the prototype is being improved to allow for wireless use.

The architecture requires low latency and high bandwidth, so a 5G network is recommended. All endpoints are interconnected through a VPN, facilitating port forwarding needs in ROS 1. Using ROS 1 to manage the physical robot was the natural choice given its mature integration with robotic hardware and local real-time control. However, to expose selected events—such as ArUco marker detections—to external components like a mobile app or digital twin across a VPN-restricted university network, MQTT provided a lightweight, asynchronous, and firewall-friendly alternative. The ROS–MQTT bridge allowed both systems to interoperate effectively while respecting their respective strengths and network constraints.

However, a 5G network can be affected depending on the position of the mobile routers, especially when HAs work indoors. We verified this by conducting some RTs from indoor locations with worse signal propagation concerning the geolocation of the base station in Málaga. In that case, the average latency was higher than 400 ms, making the control difficult and requiring lower robot speed for the same data rates. Thus, 4G/5G crowd-cells, i.e., portable cellular networks [196], are recommended for those places with poor cellular coverage in SAR scenarios. Finally, the most realistic experiment reported the successful remote control of a 6-DOF manipulator, performing an ultrasound scan. The tele-operator used the ad hoc ROS-Mobile app to control the robot with an average communication latency of 58 ms over a distance of 2300 km.

5.4 Discussion and Conclusions

This chapter addresses the third contribution of this thesis: enhancing the situational awareness of heterogeneous agents for performance in demanding application scenarios requiring soft real-time capabilities, such as SAR and telemedicine.

For the UGV, efforts are directed towards extending its situational awareness to facilitate autonomous navigation to points of interest triggered by environmental events, with the IoRT overseeing path planning and managing UGV availability.

For the robotic arm, an IoRT-based system improves situational awareness by sharing positional data and object information relative to its end-effector. Its implementation is demonstrated in a medical tele-ecography application, enabled by a synchronized Digital Twin (DT) to ensure precise tele-operation over long distances.

Smartphones applications are now introduced as part of the PaaS, capable of sending commands to robots within the operational environments, such as requiring their assistance or tele-operating them. Thus, a smartphone no longer only influences the agent's perception, but also their interaction with their application scenario.

Additionally, the *SARFIS tool* incorporates ROS nodes to facilitate interaction between the SAR-FIS and ROS-FIS components of the IoCA architecture.

Finally, DTs are proposed as a service available in MEC centers to enhance the tele-operator's understanding of the robot's operational environment by providing comprehensive and multidimensional perspectives. Thus, the smartphone, working as a Cloud-Edge device, is used to tele-operate a robotic manipulator from a very distant location. Finally, the IoRT-in-hand emerges as a connected platform, resulting from the evolutions of the IoCA architecture.

Therefore, the *SARFIS tool*, VEROSIM, and the Android applications are the elements of an IoRT platform for providing services to agents in remote scenarios.

Introducing smartphones as Cloud-Edge devices enables not only the transmission of commands to robotic agents but also enhances overall situational awareness by providing contextual information about the operational environment. This new capability facilitates dynamic task assignment in response to environmental events.

A UGV has been demonstrated to autonomously navigate to a point of interest defined either by environmental sensor events or user requests via a smartphone. Integrating a H-WSN with remotely located control centers (FCC and BCC) validates the proposed architecture over long distances and through commercial cellular networks (4G/5G). This demonstrates the viability of scalable, distributed systems for remote robot operation, improving situational awareness by considering both local sensor data and global control information.

In addition, the *IoRT-in-hand* has been introduced as a smart end-effector device integrating sensors (RGB camera, force sensor, ultrasound scanner) and a mini-PC connected to the Internet. This Local-Edge device augments the manipulator's situational awareness by incorporating rich environmental data and object-related information. In a remote medical scenario, it enables tele-echography, supported by a DT, allowing a medical specialist to operate the manipulator remotely over long distances with real-time visual, sensory, and virtual feedback.

The evolution of the IoCA architecture, integrating ROS, MQTT, smartphones as Cloud-Edge devices, and DTs in MEC centers, marks significant progress toward a flexible IoRT platform for connecting agents in remote, outdoor, unstructured and disaster (ROUD) scenarios. This system effectively manages perception and supports autonomous or tele-operated control of heterogeneous robots, demonstrating robustness and scalability through sensor network integration, efficient long-distance communication, and MEC-based processing.

Chapter 6

Final Remarks

6.1 Conclusions

The information gathered from a mobile robot's operational environment can vary widely, encompassing a wide range of data types that can be utilized for various purposes depending on the specific application scenario. In Disaster Robotics, extracting data of varying sizes, formats, and contexts (e.g., environmental, chemical, geometric, temporal, and physical) poses significant challenges, particularly in remote, outdoor, unstructured, and disaster (ROUD) scenarios, where the absence of infrastructure to support communication and processing for deployed robots is a common issue.

Therefore, this pre-doctoral research focuses on resource-constrained scenarios addressing the co-design of perception and communication architectures, leveraging implementations across the Edge-Cloud continuum to facilitate sharing situational awareness between human and robotic agents in catastrophes. Situational awareness allows emergency robots to navigate hazardous environments, assess their location relative to the affected area, and adapt to real-time conditions. Thus, situational awareness, combined with precise positioning and robust telecommunications, empowers robots to effectively perform risk identification, environmental monitoring, and search and rescue operations.

The aim is to develop and enhance experimental solutions for Search and Rescue (SAR) agents through co-designing and integrating an Internet of Robotic Things (IoRT) platform. The proposed IoRT exhibits diverse characteristics regarding power consumption, throughput, latency, and coverage, both onboard and within the robot's operational environment. These perception systems allow each agent to be aware of its operational environment, leveraging Hybrid Wireless Sensor Networks (H-WSN), i.e., with static and mobile End-Devices (EDs), including sensor and concentrator nodes, built on different Personal Area Network (PAN) and Low Power Wide Area Network (LPWAN) technologies. Thus, the proposed IoRT platform

is designed to interconnect heterogeneous robots in ROUD scenarios, including UAVs, UGVs, and manipulators through heterogeneous hybrid wireless transceiver networks (H2WTN). In addition, the platform includes high bandwidth ad-hoc 5G sensors, including Light Detection and Ranging (LiDAR) systems and modern smartphones to obtain information from the agents' environment and their position within it, enabling data synchronization for offline and online operation.

The H2WTN can operate without Internet connectivity, delivering reliable services and information from ROUD environments over extended periods, even in the absence of traditional infrastructure. The integration of PAN, WLAN, and LPWAN technologies enables diverse and adaptable network topologies, while the introduction of online and offline multicasting techniques optimizes communication efficiency and adaptability to dynamic scenarios. The strategy of using mobile robots with a simultaneous dual role has proven to be highly advantageous in ROUD scenarios. In particular, the deployment of UGVs and UAVs acting as mobile data sources and sinks has been experimentally validated. However, ZEDs are only recommended to cover events at medium distances from SAR agents. Zigbee can be helpful, but it is not decisive for the detection of semi-buried victims, as tests indicate that the ZC in charge of discovering lost ZEDs needs to be too close to them (at least 20 m). Nevertheless, the UGV can provide more detections by proximity, which helps estimate the distance to the victim, while the UAV allows faster preliminary terrain reconnaissance. This allows for covering wider areas and detecting even semi-buried devices. In the H2WTN:

- The UAV's BLE scanner, with a direct LoS towards potential victims (PVs) on the ground, demonstrated higher sensitivity to transmissions than the UGV's scanner, which is more affected by interference and obstacles.

- The LPWAN option enables the coverage of large areas while transmitting data packets at regular intervals for extended periods of autonomy. The proposal to extract information from kilometers away from the operational environment of the robots has allowed the establishment of concentrated nodes with Internet access in areas far away from the ROUD scenario.

- The deployed LoRa H-WSN forms a *star of stars* configuration, unlike the Zigbee network, which is a set of independent stars that later aggregate the collected information. As a result, the Zigbee network provides information more slowly, whereas the LoRa network can deliver soft real-time information. In both cases, georeferenced sensory data are processed, enabling real-time tracking of agents carrying sensor nodes.

- PAN technologies serve as a close perception and detection system, while LPWAN technologies are designed to gather data from remote locations to the operational

environment. Consequently, it is essential to distinguish what is useful to transmit and how to do so (e.g., format, order, packet size, synchronism, power consumption, radio range, and sensitivity), considering the position of the EDs.

In all cases presented (BLE, Zigbee, and LoRa), the concentrator nodes must have access to a WLAN network so that they can transmit the information to remote control centers. In this regard, a WiFi mesh network with mobile mesh nodes is proposed to cover large areas in the ROUD scenario. In this way, it has been possible to establish a stable communications infrastructure capable of enabling a two-way transfer of information between the agents and the control centers.

However, combining the information captured by a team of agents is necessary to build a shared situational awareness to support decision-making. This decision-making (algorithms) can be executed by a remote human operator or autonomously, i.e., carried out by each robot. To this end, the Internet of Cooperative Agents (IoCA) architecture is presented, offering an intelligent link between perception and action through low-latency and high-bandwidth data services provided through network slicing. These services are distributed between Local- and Cloud-Edge computing devices deployed in the IoRT platform and a Feedback Information System (FIS), respectively. Local-Edge devices correspond to local hosts in the agents' operational environments, while Cloud-Edge devices are part of the Multi-Access Edge Computing (MEC) centers. Using these data, Local-Edge devices act as an Infrastructure as a Service (IaaS), providing and receiving services at the Edge, operating close to the operational environments, and connecting to robots via WLAN/LAN networks. Similarly, Cloud-Edge devices enable broader reach and integration with Internet-connected resources, acting as a Platform as a Service (PaaS). Moreover, for communication, agents have one or more User Equipment (UE) devices (e.g., a 5G router or a 5G smartphone).

On the other hand, the robot's onboard computing equipment may sometimes require specific algorithms or services that are not locally available, being necessary to share computing resources belonging to other robots operating in the same scenario (Edge) or to servers hosted remotely (Cloud). While sending raw data might be essential for specific applications, bandwidth limitations often demand local preprocessing to transmit only the most relevant information. MEC centers address these challenges by intelligently distributing processing tasks across operational environments, including the onboard computers of deployed robotic agents. This approach helps reduce operational latencies and supports meeting real-time requirements effectively. Two use cases of the IoCA architecture are presented: a comparison between a SLAM running on the Edge and a SLAM running on the Cloud; and a multilateration algorithm running on the Cloud to position detected victims in real time.

The remote SLAM (executed in the Cloud) is successfully implemented using network slicing techniques, integrated into a pilot 5G network free of commercial clients. This setup enables comprehensive testing of the end-to-end architecture under bandwidth saturation conditions, demonstrating the system's ability to maintain performance and adapt to the demands of high-data-rate tasks. This Cloud version saves energy consumption and space on the robot, allowing it to separate the map construction and transmission of the sensory raw data (the cloud point). When there is no option to use the Cloud, Edge SLAM can be run. The proposed algorithm has not led to a bottleneck in processing.

Additionally, this thesis explores the development of mobile applications to integrate smartphones into robotics.

Integrating H-WSN with high-bandwidth ad-hoc sensors is proposed to enhance robots' location awareness, ensuring that their pose (position and orientation) remains continuously aligned with environmental measurements. Including static and mobile nodes increases system redundancy and flexibility, enabling more effective deployment in dynamic scenarios. This diverse IoRT platform addresses key limitations faced by robots in ROUD environments, such as restricted Internet coverage, bandwidth constraints, and latency, which can otherwise hinder their ability to act safely and effectively. Additionally, digital twins are incorporated into the IoCA architecture to ensure the safe tele-operation of robots located in ROUD environments.

Finally, open-source software has been developed as applications to support SAR agents in disaster scenarios. Table 6.5 summarizes the applications created and implemented in various realistic emergency field experiments. All applications have been designed to be compatible with ROS 1 and ROS 2 architectures, enabling sensing, monitoring, and data processing for SAR agents. The inclusion of smartphones in ROUD scenarios has facilitated the design of robotic tasks by meeting a combination of mobile communication and distributed processing requirements. Additionally, access to experimentation with a 5G pilot network, free from commercial users, has allowed the design of an IoRT platform tailored for ROUD scenarios, including insights into the data traces required by the agents. For instance, it has been possible to manage bandwidth network slicing to build real-time SLAM wirelessly while simultaneously performing other computationally and communication-intensive tasks.

The *SAR-FIS tool*, designed and evolved to integrate different research works, has been developed by the Robotics and Mechatronics group of the UMA, the code development leader being the researcher Manuel Toscano-Moreno, to whom we are grateful for his contribution. Throughout this thesis, the *SAR-FIS tool* has evolved to integrate both ROS 1 and ROS 2 simultaneously, serving as the main Graphical User Interface (GUI) in the control centers to monitor sensor network data, ROS-integrated cameras, agent tracking, and even RTT-based

Table 6.1 GitHub repositories for robotic applications in ROUD scenarios

Development	Characteristics	Repository
UMA-ROS-Android	ROS 1 integration in Android to sensor SAR agents	https://github.com/Juan013/uma-ros-android
UR2A	Remote command reception, sensory transmission through ROS 2, Fractal ArUco detection	https://github.com/Robotics-Mechatronics-UMA/UMA-ROS2-Android
Ad-hoc ROS Mobile	Digital twin visualization and control	https://github.com/Robotics-Mechatronics-UMA/Adhoc-ROS-Mobile
ILS algorithm for an FTM-based H-WSN	ROS 1 positioning system and tracking	https://github.com/Robotics-Mechatronics-UMA/ROUD-PositioningSystem.git
ILS algorithm for an FTM-based One Single Agent (OSA)	ROS 2 positioning system	https://github.com/Robotics-Mechatronics-UMA/OSA-PositioningSystem.git

detection and multilateration-based position estimation. However, specific programs and algorithms, such as offline/online synchronization of databases hosted on concentrator nodes, PAN, WLAN, and LPWAN network configuration and monitoring packages, multilateration, remote control of manipulators, as well as the development and control of digital twins (developed with the VEROSIM tool), router traffic monitoring, remote execution of SLAM, or network slicing control, have been developed independently, distributing the computational load across different hosts in the IoCA architecture.

6.2 Future Works

The following section outlines potential directions for future research and development, building upon the findings and contributions of this doctoral thesis to further advance the field. Thus, future lines are:

- To enhance the integration of mobile robots as active mobile nodes within diverse H-WSNs to optimize georeferenced data extraction from operational environments. This data can be collaboratively processed using onboard Edge computing resources to generate detailed terrain and environmental information maps. A key objective is to explore advanced strategies for **fusing data from multiple agents equipped with identical sensory and communication payloads, navigating the same area of interest**. These agents may perceive the environment from different contexts, times, and perspectives. Integrating this collective information into ROS 2 will enable improved interoperability and data sharing, providing richer inputs for advanced algorithms running on MEC centers (Cloud computing) to support real-time decision-making and situational awareness.

- To include new functionalities into the *SAR-FIS tool*. SAR-FIS was the part with non-ROS things, and as ROS was introduced in the systems, non-ROS programs reduced their space in the FIS. However, in ROS 1, this was detrimental, given the centralized nature of the architecture.

  It is possible to have several instances of *SAR-FIS tool* open and to allocate the CPU consumption of each host to one or the other. It has been seen that CPU consumption can affect information monitoring, especially when the frequency of data received under ROS subscription is high. For this reason, the tool should be scaled up gradually because multiple operators are needed.

  Eventually, a new artificial intelligence (AI) entity could be created to replace X-FIS, which could identify the application scenarios and their requirements when running. This AI-FIS could run different generative models for autonomous processing, filtering, and monitoring data, identification of new agents, identification of improvements, etc.

  All these potential extensions of the *SAR-FIS tool*, developed in MATLAB, could be integrated into a MATLAB toolbox.

- **To explore the use of mobile 6G crowd cells to improve connectivity in ROUD scenarios, allowing robots to act as big data MULEs**.

  This paradigm shift would advance the IoCA architecture, transitioning from prioritizing connectivity to enabling hard real-time action in ROUD scenarios. For instance, a collaborative SLAM process could integrate georeferenced data collected by various IoRT platforms, classifying risk zones and areas with a high probability of locating victims by fusing lightweight data with heavier data sources such as cameras and LiDAR. This processing would primarily occur locally on each robot.

  However, the more information a robot must handle, the greater the need for a high-bandwidth, low-latency network in the operational scenario to transfer the big data to MEC centers for algorithm execution (e.g., SLAM).

  Collaborative SLAM could also be performed using updated maps received at MEC centers from different robots in the field, incorporating data from the H-WSNs. In this context, investigating the use of 6G crowd cells in the operational area would be beneficial. These mobile cells could be strategically positioned in zones with optimal coverage to serve as checkpoints for robots acting as big data MULEs. The MEC could be mobile if the crowd cell is mounted on a robot that provides internet access to others.

- To enhance the UR2A application, which has already been developed to detect and estimate the distance to fiducial fractal markers. Future improvements could focus on increasing detection accuracy, extending compatibility with additional marker types, and integrating advanced vision-based localization algorithms. Leveraging this Android application for tasks such as object tracking, pose estimation, and cooperative mapping between robots could significantly improve perception and interaction capabilities in ROUD environments. These tasks would enhance situational awareness, enabling autonomous robot navigation. For example, the IoRT platforms developed in this work to increase UAV payload capacity could also place fiducial markers on surfaces visible to other agents (e.g., the underside of a UAV or the top of a UGV). This could enhance inter-agent localization using vision-based techniques.

- **To track human agents' health status through specialized sensors linked via Bluetooth to the smartphone**. Then, various physiological parameters can be captured, such as electrodermal activity (EDA), electrocardiogram (ECG), blood oxygen levels, or lung capacity. This information could be helpful to detect and evaluate the stress of the subjects (e.g., the rescuers) wearing the sensors [197–199], and can be published by implementing ROS nodes.

- To implement an efficient method **to filter and distinguish portable devices likely carried by victims from irrelevant ones**. Additionally, the system should be improved to support real-time BLE data collection and visualization.

- **To optimize the mesh network's efficiency inside tunnels and underground scenarios**, considering different network configurations to achieve higher bandwidth. The used EDs were chosen to work with the 802.11 mc standard to incorporate multilateration algorithms to utilize FTM technology, aiding in estimating the relative distance between all mesh nodes from the outdoor position of the repeater-UGV to the interior positions of the other mesh nodes. Future studies could also analyze techniques for autonomous UGV navigation indoors, providing coverage and Internet access to underground users, merging FTM and UWB technologies to detect and locate potential victims indoors and outdoors.

- To enhance the integration achieved with the IoCA architecture, emphasizing the concept that effective team communication can serve as a predictor of team performance. Initially, SAR-FIS accessed the robots' onboard cameras through requests to web servers hosted on public sockets, requiring additional PCs at the FCC and the BCC to monitor the information published by ROS nodes. **Integrating ROS-FIS in SAR-FIS**

eliminates that requirement, simplifying the deployment and operation of the system. ROS allows working with heavy messages, such as raw image flow or the point cloud of a LiDAR, but also with audio so that processing and machine learning techniques can be applied, allowing for the implementation of many applications.

- The resulting FTM-based localization system is accurate, low-cost, and easy to deploy. Moreover, the final strategy, where only one FTM anchor is needed, is faster than others in detecting and geolocating the PVs. The proposed system has significant potential for improvement, as it can be done with other agents (UAVs and UGVs) so that each mobile agent's detections are computed only for it. In addition, it helps compare the estimates of one agent concerning another. All the agents' forecasts concerning the same PV can be filtered, either with an EKF or a simpler filter, and even greater precision can be obtained. However, this is still under investigation. It remains to be applied in the Cloud as an extra algorithm that acts on the location estimated by each agent for a specific detected PV. In other words, **a combination of the two presented FTM-based strategies can be applied**. In this case, the number of RTT detections by each agent can be exploited (as in strategy B) since it does not depend on secondary anchors to apply multilateration, each position of the mobile anchor (robot) being a new position from which multilateration is computed. Suppose this is used for several mobile agents carrying anchors. In that case, there will be different estimates from different points, taking advantage of the location awareness of each agent to locate a victim carrying a WiFi device. The results of each agent would be communicated to the FIS, from where a result could be computed that makes use of the outputs of each agent, thus achieving greater accuracy.

- **To distinguish between WiFi signals from non-victim devices and those of actual victims, as the system currently detects all WiFi devices**. Developing algorithms to filter out irrelevant signals would improve detection accuracy and reliability. The challenge lies in taking a step further by adapting, through bandwidth network slicing, the images captured by the UAV to the positions estimated by it, as it is used as One Single Agent (OSA).

- To expand the SAR-IoCA implementation with several enhancements. These include **integrating a mobile manipulator synchronized with a DT server, utilizing the MQTT-ROS bridge for communication, enabling teleoperations from distant cells with low latency through 5G/6G networks, and developing an application that addresses diverse use cases within the IoCA framework**. Additional efforts will aim to improve interaction between the teleoperator and the DT, implement location-

aware services based on ROS 2, and explore imitation learning techniques for robotic manipulators.

- **To advance the developments of the IoRT-in-hand for realistic medical applications**, such as tele-REBOA [200], by integrating them into disaster scenarios through a mobile manipulator. This would involve remotely connecting robots and medical specialists, leveraging low-latency Edge-Cloud architectures and high-bandwidth 5G networks to enable efficient real-time operations.

- **To implement a real-time Streaming Protocol (RTSP) server is considered to acquire high-quality ultrasound images**, reducing the workload of the mobile router in the operational environment. Some tests show this server could add two seconds to the current latency, which is unacceptable for tele-operation, but it can be for soft real-time remote diagnosis at the health center. HAs at this MEC center could apply AI to detect illnesses. In addition, data mining based on machine learning should be explored to discover patterns and insights to improve operational efficiency, detect anomalies or fraud, predict future trends or events, and optimize resource allocation. Finally, the robotic ultrasound manipulator should be licensed.

  Better compression—and therefore lower bandwidth usage—could be achieved using more advanced codecs such as **H.264**, or ideally **H.265** (HEVC). These codecs leverage inter-frame compression, enabling significant reductions in data rate while maintaining acceptable visual quality—especially useful in environments with limited bandwidth and variable connectivity, such as tunnel-based mesh networks. In contrast to MJPEG, which compresses each frame independently, H.264 and H.265 exploit temporal redundancy between frames, making them far more efficient for video streaming.

  To implement this efficiently in future deployments, especially within a **ROS 2**-based architecture, the use of **DDS** (Data Distribution Service) middleware provides a promising solution. DDS is designed for real-time, scalable, and flexible communication, and can be configured to handle video streams as custom message types or as payloads in optimized transport channels. In particular, ROS 2 nodes could publish compressed video frames—encoded using H.264 or H.265—as binary blobs (e.g., `sensor_msgs/msg/CompressedImage` or custom messages), while DDS Quality of Service (QoS) policies such as *reliability*, *latency budget*, and *transport priority* could be fine-tuned to ensure consistent delivery under varying network conditions.

  This approach would allow higher-resolution video to be transmitted more efficiently, minimize network congestion, and provide better synchronization and robustness for

multi-agent systems operating in challenging environments. As such, integrating hardware-accelerated compression and DDS-based transport mechanisms within ROS 2 represents a key step toward more scalable and adaptive robotic deployments in real-world scenarios.

- **To integrate all the proposed Edge-Cloud services in the IoCA architecture into a web platform, making the PaaS remotely manageable**. This would prevent the generation of extra throughput each time a Local-Edge or Cloud-Edge device subscribes to any information.

Resumen de la tesis doctoral

Introducción

Esta tesis doctoral tiene un carácter práctico sólido y todos los resultados se han obtenido en una serie de ejercicios realistas de respuesta ante desastres que tuvieron lugar en el Laboratorio y Área de Experimentación en Nuevas Tecnologías para Emergencias y Catástrofes (LAENTIEC) en Málaga (España). Así, las contribuciones de esta tesis doctoral son resultado de una participación activa en todas las Jornadas Internacionales de Seguridad, Emergencias y Catástrofes (JEMERG), organizadas por la Cátedra del mismo nombre, de la Universidad de Málaga (UMA), entre 2019 y 2024. El Departamento de Ingeniería de Sistemas y Automática (ISA) colabora cada año en JEMERG, evento dirigido por el coordinador de la cátedra, el Dr. Jesús Miranda. El área operativa del LAENTIEC, que abarca una superficie utilizable de 115,000 m², sirve como escenario al aire libre para el marco de JEMERG, caracterizado por un perfil de terreno irregular y vegetación densa en ciertas áreas debido a la proximidad de un arroyo. Un área de Comando, Comunicación y Control (CCCA) compuesta por varias carpas se monta en el área de LAENTIEC para apoyar a los equipos de SAR. Cada año, nuestro grupo de investigación proporciona servicios novedosos a través de robots heterogéneos desde el Centro de Control Avanzado (FCC), ubicado dentro de la carpa del ISA en el CCCA.

JEMERG ha involucrado una amplia variedad de primeros intervinientes, incluidos cuerpos de bomberos, grupos de atención sanitaria, la policía nacional, equipos militares como la fuerza aérea, la Unidad Militar de Emergencias (UME) y la Legión. Otros participantes incluyen personal de protección civil, actores simulando víctimas (interpretados por estudiantes de teatro), medios de comunicación autorizados y visitantes. Además, grupos de investigación y empresas especializadas en robótica de campo colaboran en diversos escenarios dentro de JEMERG para desarrollar nuevas tecnologías que mejoren las tareas de búsqueda y rescate (SAR).

Los grupos de investigación del Instituto de Mecatrónica (IMECH) de la UMA, incluidos el Equipo de Robótica y Mecatrónica de la UMA (integrado en el Departamento de Ingeniería de Sistemas y Automática), aportan sus recursos para estos ejercicios anuales, cuyo objetivo

es probar y evaluar las tecnologías emergentes en desarrollo e investigación, trabajando junto a primeros intervinientes reales en misiones SAR. Las tareas se realizan bajo condiciones realistas, cumpliendo con estrictos plazos y protocolos de seguridad. Estas misiones implican la búsqueda y rescate de personas en escenarios de emergencia causados por factores ambientales, terrorismo o desastres naturales como incendios, terremotos e inundaciones.

Los datos exteroceptivos son fundamentales en la robótica de campo, donde los robots deben comprender su entorno operativo. En particular, un robot móvil tiene que percibir información muy diversa sobre su entorno mientras navega por un área de interés. Adquirir y procesar esta información, junto con su formato adecuado, es una tarea desafiante. Los datos deben ser comprimidos de manera eficiente y transmitidos de forma segura sin comprometer su calidad, asegurando su accesibilidad remota en diversos escenarios de aplicación. Además, los robots móviles de exploración, incluidos los rovers planetarios y las plataformas robóticas de SAR, requieren Conciencia Situacional (SA) para interactuar eficazmente con su entorno. La SA permite a estos robots tomar decisiones informadas basadas en su ubicación espacial en un mapa y las condiciones en tiempo real de su entorno inmediato. La SA va de la mano con el posicionamiento y las telecomunicaciones.

En escenarios de desastre, como guerras o catástrofes naturales, los sistemas de posicionamiento y comunicación efectivos son cruciales para la coordinación y toma de decisiones en misiones de búsqueda y rescate. Sin embargo, estos sistemas enfrentan desafíos significativos bajo tales condiciones.

Todos los Sistemas Globales de Navegación por Satélite (GNSS) proporcionan capacidades de geolocalización sin requerir una conexión a Internet. Por ejemplo, un teléfono inteligente puede registrar su posición GNSS incluso en modo avión. Si bien esta independencia de la infraestructura terrestre hace que el GNSS sea invaluable en escenarios de desastre, su fiabilidad puede verse comprometida. Los problemas de falta de línea de visión (LoS) causados por nubes, terreno, vegetación u obstáculos artificiales suelen degradar la precisión de las señales GNSS. Además, las tormentas geomagnéticas y las interrupciones intencionadas de la señal, como el bloqueo en conflictos militares, pueden reducir aún más su efectividad. Estas limitaciones resaltan la importancia de desarrollar sistemas de posicionamiento secundarios para garantizar redundancia.

Además, la fiabilidad de los sistemas de comunicación es crítica en escenarios de desastre, ya que deben permitir la coordinación, el intercambio de información y la planificación de respuestas. Estos sistemas, que pueden incluir tecnologías satelitales, celulares y basadas en radio, son especialmente vitales en áreas donde la infraestructura terrestre, como el WiFi o las estaciones base de telefonía móvil, ha sido destruida o es poco confiable.

Los obstáculos pueden bloquear de forma crítica el flujo de información, incluidos los datos de sensores esenciales para que los sistemas robóticos perciban e interactúen con su entorno. Asegurar una conectividad continua es fundamental, ya que los robots dependen de datos de tiempo real blando y duro, así como de comunicaciones síncronas y asíncronas, para construir SA, compartir información y realizar tareas de forma autónoma.

Para abordar estos desafíos en escenarios de desastre, es necesario investigar el diseño, desarrollo e implementación práctica de arquitecturas capaces de soportar la percepción, las telecomunicaciones, el procesamiento distribuido y la conciencia de ubicación.

Contribuciones

Esta tesis tiene como objetivo impulsar la cooperación de agentes heterogéneos, tanto robóticos como humanos, en entornos ROUD a través del despliegue de un Internet de las Cosas Robóticas (IoRT, por sus siglas en inglés) compuesto por elementos (e.g., enrutadores, gateways, actuadores y sensores) de consumo de energía, ancho de banda, latencia y cobertura variados, tanto a bordo como dentro del entorno operativo del robot. Este trabajo se centra en escenarios con limitaciones de recursos, abordando el co-diseño de arquitecturas de percepción y comunicación, aprovechando implementaciones a través del continuo edge-cloud para facilitar el intercambio de SA entre agentes humanos y robóticos en situaciones de emergencia. Los medios sensoriales integrados incluyen Redes de Sensores Inalámbricos (WSN, por sus siglas en inglés) basadas en diversas tecnologías, sensores de alto ancho de banda como los sistemas LiDAR (Light Detection and Ranging), y teléfonos inteligentes modernos. Estos sistemas se utilizan para recopilar información sobre el entorno de los agentes y su posición dentro de él, permitiendo la sincronización de datos para operaciones tanto en línea como sin cobertura. Además, esta tesis presenta los resultados de escenarios realistas desarrollados en colaboración con profesionales del sector de emergencias en JEMERG.

El IoRT propuesto incluye diferentes Redes Híbridas de Sensores Inalámbricos (H-WSN, por sus siglas en inglés), es decir, con Dispositivos Finales (EDs, por sus siglas en inglés) estáticos y móviles, incluidos nodos sensores y concentradores, y sensores ad hoc de gran ancho de banda que integran diversas tecnologías, dependiendo de las necesidades.

Además, se propone el uso de redes de baja potencia, particularmente Redes de Área Personal (PAN, por sus siglas en inglés) y Redes de Área Amplia de Bajo Consumo (LPWAN, por sus siglas en inglés) para asegurar la adquisición de datos a largo plazo en escenarios ROUD. Se propone que los agentes SAR lleven EDs PAN y LPWAN. Las tecnologías PAN se sugieren como un sistema cercano de percepción y detección, mientras que las

tecnologías LPWAN están diseñadas para obtener datos de áreas remotas hacia el entorno operativo. Por lo tanto, es esencial distinguir qué es útil transmitir y cómo hacerlo (formato, orden, tamaño de paquete, sincronización, consumo de energía, alcance radioeléctrico y sensibilidad), considerando la posición de los EDs.

Por otro lado, el equipo de computación a bordo del robot puede requerir a veces algoritmos o servicios específicos que no están disponibles localmente, siendo necesario compartir recursos de computación pertenecientes a otros robots que operan en el mismo escenario (Edge) o a servidores alojados remotamente (Cloud). Si bien el envío de datos sin procesar puede ser esencial para ciertas aplicaciones, las limitaciones de ancho de banda a menudo requieren preprocesamiento local para transmitir solo la información más relevante. Los centros de Computación de Borde de Múltiples Accesos (MEC, por sus siglas en inglés) abordan estos desafíos distribuyendo inteligentemente las tareas de procesamiento en todo el escenario de la aplicación, incluidos los ordenadores a bordo de los agentes robóticos desplegados. Este enfoque ayuda a reducir las latencias operacionales y apoya el cumplimiento de los requisitos en tiempo real de manera efectiva.

Esta tesis doctoral contribuye a responder las siguientes preguntas:

- ¿Cómo se puede obtener información de terrenos peligrosos, donde se cree que hay personas perdidas o atrapadas, y donde es necesario un rescate rápido?
- ¿Cómo se puede distribuir de manera segura toda la información recopilada a robots móviles y operadores humanos en ubicaciones remotas, y cómo se puede aplicar esto a casos de uso SAR?
- ¿Cómo se pueden gestionar equipos de robots terrestres y aéreos de forma remota basándose en la información que comparten desde el área operativa?

Las principales contribuciones son las siguientes:

- **Contribución 1**. Codiseño, desarrollo y despliegue de H-WSN y sistemas robóticos de emergencia en tiempo real suave, uniendo de manera efectiva los campos tradicionalmente separados de telecomunicaciones y robótica de campo. Las H-WSN se basan en tecnologías de baja potencia (incluyendo PAN y LPWAN), utilizando topologías diversas adaptadas para su aplicación y evaluación en entornos ROUD bajo condiciones realistas. Diseñadas para operar sin conectividad a Internet, estas redes garantizan un rendimiento fiable incluso cuando la infraestructura tradicional no está disponible. Además, se desarrollan técnicas de multidifusión tanto online como offline para optimizar la eficiencia de las comunicaciones.

- **Contribución 2**. Diseño, implementación y evolución de una arquitectura Edge-Cloud para integrar un IoRT diverso, incluyendo H-WSN y sensores ad hoc 5G con requisitos de baja latencia y ancho de banda adaptativo (segmentación de red). Indicadores clave como latencia, rendimiento, pérdida de paquetes, cobertura, calidad de servicio y carga de procesamiento (ya sea distribuida cerca o lejos del lugar de generación de datos) son cuidadosamente regulados para asegurar un rendimiento óptimo. La arquitectura propuesta ha sido implementada en misiones SAR realistas en escenarios ROUD.
- **Contribución 3**. Mejora de la SA de agentes heterogéneos (incluyendo humanos, vehículos terrestres y aéreos no tripulados, y manipuladores robóticos) para aplicaciones exigentes que requieren capacidades en tiempo real suave, como SAR y telemedicina, mediante el uso de Gemelos Digitales (DT, por sus siglas en inglés) y telecontrol. Esto se logra integrando redes ROS con clientes y servidores MQTT.

Las tablas siguientes muestran la producción científica de esta tesis.

Table 6.2 Publicaciones de calidad (JCR)

Work	Title	Authors	Journal	Year	Quartile
1	Development and implementation of a hybrid wireless sensor network of low power and long range for urban environments.	**Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Serón, J., Gómez-Ruiz, J. A., García-Cerezo, A.	Sensors	2021	**T1, Q2**
2	The Internet of Cooperative Agents architecture (X-IoCA) for robots, hybrid sensor networks, and MEC centers in complex environments: a Search and Rescue case study.	**Bravo-Arrabal, J.**, Toscano-Moreno, M., Fernández-Lozano, J. J., Mandow, A., Gómez-Ruiz, J. A., García-Cerezo, A.	Sensors	2021	**T1, Q2**
3	Realistic deployment of Hybrid Wireless Sensor Networks based on Zigbee and LoRa for Search and Rescue applications.	**Bravo-Arrabal, J.**, Zambrana, P., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Barba, J. S., García-Cerezo, A.	IEEE Access	2022	**T1, Q2**
4	Bluetooth Low Energy for close detection in Search and Rescue missions with robotic platforms: an experimental evaluation.	Cantizani-Estepa, J., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Fortes, S., Barco, R., García-Cerezo, A., Mandow, A.	IEEE Access	2022	**T1, Q2**

Estructura de la tesis

Esta tesis doctoral, titulada *Plataforma de Internet de cosas robóticas para conectar agentes en escenarios de desastre*, ha sido escrita como una monografía y se ha dividido en seis capítulos. Excepto el presente capítulo y el dedicado a conclusiones y trabajo futuro, cada capítulo comienza con una introducción que define el problema científico abordado, resume los desafíos, presenta las contribuciones realizadas y finaliza con una sección de evaluación de los resultados experimentales obtenidos. Las referencias bibliográficas y los anexos se encuentran al final del documento.

Table 6.3 Publicaciones en congresos

Work	Title	Authors	Conference	Year
1	Sistema de comunicación de respaldo mediante tecnología LoRa con hardware y software abierto para aplicaciones de robótica de emergencias.	Manrique Balmaceda, R. F., Vázquez-Martín, R., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., García-Cerezo, A.	XLII Jornadas de Automática (Castellón, Spain)	2021
2	Integrating ROS and Android for rescuers in a Cloud Robotics architecture: application to a casualty evacuation exercise.	Toscano-Moreno, M., **Bravo-Arrabal, J.**, Sánchez-Montero, M., Barba, J. S., Vázquez-Martín, R., Fernández-Lozano, J. J., García-Cerezo, A.	IEEE SSRR (Sevilla, Spain)	2022
3	Remote planning and operation of a UGV through ROS and commercial mobile networks.	Sánchez-Montero, M., Toscano-Moreno, M., **Bravo-Arrabal, J.**, Serón Barba, et al.	Iberian Robotics Conference (Zaragoza, Spain)	2022
4	Field report on experimental comparison of a WiFi mesh network against commercial 5G in an underground disaster environment.	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Mandow, A., Barco-Moreno, R., García-Cerezo, A. J.	IEEE SSRR (Fukushima, Japan)	2023
5	UR2A: comunicación bidireccional Android-ROS 2 para arquitecturas edge-cloud en sistemas robóticos conectados.	Córdoba-Ramos, M., **Bravo-Arrabal, J.**, Fernández-Lozano, J. J., Mandow, A., García-Cerezo, A.	XLV Jornadas de Automática (Málaga, Spain)	2024

Table 6.4 Artículos de calidad (JCR) enviados para su publicación

Work	Title	Authors	Journal	Quartile
1	Real-Time FTM-based Victim Positioning System Using Heterogeneous Robots in Remote and Outdoor Scenarios	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Toscano-Moreno, M., Serón-Barba, J., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Khatib, E. J., Barco, R., García-Cerezo, A.	Submitted to IEEE Access	**T1, Q2**
2	Real-Time FTM-based Victim Positioning System for OSA: One Search Agent	**Bravo-Arrabal, J.**, Álvarez-Merino, C. S., Toscano-Moreno, M., Serón-Barba, J., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Khatib, E. J., Barco, R., García-Cerezo, A.	Submitted to IEEE Network	**T1, Q1**
3	The IoRT-in-Hand: Tele-Robotic Echography and Digital Twins on Mobile Devices	**Bravo-Arrabal, J.**, Cheng, Z., Fernández-Lozano, J. J., Gómez-Ruiz, J. A., Schlette, C., Savarimuthu, T. R., García-Cerezo, A.	Submitted to Internet of Things (Elsevier)	**T1, Q1**

El Capítulo 2, *Contexto y Estado del Arte en Robótica de Emergencia*, presenta una revisión de la literatura sobre los desafíos y soluciones existentes relacionados con la percepción y las arquitecturas Edge-Cloud para robótica de emergencia.

El Capítulo 3, *Percepción y Comunicación mediante Redes Inalámbricas en Escenarios Complejos*, detalla el diseño, desarrollo y despliegue de redes de baja potencia compuestas por nodos estáticos y móviles. Estos nodos móviles están montados en unidades móviles (e.g., un UAV, un perro de búsqueda y rescate o un humano). Estas redes H-WSN pueden operar sin conectividad a Internet, manteniendo un servicio constante y proporcionando información fiable desde entornos no urbanizados (ROUD), incluso sin infraestructuras tradicionales durante largos periodos, aprovechando tecnologías PAN, WLAN y LPWAN en diversas topologías. Además, el capítulo introduce técnicas de multicast online y offline para

optimizar la eficiencia de las comunicaciones, garantizando la adaptabilidad del sistema a escenarios cambiantes. Se presenta la Contribución 1.

El Capítulo 4, *Fomento de la Cooperación entre Agentes mediante MEC y Dispositivos Móviles*, describe el desarrollo de la arquitectura del Internet de los Agentes Cooperativos (IoCA), que surge de la necesidad de relacionar los servicios proporcionados y requeridos por los distintos agentes desplegados, facilitando su cooperación mediante una distribución inteligente entre el Borde y la Nube. Aquí, los centros de cómputo remoto o Multi-Access Edge Computing (MEC) ofrecen recursos y servicios basados en Network Slicing, conciencia de localización (LA, por sus siglas en inglés) o necesidades de control. Además, se describe el desarrollo de aplicaciones móviles para promover el uso de smartphones en robótica, comentando sus ventajas y desventajas. Se presenta la Contribución 2.

El Capítulo 5, *Conciencia Situacional para el Control de Robots Heterogéneos*, aborda cómo los agentes interactúan con su entorno operativo mediante objetivos, aumentando su conocimiento del espacio de trabajo. La arquitectura IoCA ha evolucionado para implementar nuevas características, incluyendo control y Gemelos Digitales. Se explica la Contribución 3.

El Capítulo 6, *Conclusiones Finales*, destaca las contribuciones más relevantes de esta tesis y propone líneas de investigación futura que, hasta la fecha y según el conocimiento del autor, no han sido abordadas en la literatura.

Conclusiones

La información recopilada del entorno operativo de un robot móvil puede variar ampliamente, abarcando una gran variedad de tipos de datos que pueden ser utilizados para diversos fines según el escenario de aplicación específico. En Robótica de Desastres, extraer datos de diferentes tamaños, formatos y contextos (e.g., ambientales, químicos, geométricos, temporales y físicos) presenta desafíos significativos, particularmente en escenarios remotos, al aire libre, no estructurados y de desastre (ROUD, por sus siglas en inglés), donde la ausencia de infraestructura para apoyar la comunicación y el procesamiento de los robots desplegados es un problema común.

Por lo tanto, esta investigación predoctoral se centra en escenarios con limitaciones de recursos, abordando el co-diseño de arquitecturas de percepción y comunicación, aprovechando implementaciones a lo largo del continuo Edge-Cloud para facilitar el intercambio de Conciencia Situacional (SA, por sus siglas en inglés) entre agentes humanos y robóticos en catástrofes. SA permite a los robots de emergencia navegar por entornos peligrosos, evaluar su ubicación en relación con el área afectada y adaptarse a las condiciones en tiempo real. Así, SA, combinada con posicionamiento preciso y telecomunicaciones robustas, capacita a

los robots para realizar de manera efectiva identificación de riesgos, monitoreo ambiental y operaciones de búsqueda y rescate.

El objetivo es desarrollar y mejorar soluciones experimentales para agentes de Búsqueda y Rescate (SAR, por sus siglas en inglés) mediante el co-diseño e integración de una plataforma de Internet de las Cosas Robóticas (IoRT, por sus siglas en inglés). La IoRT propuesta exhibe diversas características en cuanto a consumo de energía, ancho de banda, latencia y cobertura, tanto a bordo como dentro del entorno operativo del robot. Estos sistemas de percepción permiten que cada agente sea consciente de su entorno operativo, aprovechando Redes de Sensores Inalámbricos Híbridas (H-WSN, por sus siglas en inglés), es decir, con Dispositivos Finales (EDs, por sus siglas en inglés) estáticos y móviles, incluyendo nodos sensores y concentradores, basados en diferentes tecnologías de Redes de Área Personal (PAN, por sus siglas en inglés) y Redes de Área Amplia de Baja Potencia (LPWAN, por sus siglas en inglés). Así, la plataforma IoRT propuesta está diseñada para interconectar robots heterogéneos en escenarios ROUD, incluyendo UAVs, UGVs y manipuladores, a través de redes de transceptores inalámbricos híbridos heterogéneos (H2WTN, por sus siglas en inglés). Además, la plataforma incluye sensores ad-hoc de alta capacidad de banda ancha 5G, incluyendo sistemas de Detección y Rango de Luz (LiDAR, por sus siglas en inglés) y smartphones modernos para obtener información sobre el entorno de los agentes y su posición dentro de él, permitiendo la sincronización de datos para operaciones tanto en línea como fuera de línea.

La H2WTN puede operar sin conectividad a Internet, proporcionando servicios e información confiables desde entornos ROUD durante períodos prolongados, incluso en ausencia de infraestructura tradicional. La integración de tecnologías PAN, WLAN y LPWAN habilita topologías de red diversas y adaptables, mientras que la introducción de técnicas de multicasting en línea y fuera de línea optimiza la eficiencia de comunicación y la adaptabilidad a escenarios dinámicos. La estrategia de utilizar robots móviles con un rol dual simultáneo ha demostrado ser altamente ventajosa en escenarios ROUD. En particular, el despliegue de UGVs y UAVs actuando como fuentes y sumideros móviles de datos ha sido validado experimentalmente. Sin embargo, se recomienda que los ZEDs solo cubran eventos a distancias medias de los agentes SAR. Zigbee puede ser útil, pero no es decisivo para la detección de víctimas semi-enterradas, ya que las pruebas indican que el ZC encargado de descubrir ZEDs perdidos debe estar demasiado cerca de ellos (al menos 20 m). No obstante, el UGV puede proporcionar más detecciones por proximidad, lo que ayuda a estimar la distancia a la víctima, mientras que el UAV permite un reconocimiento preliminar más rápido del terreno. Esto permite cubrir áreas más grandes y detectar incluso dispositivos semi-enterrados. En la H2WTN:

- El escáner BLE del UAV, con una línea de vista directa hacia posibles víctimas (PVs) en el suelo, demostró una mayor sensibilidad a las transmisiones que el escáner del UGV, que se ve más afectado por interferencias y obstáculos.

- La opción LPWAN permite la cobertura de grandes áreas mientras transmite paquetes de datos a intervalos regulares durante períodos prolongados de autonomía. La propuesta de extraer información a kilómetros de distancia del entorno operativo de los robots ha permitido el establecimiento de nodos concentradores con acceso a Internet en áreas alejadas del escenario ROUD.

- La H-WSN LoRa desplegada forma una configuración de *estrella de estrellas*, a diferencia de la red Zigbee, que es un conjunto de estrellas independientes que luego agregan la información recopilada. Como resultado, la red Zigbee proporciona información de manera más lenta, mientras que la red LoRa puede entregar información en tiempo real. En ambos casos, los datos sensoriales georreferenciados se procesan, lo que permite el seguimiento en tiempo real de los agentes que llevan nodos sensores.

- Las tecnologías PAN sirven como un sistema de percepción y detección cercano, mientras que las tecnologías LPWAN están diseñadas para recopilar datos desde ubicaciones remotas hacia el entorno operativo. En consecuencia, es esencial distinguir qué es útil transmitir y cómo hacerlo (e.g., formato, orden, tamaño de paquete, sincronización, consumo de energía, alcance de radio y sensibilidad), considerando la posición de los EDs.

En todos los casos presentados (BLE, Zigbee y LoRa), los nodos concentradores deben tener acceso a una red WLAN para poder transmitir la información a los centros de control remoto. En este sentido, se propone una red malla WiFi con nodos malla móviles para cubrir grandes áreas en el escenario ROUD. De esta manera, se ha logrado establecer una infraestructura de comunicaciones estable capaz de habilitar un intercambio bidireccional de información entre los agentes y los centros de control.

Sin embargo, combinar la información capturada por un equipo de agentes es necesario para construir una conciencia situacional compartida que respalde la toma de decisiones. Esta toma de decisiones (algoritmos) puede ser ejecutada por un operador humano remoto o de manera autónoma, es decir, llevada a cabo por cada robot. Para este fin, se presenta la arquitectura de Internet de Agentes Cooperativos (IoCA, por sus siglas en inglés), que ofrece un enlace inteligente entre percepción y acción a través de servicios de datos de baja latencia y alto ancho de banda proporcionados mediante redes de segmentación. Estos servicios se distribuyen entre dispositivos de computación Local- y Cloud-Edge desplegados en la

plataforma IoRT y un Sistema de Información de Retroalimentación (FIS, por sus siglas en inglés), respectivamente. Los dispositivos Local-Edge corresponden a los hosts locales en los entornos operativos de los agentes, mientras que los dispositivos Cloud-Edge son parte de los centros de Computación en el Borde de Múltiples Accesos (MEC, por sus siglas en inglés). Usando estos datos, los dispositivos Local-Edge actúan como Infraestructura como Servicio (IaaS), proporcionando y recibiendo servicios en el borde, operando cerca de los entornos operativos y conectándose con los robots a través de redes WLAN/LAN. De manera similar, los dispositivos Cloud-Edge permiten un mayor alcance e integración con recursos conectados a Internet, actuando como Plataforma como Servicio (PaaS). Además, para la comunicación, los agentes cuentan con uno o más dispositivos de Equipo de Usuario (UE, por sus siglas en inglés) (e.g., un router 5G o un smartphone 5G).

Por otro lado, el equipo de computación a bordo del robot puede necesitar en ocasiones algoritmos o servicios específicos que no están disponibles localmente, siendo necesario compartir recursos de computación pertenecientes a otros robots que operan en el mismo escenario (Edge) o a servidores alojados de forma remota (Cloud). Mientras que enviar datos sin procesar puede ser esencial para aplicaciones específicas, las limitaciones de ancho de banda a menudo exigen un preprocesamiento local para transmitir solo la información más relevante. Los centros MEC abordan estos desafíos distribuyendo inteligentemente las tareas de procesamiento entre los entornos operativos, incluidos los ordenadores a bordo de los agentes robóticos desplegados. Este enfoque ayuda a reducir las latencias operativas y apoya el cumplimiento de los requisitos en tiempo real de manera efectiva. Se presentan dos casos de uso de la arquitectura IoCA: una comparación entre un SLAM ejecutado en Edge y un SLAM ejecutado en Cloud; y un algoritmo de multilateración ejecutado en Cloud para posicionar a las víctimas detectadas en tiempo real.

El SLAM remoto (ejecutado en la Cloud) se implementa con éxito utilizando técnicas de segmentación de red, integradas en una red piloto 5G libre de clientes comerciales. Esta configuración permite realizar pruebas exhaustivas de la arquitectura de extremo a extremo bajo condiciones de saturación de ancho de banda, demostrando la capacidad del sistema para mantener el rendimiento y adaptarse a las demandas de tareas de alta tasa de datos. Esta versión en la Cloud ahorra consumo de energía y espacio en el robot, permitiéndole separar la construcción del mapa y la transmisión de los datos sensoriales sin procesar (el punto en nube). Cuando no es posible utilizar la Cloud, se puede ejecutar el SLAM en Edge. El algoritmo propuesto no ha generado cuellos de botella en el procesamiento.

Además, esta tesis explora el desarrollo de aplicaciones móviles para integrar smartphones en la robótica.

Se propone integrar H-WSN con sensores ad-hoc de alta capacidad de banda ancha para mejorar la Conciencia de Localización (LA, por sus siglas en inglés) de los robots, asegurando que su pose (posición y orientación) permanezca continuamente alineada con las mediciones ambientales. La inclusión de nodos estáticos y móviles aumenta la redundancia y flexibilidad del sistema, permitiendo un despliegue más efectivo en escenarios dinámicos. Esta diversa plataforma IoRT aborda limitaciones clave que enfrentan los robots en entornos ROUD, como la cobertura de Internet restringida, las limitaciones de ancho de banda y la latencia, lo que podría obstaculizar su capacidad para actuar de manera segura y efectiva. Además, se incorporan gemelos digitales en la arquitectura IoCA para garantizar la teleoperación segura de los robots ubicados en entornos ROUD.

Finalmente, se ha desarrollado software de código abierto como aplicaciones para apoyar a los agentes SAR en escenarios de desastre. La tabla 6.5 resume las aplicaciones creadas e implementadas en varios experimentos de campo realistas de emergencia. Todas las aplicaciones han sido diseñadas para ser compatibles con las arquitecturas ROS 1 y ROS 2, permitiendo la percepción, monitoreo y procesamiento de datos para los agentes SAR.

La inclusión de smartphones en escenarios ROUD ha facilitado el diseño de tareas robóticas al cumplir con una combinación de requisitos de comunicación móvil y procesamiento distribuido. Además, el acceso a la experimentación con una red piloto 5G, libre de usuarios comerciales, ha permitido el diseño de una plataforma IoRT adaptada a escenarios ROUD, incluyendo perspectivas sobre las huellas de datos requeridas por los agentes. Por ejemplo, ha sido posible gestionar la segmentación de ancho de banda de la red para construir SLAM en tiempo real de forma inalámbrica mientras se realizan simultáneamente otras tareas computacionalmente y comunicacionalmente intensivas.

Table 6.5 GitHub repositories for robotic applications in ROUD scenarios

Development	**Characteristics**	**Repository**
UMA-ROS-Android	ROS 1 integration in Android to sensor SAR agents	https://github.com/Juan013/uma-ros-android
UR2A	Remote command reception, sensory transmission through ROS 2, Fractal ArUco detection	https://github.com/Robotics-Mechatronics-UMA/UMA-ROS2-Android
Ad-hoc ROS Mobile	Digital twin visualization and control	https://github.com/Robotics-Mechatronics-UMA/Adhoc-ROS-Mobile
ILS algorithm for an FTM-based H-WSN	ROS 1 positioning system and tracking	https://github.com/Robotics-Mechatronics-UMA/ROUD-PositioningSystem.git
ILS algorithm for an FTM-based One Single Agent (OSA)	ROS 2 positioning system	https://github.com/Robotics-Mechatronics-UMA/OSA-PositioningSystem.git

La herramienta *SAR-FIS*, diseñada y evolucionada para integrar diferentes trabajos de investigación, ha sido desarrollada por el grupo de Robótica y Mecatrónica de la UMA, siendo el líder del desarrollo del código el investigador Manuel Toscano-Moreno, a quien

agradecemos su contribución. A lo largo de esta tesis, la herramienta *SAR-FIS* ha evolucionado para integrar tanto ROS 2 como ROS 1 simultáneamente, sirviendo como la principal Interfaz Gráfica de Usuario (GUI) en los centros de control para monitorear los datos de la red de sensores, cámaras integradas con ROS, el seguimiento de agentes e incluso la detección basada en RTT y la estimación de posición basada en multilateración. Sin embargo, programas y algoritmos específicos, como la sincronización offline/online de bases de datos alojadas en nodos concentradores, paquetes de configuración y monitoreo de redes PAN, WLAN y LPWAN, multilateración, control remoto de manipuladores, así como el desarrollo y control de gemelos digitales (desarrollados con la herramienta VEROSIM), monitoreo de tráfico de routers, ejecución remota de SLAM o control de segmentación de red, han sido desarrollados de manera independiente, distribuyendo la carga computacional entre diferentes hosts en la arquitectura IoCA.

Trabajos futuros

La siguiente sección describe posibles direcciones para la investigación y el desarrollo futuros, basándose en los hallazgos y contribuciones de esta tesis doctoral para avanzar aún más en el campo. Así, las futuras líneas son:

- **Fusionar datos de múltiples agentes equipados con cargas útiles sensoriales y de comunicación idénticas, que navegan por la misma área de interés**. Estos agentes pueden percibir el entorno desde diferentes contextos, tiempos y perspectivas. Integrar esta información colectiva en ROS 2 permitirá mejorar la interoperabilidad y el intercambio de datos, proporcionando entradas más ricas para algoritmos avanzados que se ejecuten en centros MEC (computación en la nube) para apoyar la toma de decisiones en tiempo real y la conciencia situacional.

- Incluir nuevas funcionalidades en la *herramienta SAR-FIS*. La herramienta debería escalarse gradualmente porque se necesitan múltiples operadores.

  Eventualmente, se podría crear una nueva entidad de inteligencia artificial (IA) para reemplazar a X-FIS en la arquitectura IoCA, que podría identificar los escenarios de aplicación y sus requisitos al ejecutarse. Esta IA-FIS podría ejecutar diferentes modelos generativos para el procesamiento autónomo, filtrado y monitoreo de datos, identificación de nuevos agentes, identificación de mejoras, etc.

  Todas estas posibles extensiones de la *herramienta SAR-FIS*, desarrolladas en MATLAB, podrían integrarse en una *toolbox* de MATLAB.

- **Explorar el uso de celdas móviles 6G de multitud para mejorar la conectividad en escenarios ROUD, permitiendo que los robots actúen como grandes MULEs de datos**.

- Mejorar la aplicación UR2A, que ya ha sido desarrollada para detectar y estimar la distancia a marcadores fractales fiduciarios. Las mejoras futuras podrían centrarse en aumentar la precisión de detección, ampliar la compatibilidad con tipos adicionales de marcadores e integrar algoritmos avanzados de localización basada en visión. Aprovechar esta aplicación para Android en tareas como el seguimiento de objetos, estimación de pose y mapeo cooperativo entre robots podría mejorar significativamente las capacidades de percepción e interacción en entornos ROUD.

- **Rastrear el estado de salud de los agentes humanos a través de sensores especializados conectados por Bluetooth al smartphone**, capturando parámetros fisiológicos, como la actividad electrodermal (EDA), el electrocardiograma (ECG), los niveles de oxígeno en sangre o la capacidad pulmonar. Esta información podría ser útil para detectar y evaluar el estrés de los sujetos (e.g., los rescatistas) que llevan los sensores[197–199], y se puede publicar implementando nodos de ROS.

- Implementar un método eficiente **para filtrar y distinguir dispositivos portátiles que probablemente lleven las víctimas de los irrelevantes**. Además, el sistema debería mejorarse para soportar la recopilación y visualización de datos BLE en tiempo real.

- **Optimizar la eficiencia de la red mallada dentro de túneles y escenarios subterráneos**, considerando diferentes configuraciones de red para lograr un mayor ancho de banda.

- **Integrar ROS-FIS en SAR-FIS**.

- **Aplicar una combinación de las dos estrategias basadas en FTM presentadas**. En este caso, se puede aprovechar el número de detecciones RTT de cada agente (como en la estrategia B), ya que no depende de *anchors* secundarios para aplicar la multilateración, siendo cada posición del *anchor* móvil (robot) una nueva posición desde la cual se calcula la multilateración. Si esto se utiliza para varios agentes móviles que llevan *anchors*, habrá diferentes estimaciones desde diferentes puntos, aprovechando la conciencia de localización (SA) de cada agente para localizar a una víctima que lleve un dispositivo WiFi. Los resultados de cada agente se comunicarían al FIS, desde donde se podría calcular un resultado que haga uso de las salidas de cada agente, logrando así una mayor precisión.

- **Distinguir entre señales WiFi de dispositivos no víctimas y aquellas de víctimas reales, ya que el sistema actualmente detecta todos los dispositivos WiFi**.
- Ampliar la implementación de SAR-IoCA con varias mejoras. Estas incluyen **integrar un manipulador móvil sincronizado con un servidor DT, utilizando el puente MQTT-ROS para la comunicación, habilitando teleoperaciones desde celdas distantes con baja latencia a través de redes 5G/6G, y desarrollar una aplicación que aborde diversos casos de uso dentro del marco de IoCA**.
- Avanzar en los desarrollos de IoRT-in-hand para aplicaciones médicas realistas integrándolos en escenarios de desastre a través de un manipulador móvil.
- **Se considera implementar un servidor de Protocolo de Transmisión en Tiempo Real (RTSP) para adquirir imágenes de ultrasonido de alta calidad**, reduciendo la carga de trabajo del enrutador móvil en el entorno operativo.
- **Integrar todos los servicios Edge-Cloud propuestos en la arquitectura IoCA en una plataforma web, haciendo que el PaaS sea gestionable de forma remota**. Esto evitaría la generación de un tráfico adicional cada vez que un dispositivo Local-Edge o Cloud-Edge se suscriba a cualquier información.

Bibliography

[1] Robin R Murphy. Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine*, 11(3):50–61, 2004.

[2] Jeffrey Delmerico, Stefano Mintchev, Alessandro Giusti, Boris Gromov, Kamilo Melo, Tomislav Horvat, Cesar Cadena, Marco Hutter, Auke Ijspeert, Dario Floreano, et al. The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191, 2019.

[3] Mary B Alatise and Gerhard P Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846, 2020.

[4] Nunzio Barone, Walter Brescia, Saverio Mascolo, and Luca De Cicco. Apeiron: a multimodal drone dataset bridging perception and network data in outdoor environments. In *Proceedings of the 15th ACM Multimedia Systems Conference*, pages 401–407, 2024.

[5] Selim Solmaz, Pamela Innerwinkler, Michał Wójcik, Kailin Tong, Elena Politi, George Dimitrakopoulos, Patrick Purucker, Alfred Höß, Björn W Schuller, and Reiner John. Robust robotic search and rescue in harsh environments: An example and open challenges. In *2024 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 1–8. IEEE, 2024.

[6] Albert Ko and Henry YK Lau. Robot assisted emergency search and rescue system with a wireless sensor network. *International Journal of Advanced Science and Technology*, 3(4):69–78, 2009.

[7] Chia-Yen Shih, Jesús Capitán, Pedro José Marrón, Antidio Viguria, Francisco Alarcón, Marc Schwarzbach, Maximilian Laiacker, Konstantin Kondak, José Ramiro Martínez-de Dios, and Aníbal Ollero. On the cooperation between mobile robots and wireless sensor networks. *Cooperative Robots and Sensor Networks 2014*, pages 67–86, 2014.

[8] Farrukh Pervez, Junaid Qadir, Mohsin Khalil, Touseef Yaqoob, Usman Ashraf, and Shahzad Younis. Wireless technologies for emergency response: A comprehensive review and some guidelines. *IEEE Access*, 6:71814–71838, 2018.

[9] Arturo Torres-González, José Ramiro Martinez-de Dios, and Aníbal Ollero. Robot-WSN cooperation for scalable simultaneous localization and mapping. In *Cooperative Robots and Sensor Networks 2014*, pages 25–41. Springer, 2014.

[10] Hongling Wang, Chengjin Zhang, Yong Song, Bao Pang, and Guangyuan Zhang. Three-Dimensional Reconstruction Based on Visual SLAM of Mobile Robot in Search and Rescue Disaster Scenarios. *Robotica*, 38(2):350–373, 2020.

[11] Wenbang Deng, Kaihong Huang, Xieyuanli Chen, Zhiqian Zhou, Chenghao Shi, Ruibin Guo, and Hui Zhang. Semantic RGB-D SLAM for rescue robot navigation. *IEEE Access*, 8:221320–221329, 2020.

[12] Björn Lindqvist, Christoforos Kanellakis, Sina Sharif Mansouri, Ali-akbar Aghamohammadi, and George Nikolakopoulos. Compra: A compact reactive autonomy framework for subterranean MAV based Search-and-Rescue operations. *Journal of Intelligent & Robotic Systems*, 105(3):49, 2022.

[13] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing: Platform and applications. In *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*, pages 73–78. IEEE, 2015.

[14] Muhammad Alee Khan, Nasir Saeed, Arbab Waheed Ahmad, and Chankil Lee. Location Awareness in 5G Networks Using RSS Measurements for Public Safety Applications. *IEEE Access*, 5:21753–21762, 2017.

[15] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing: A survey. *Future Generation Computer Systems*, 97:219–235, 2019.

[16] Vitor AM Jorge, Roger Granada, Renan G Maidana, Darlan A Jurak, Guilherme Heck, Alvaro PF Negreiros, Davi H Dos Santos, Luiz MG Gonçalves, and Alexandre M Amory. A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions. *Sensors*, 19(3):702, 2019.

[17] Barbara Arbanas Ferreira, Tamara Petrović, Matko Orsag, J Ramiro Martínez-de Dios, and Stjepan Bogdan. Distributed allocation and scheduling of tasks with cross-schedule dependencies for heterogeneous multi-robot teams. *IEEE Access*, 2024.

[18] Björn Lindqvist, Samuel Karlsson, Anton Koval, Ilias Tevetzidis, Jakub Haluška, Christoforos Kanellakis, Ali-akbar Agha-mohammadi, and George Nikolakopoulos. Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue. *Robotics and Autonomous Systems*, 154:104134, 2022.

[19] C Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann, Fabian Günther, Marco Tranzatto, Peter Fankhauser, and Marco Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.

[20] Jesús M Gómez-de Gabriel, Juan M Gandarias, Francisco J Pérez-Maldonado, Francisco J García-Núñcz, Emilio J Fernández-García, and Alfonso J García-Cerezo. Methods for autonomous wristband placement with a search-and-rescue aerial manipulator. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7838–7844. IEEE, 2018.

[21] Mizuki Nakajima and Motoyasu Tanaka. Motion control of a snake robot on multiple inclined planes. *Advanced Robotics*, 38(11):784–800, 2024.

[22] Zulkifli Mansor, Addie Irawan, and Mohammad Fadhil Abas. Evolution, design, and future trajectories on bipedal wheel-legged robot: A comprehensive review. *International Journal of Robotics and Control Systems*, 3(4):673–703, 2023.

[23] Sara Cooper, Alessandro Di Fava, Carlos Vivas, Luca Marchionni, and Francesco Ferro. Ari: The social assistive robot and companion. In *2020 29th IEEE International conference on robot and human interactive communication (RO-MAN)*, pages 745–751. IEEE, 2020.

[24] Justinas Mišeikis, Pietro Caroni, Patricia Duchamp, Alina Gasser, Rastislav Marko, Nelija Mišeikienė, Frederik Zwilling, Charles De Castelbajac, Lucas Eicher, Michael Früh, et al. Lio-a personal robot assistant for human-robot interaction and care applications. *IEEE Robotics and Automation Letters*, 5(4):5339–5346, 2020.

[25] Robin R Murphy and Jennifer L Burke. Up from the rubble: Lessons learned about hri from search and rescue. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 437–441. SAGE Publications Sage CA: Los Angeles, CA, 2005.

[26] Shahinul Hoque, Farhin Farhad Riya, and Jinyuan Sun. Hri challenges influencing low usage of robotic systems in disaster response and rescue operations. *arXiv preprint arXiv:2401.15760*, 2024.

[27] Partha Pratim Ray. Internet of Robotic Things: concept, technologies, and challenges. *IEEE Access*, 4:9489–9500, 2016.

[28] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, April 2021.

[29] Hriday Bavle, Jose Luis Sanchez-Lopez, Claudio Cimarelli, Ali Tourani, and Holger Voos. From SLAM to situational awareness: Challenges and survey. *Sensors*, 23(10):4849, 2023.

[30] Adrián Jiménez-González, José Ramiro Martínez-de Dios, and Anibal Ollero. An integrated testbed for cooperative perception with heterogeneous mobile and static sensors. *Sensors*, 11(12):11516–11543, 2011.

[31] Mrutyunjay Rout and Rajarshi Roy. Dynamic deployment of randomly deployed mobile sensor nodes in the presence of obstacles. *Ad hoc networks*, 46:12–22, 2016.

[32] Javier Vales-Alonso, Francisco J Parrado-García, Pablo López-Matencio, Juan J Alcaraz, and Francisco J González-Castaño. On the optimal random deployment of wireless sensor networks in non-homogeneous scenarios. *Ad Hoc Networks*, 11(3):846–860, 2013.

[33] Drew Gislason. *Zigbee wireless networking*. Newnes, 2008.

[34] Pengfei Li, Jiakun Li, Luhua Nie, and Bo Wang. Research and application of zigbee protocol stack. In *2010 International conference on measuring technology and mechatronics automation*, volume 2, pages 1031–1034. IEEE, 2010.

[35] Shigeyuki Tateno and Yuji Okamoto. Remote monitoring system for rescue operations with wireless sensor network. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pages 551–555. IEEE, 2014.

[36] Ming Tao, Xiaoyu Hong, Chao Qu, Jie Zhang, and Wenhong Wei. Fast access for zigbee-enabled iot devices using raspberry pi. In *2018 chinese control and decision conference (CCDC)*, pages 4281–4285. IEEE, 2018.

[37] Charles Tsui, Dale Carnegie, and Qing Wei Pan. USAR robot communication using ZigBee technology. In *Progress in Robotics: FIRA RoboWorld Congress 2009, Incheon, Korea, August 16-20, 2009. Proceedings 12*, pages 380–390. Springer, 2009.

[38] T. D., V. Vishnevsky Dinh, D. T. Le, R. Kirichek, and A. Koucheryavy. *Determination of Subscribers Coordinates using Flying Network for Emergencies*. 23rd International Conference on Advanced Communication Technology (ICACT) (pp. 1-10). IEEE, 2021.

[39] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yangm, H. Lu, Y. Cao, C. Xu, and F. Gao. Swarm of micro flying robots in the wild. In *Science Robotics vol*. 7, pp. eabm5954, 10.1126/scirobotics.abm59547, pp. eabm5954, 10.1126/scirobotics.abm5954, 2022.

[40] I. Maza and A. Ollero. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*, pages 221–230. Springer, TokyoSpringer, Tokyo, 2007.

[41] Z. Gosiewski and L. Ambroziak. Formation flight control scheme for unmanned aerial vehicles. In *Robot Motion and Control*, pages 331–340. Springer, London, 2012Springer, London, 2012, 2011.

[42] C. Valeriy and S. Ihor. *Method of the multi-UAV formation flight control*, volume 167. Archived, 2018.

[43] Manuel Sánchez-Montero, Manuel Toscano-Moreno, Juan Bravo-Arrabal, Javier Serón Barba, Pablo Vera-Ortega, Ricardo Vázquez-Martín, Juan Jesús Fernandez-Lozano, Anthony Mandow, and Alfonso García-Cerezo. Remote planning and operation of a UGV through ROS and commercial mobile networks. In *ROBOT2022: Fifth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pages 271–282. Springer, 2022.

[44] J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez. "Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty," in Remote Sensing (MDPI). *vol*, 2021.

[45] Olivier Sébastien, Fanilo Harivelo, and Didier Sébastien. Using general public connected devices for disasters victims location. In *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*, pages 1–4. IEEE, 2014.

[46] P. Davidson and R. Piché. *"A survey of selected indoor positioning methods for smartphones," IEEE Communications Surveys & Tutorials, vol. 19, no. 2*. pp. 1347–1370, 2016.

[47] Nizam Kuxdorf-Alkirata, Gerrit Maus, Mustafa Gemci, and Dieter Brückmann. A passive fingerprinting approach for device-free surveillance and localization applications using a bluetooth low energy infrastructure. In *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, pages 31–36. IEEE, 2020.

[48] A Alper Başak and Murat H Sazli. Accurate indoor localization with optimized fingerprinting algorithm. In *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)*, pages 149–152. IEEE, 2017.

[49] Fahad Alhomayani and Mohammad H Mahoor. Deep learning methods for fingerprint-based indoor positioning: A review. *Journal of Location Based Services*, 14(3):129–200, 2020.

[50] Elias Hatem, Sergio Fortes, Elizabeth Colin, Sara Abou-Chakra, Jean-Marc Laheurte, and Bachar El-Hassan. Accurate and low-complexity auto-fingerprinting for enhanced reliability of indoor localization systems. *Sensors*, 21(16):5346, 2021.

[51] Tim Kluge, Christin Groba, and Thomas Springer. Trilateration, fingerprinting, and centroid: taking indoor positioning with bluetooth le to the wild. In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 264–272. IEEE, 2020.

[52] Maria Fazio, Antonio Celesti, and Massimo Villari. Improving proximity detection of mesh beacons at the edge for indoor and outdoor navigation. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2020.

[53] Takuto Jikyo, Takahiro Yamanishi, Tomio Kamada, Ryo Nishide, Chikara Ohta, Kenji Oyama, and Takenao Ohkawa. A study on outdoor localization method based on deep learning using model-based received power estimation data of low power wireless tag. *IEICE Communications Express*, 8(12):524–529, 2019.

[54] Moisés Lodeiro-Santiago, Iván Santos-González, Pino Caballero-Gil, and Cándido Caballero-Gil. Secure system based on UAV and BLE for improving SAR missions. *Journal of Ambient Intelligence and Humanized Computing*, 11(8):3109–3120, 2020.

[55] Zhuorui Yang, James Schafer, and Aura Ganz. Disaster response: victims' localization using bluetooth low energy sensors. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–4. IEEE, 2017.

[56] Juan Bravo-Arrabal, Juan Jesús Fernández-Lozano, Javier Serón, Jose Antonio Gomez-Ruiz, and Alfonso García-Cerezo. Development and Implementation of a Hybrid Wireless Sensor Network of Low Power and Long Range for Urban Environments. *Sensors*, 21(2):567, 2021.

[57] Restrepo J. Xx itu radio regulations. In *IEEE ICRA Workshop on Open Source Software*, volume Retrieved from: https://www.itu.int/pub/R-REG-RR-2020, pages 1–6, Geneva, Switzerland, 2022.

[58] Rahul C Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad hoc networks*, 1(2-3):215–233, 2003.

[59] Yixuan Xu, Xi Chen, Anfeng Liu, and Chunhua Hu. A latency and coverage optimized data collection scheme for smart cities based on vehicular ad-hoc networks. *Sensors*, 17(4):888, 2017.

[60] Mohammed Al Mazaideh and Janos Levendovszky. A multi-hop routing algorithm for wsns based on compressive sensing and multiple objective genetic algorithm. *Journal of Communications and Networks*, 23(2):138–147, 2021.

[61] Onur Tekdas, Volkan Isler, Jong Hyun Lim, and Andreas Terzis. Using mobile robots to harvest data from sensor fields. *IEEE Wireless Communications*, 16(1):22–28, 2009.

[62] J-M Martinez-Caro and M-D Cano. IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study. *Wireless Communications and Mobile Computing*, 2019, 2019.

[63] Muhammad Asghar Khan, Ijaz Mansoor Qureshi, and Fahimullah Khanzada. A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET). *Drones*, 3(1):16, 2019.

[64] RF Manrique Balmaceda, Ricardo Vázquez-Martín, J Bravo Arrabal, Juan Jesús Fernández-Lozano, and Alfonso García-Cerezo. Sistema de comunicación de respaldo mediante tecnología lora con hardware y software abierto para aplicaciones de robótica de emergencias. In *XLII Jornadas de Automática*, pages 581–587. Universidade da Coruña, Servizo de Publicacións, 2021.

[65] Philip J Basford, Florentin MJ Bulot, Mihaela Apetroaie-Cristea, Simon J Cox, and Steven J Ossont. LoRaWAN for smart city IoT deployments: A long term evaluation. *Sensors*, 20(3):648, 2020.

[66] Salaheddin Hosseinzadeh, Naeem Ramzan, Mehdi Yousefi, Krystyna Curtis, and Hadi Larijani. Impact of spreading factor on LoRaWAN propagation in a metropolitan environment. In *2020 International Conference on UK-China Emerging Technologies (UCET)*, pages 1–4. IEEE, 2020.

[67] Abdul Waheed Khan, Abdul Hanan Abdullah, Mohammad Hossein Anisi, and Javed Iqbal Bangash. A comprehensive study of data collection schemes using mobile sinks in wireless sensor networks. *Sensors*, 14(2):2510–2548, 2014.

[68] Gurkan Tuna, V Cagri Gungor, and Kayhan Gulez. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, 13:54–68, 2014.

[69] Souaad Boussoufa-Lahlah, Fouzi Semchedine, and Louiza Bouallouche-Medjkoune. Geographic routing protocols for Vehicular Ad hoc NETworks (VANETs): A survey. *Vehicular Communications*, 11:20–31, 2018.

[70] Hussein Mroue, Benoît Parrein, Sofiane Hamrioui, Przemyslaw Bakowski, Abbass Nasser, Eduardo Motta Cruz, and Wilfried Vince. LoRa+: an extension of LoRaWAN protocol to reduce infrastructure costs by improving the Quality of Service. *Internet of Things*, page 100176, 2020.

[71] René Brandborg Sørensen, Dong Min Kim, Jimmy Jessen Nielsen, and Petar Popovski. Analysis of latency and MAC-layer performance for class a LoRaWAN. *IEEE Wireless Communications Letters*, 6(5):566–569, 2017.

[72] Jorge Navarro-Ortiz, Sandra Sendra, Pablo Ameigeiras, and Juan M Lopez-Soler. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things. *IEEE Communications Magazine*, 56(2):60–67, 2018.

[73] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the limits of LoRaWAN. *IEEE Communications magazine*, 55(9):34–40, 2017.

[74] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A survey of LoRaWAN for IoT: From technology to application. *Sensors*, 18(11):3995, 2018.

[75] Eugen Harinda, Salaheddin Hosseinzadeh, Hadi Larijani, and Ryan M Gibson. Comparative Performance Analysis of Empirical Propagation Models for LoRaWAN 868MHz in an Urban Scenario. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 154–159. IEEE, 2019.

[76] Hendrik Linka, Michael Rademacher, Osianoh Glenn Aliu, and Karl Jonas. *Path loss models for low-power wide-area networks: Experimental results using LoRa*. VDE Verlag, 2018.

[77] Alexandru Lavric and Valentin Popa. Performance evaluation of LoRaWAN communication scalability in large-scale wireless sensor networks. *Wireless Communications and Mobile Computing*, 2018, 2018.

[78] Lluís Casals, Bernat Mir, Rafael Vidal, and Carles Gomez. Modeling the energy performance of LoRaWAN. *Sensors*, 17(10):2364, 2017.

[79] Akram H Jebril, Aduwati Sali, Alyani Ismail, and Mohd Fadlee A Rasid. Overcoming limitations of LoRa physical layer in image transmission. *Sensors*, 18(10):3257, 2018.

[80] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. On the limits of LoRaWAN channel access. In *2016 International Conference on Engineering and Telecommunication (EnT)*, pages 10–14. IEEE, 2016.

[81] Tahsin CM Dönmez and Ethiopia Nigussie. Security of lorawan v1.1 in backward compatibility scenarios. *Procedia computer science*, 134:51–58, 2018.

[82] Woo-Jin Sung, Hyeong-Geun Ahn, Jong-Beom Kim, and Seong-Gon Choi. Protecting end-device from replay attack on lorawan. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 167–171. IEEE, 2018.

[83] Xueying Yang, Evgenios Karampatzakis, Christian Doerr, and Fernando Kuipers. Security vulnerabilities in lorawan. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 129–140. IEEE, 2018.

[84] David Gauthier, Paul Freedman, Gregory Carayannis, and Alfred Malowany. Interprocess communication for distributed robotics. *IEEE Journal on Robotics and Automation*, 3(6):493–504, 1987.

[85] Saeed H Alsamhi, Ou Ma, and Mohd Samar Ansari. Convergence of machine learning and robotics communication in collaborative assembly: mobility, connectivity and future perspectives. *Journal of Intelligent & Robotic Systems*, pages 1–26, 2019.

[86] Jon A Webb. Latency and bandwidth considerations in parallel robotics image processing. In *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, pages 230–239, 1993.

[87] Mehran Anvari, Tim Broderick, Harvey Stein, Trevor Chapman, Moji Ghodoussi, Daniel W Birch, Craig Mckinley, Patrick Trudeau, Sanjeev Dutta, and Charles H Goldsmith. The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery. *Computer Aided Surgery*, 10(2):93–99, 2005.

[88] Alfin Hikmaturokhman, Kalamullah Ramli, and Muhammad Suryanegara. Spectrum Considerations for 5G in Indonesia. In *2018 International Conference on ICT for Rural Development (IC-ICTRuDev)*, pages 23–28. IEEE, 2018.

[89] Wonil Roh. 5G mobile communications for 2020 and beyond-vision and key enabling technologies. *Key note: at IEEE WCNC*, 2014.

[90] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. Edge computing in 5G: A review. *IEEE Access*, 7:127276–127289, 2019.

[91] Ian F Akyildiz, Ahan Kak, and Shuai Nie. 6G and beyond: The future of wireless communications systems. *IEEE access*, 8:133995–134030, 2020.

[92] Baoling Qin, Qiao Luo, Yunshi Luo, Jianwei Zhang, Jianjie Liu, and Liangyun Cui. Research and Application of Key Technologies of Edge Computing for Industrial Robots. *Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020*, pages 2157–2164, 6 2020.

[93] James Kuffner. Cloud-enabled humanoid robots. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, Nashville TN, United States, Dec.*, 2010.

[94] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[95] Waymo. Waymo. https://waymo.com/. Accedido en junio de 2021.

[96] Rihab Chaâri, Fatma Ellouze, Anis Koubâa, Basit Qureshi, Nuno Pereira, Habib Youssef, and Eduardo Tovar. Cyber-physical systems clouds: A survey. *Computer Networks*, 108:260–278, 2016.

[97] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on automation science and engineering*, 12(2):398–409, 2015.

[98] Jiafu Wan, Shenglong Tang, Hehua Yan, Di Li, Shiyong Wang, and Athanasios V Vasilakos. Cloud robotics: Current status and open issues. *IEEE Access*, 4:2797–2807, 2016.

[99] Wuhui Chen, Yuichi Yaguchi, Keitaro Naruse, Yutaka Watanobe, Keita Nakamura, and Jun Ogawa. A study of robotic cooperation in cloud robotics: Architecture and challenges. *IEEE Access*, 6:36662–36682, 2018.

[100] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering*, 12(2):398–409, 2015.

[101] Rajesh Arumugam, Vikas Reddy Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, Foong Foo Kong, A Senthil Kumar, Kang Dee Meng, and Goh Wai Kit. Davinci: A cloud computing framework for service robots. In *2010 IEEE international conference on robotics and automation*, pages 3084–3089. IEEE, 2010.

[102] Luis Riazuelo, Moritz Tenorth, Daniel Di Marco, Marta Salas, Dorian Gálvez-López, Lorenz Mösenlechner, Lars Kunze, Michael Beetz, Juan D Tardós, Luis Montano, et al. Roboearth semantic mapping: A cloud enabled knowledge-based approach. *IEEE Transactions on Automation Science and Engineering*, 12(2):432–443, 2015.

[103] Koji Kamei, Shuichi Nishio, Norihiro Hagita, and Miki Sato. Cloud networked robotics. *IEEE Network*, 26(3):28–34, 2012.

[104] Mahbuba Afrin, Jiong Jin, Ashfaqur Rahman, Yu-Chu Tian, and Ambarish Kulkarni. Multi-objective resource allocation for edge cloud based robotic workflow in smart factory. *Future Generation Computer Systems*, 97:119–130, 2019.

[105] Leonardo Militano, Adriana Arteaga, Giovanni Toffetti, and Nathalie Mitton. The Cloud-to-Edge-to-IoT Continuum as an Enabler for Search and Rescue Operations. *Future Internet 2023, Vol. 15, Page 55*, 15:55, 1 2023.

[106] Giovanni Toffetti and Thomas Michael Bohnert. Cloud robotics with ROS. In *Robot Operating System (ROS) The Complete Reference (Volume 4)*, pages 119–146. Springer, 2019.

[107] Mohit Singh Gosain, Nikhil Aggarwal, and Rajeev Kumar. A Study of 5G and Edge Computing Integration with IoT- A Review. *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2023*, pages 705–710, 2023.

[108] Kaiyuan Chen, Faraz Sadeghi, and Ken Goldberg. FogROS: An Adaptive Framework for Automating Cloud and Fog Robotics Deployments. *arXiv preprint arXiv:2108.11355*, 2021.

[109] Kaiyuan Chen, Michael Laskey, Jeffrey Ichnowski, and Ken Goldberg. FogROS2: An Adaptive Platform for Cloud and Edge Robotics Using ROS 2. *arXiv preprint arXiv:2205.09778*, 2022.

[110] Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 20(4):2961–2991, 2018.

[111] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys and Tutorials*, 19(3):1628–1656, 2017.

[112] Huaxi Zhang and Lei Zhang. Cloud robotics architecture: trends and challenges. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 362–3625. IEEE, 2019.

[113] Alessio Botta, Luigi Gallo, and Giorgio Ventre. Cloud, Fog, and Dew Robotics: Architectures for next generation applications. In *2019 7th IEEE international conference on mobile cloud computing, services, and engineering (MobileCloud)*, pages 16–23. IEEE, 2019.

[114] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 98:289–330, 2019.

[115] G. Valecce, S. Strazzella, and L.A. Grieco. On the interplay between 5G, mobile edge computing and robotics in smart agriculture scenarios. *Lecture Notes in Computer Science*, 11803 LNCS:549–559, 2019.

[116] Albert Ko, Henry YK Lau, and Rex PS Sham. Application of distributed wireless sensor network on humanitarian search and rescue systems. In *2008 Second International*

Conference on Future Generation Communication and Networking, volume 2, pages 328–333. IEEE, 2008.

[117] Josh D Freeman, Vinu Omanan, and Maneesha V Ramesh. Wireless integrated robots for effective search and guidance of rescue teams. In *2011 Eighth International Conference on Wireless and Optical Communications Networks*, pages 1–5. IEEE, 2011.

[118] Shaik Shabana Anjum, Rafidah Md Noor, and Mohammad Hossein Anisi. Review on manet based communication for search and rescue operations. *Wireless personal communications*, 94(1):31–52, 2017.

[119] Tasneem Darwish, Kamalrulnizam Abu Bakar, and Ahlam Hashim. Green geographical routing in vehicular ad hoc networks: Advances and challenges. *Computers & Electrical Engineering*, 64:436–449, 2017.

[120] Fahad Taha Al-Dhief, Naseer Sabri, S Fouad, NM Abdul Latiff, and Musatafa Abbas Abbood Albader. A review of forest fire surveillance technologies: Mobile ad-hoc network routing protocols perspective. *Journal of King Saud University-Computer and Information Sciences*, 31(2):135–146, 2019.

[121] Wang Liu, Kejie Lu, Jianping Wang, Guoliang Xing, and Liusheng Huang. Performance analysis of wireless sensor networks with mobile sinks. *IEEE transactions on vehicular technology*, 61(6):2777–2788, 2012.

[122] Imad Jawhar, Nader Mohamed, Jameela Al-Jaroodi, and Sheng Zhang. Data communication in linear wireless sensor networks using unmanned aerial vehicles. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 492–499. IEEE, 2013.

[123] Carlos Alberto Socarrás Bertiz, Juan Jesús Fernández Lozano, Jose Antonio Gomez-Ruiz, and Alfonso García-Cerezo. Integration of a mobile node into a hybrid wireless sensor network for urban environments. *Sensors*, 19(1):215, 2019.

[124] Zhang Lin, Huan-Chao Keh, Ruikun Wu, and Diptendu Sinha Roy. Joint data collection and fusion using mobile sink in heterogeneous wireless sensor networks. *IEEE Sensors Journal*, 21(2):2364–2376, 2020.

[125] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. *IEEE communications standards magazine*, 1(2):70–76, 2017.

[126] Shunliang Zhang. An overview of network slicing for 5G. *IEEE Wireless Communications*, 26(3):111–117, 2019.

[127] Q. V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W. J. Hwang, and Z. Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8:116974–117017, 2020.

[128] Anis Koubaa, Maram Alajlan, and Basit Qureshi. Roslink: bridging ros with the internet-of-things for cloud robotics. In *Robot Operating System (ROS)*, pages 265–283. Springer, 2017.

[129] Raffaele Limosani, Alessandro Manzi, Laura Fiorini, Paolo Dario, and Filippo Cavallo. Connecting ros and fiware: Concepts and tutorial. In *Robot operating system (ROS)*, pages 449–475. Springer, 2019.

[130] M. Marchese, A. Moheddine, and F. Patrone. UAV and satellite employment for the internet of things use case. In *IEEE Aerospace Conference Proceedings*, 2020.

[131] R. C. Mello, S. D. Sierra, M. Munera, C. A. Cifuentes, M. R. N. Ribeiro, and A. Frizera-Neto. Cloud robotics experimentation testbeds: A cloud-based navigation case study. 2019.

[132] J. Ramos, R. Ribeiro, D. Safadinho, J. Barroso, C. Rabadão, and A. Pereira. Distributed architecture for unmanned vehicle services. *Sensors*, 21(4):1–33, 2021.

[133] Abderrahime Filali, Amine Abouaomar, Soumaya Cherkaoui, Abdellatif Kobbane, and Mohsen Guizani. Multi-access edge computing: A survey. *IEEE Access*, 8:197017–197046, 2020.

[134] Robin R Murphy. *Disaster robotics*. MIT press, 2017.

[135] Shinji Kawatsuma, Mineo Fukushima, and Takashi Okada. Emergency response by robots to Fukushima-Daiichi accident: summary and lessons learned. *Industrial Robot: An International Journal*, 39(5):428–435, 2012.

[136] Yu Yamauchi, Yuichi Ambe, Hikaru Nagano, Masashi Konyo, Yoshiaki Bando, Eisuke Ito, Solvi Arnold, Kimitoshi Yamazaki, Katsutoshi Itoyama, Takayuki Okatani, et al. Development of a continuum robot enhanced with distributed sensors for search and rescue. *Robomech Journal*, 9(1):8, 2022.

[137] Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Visual-thermal landmarks and inertial fusion for navigation in degraded visual environments. In *2019 IEEE Aerospace Conference*, pages 1–9. IEEE, 2019.

[138] Rubén González-Navarro, Da-hui Lin-Yang, Ricardo Vázquez-Martín, Alfonso José García-Cerezo, et al. Disaster Area Recognition from Aerial Images with Complex-Shape Class Detection. In *2023 IEEE International Conference on Safety, Security, and Rescue Robotics (SSRR), Fukushima, Japan*, 2023.

[139] Simon Schwaiger, Lucas Muster, Georg Novotny, Michael Schebek, Wilfried Wöber, Stefan Thalhammer, and Christoph Böhm. Ugv-cbrn: An unmanned ground vehicle for chemical, biological, radiological, and nuclear disaster response. *arXiv preprint arXiv:2406.14385*, 2024.

[140] Maira Saboia, Lillian Clark, Vivek Thangavelu, Jeffrey A Edlund, Kyohei Otsu, Gustavo J Correa, Vivek Shankar Varadharajan, Angel Santamaria-Navarro, Thomas Touma, Amanda Bouman, et al. Achord: Communication-aware multi-robot coordination with intermittent connectivity. *IEEE Robotics and Automation Letters*, 7(4):10184–10191, 2022.

[141] Achilleas Santi Seisa, Sumeet Gajanan Satpute, and George Nikolakopoulos. A kubernetes-based edge architecture for controlling the trajectory of a resource-constrained aerial robot by enabling model predictive control. *arXiv e-prints*, pages arXiv–2301, 2023.

[142] Przemyslaw Polewski, Wei Yao, Lin Cao, and Sha Gao. Marker-free coregistration of UAV and backpack LiDAR point clouds in forested areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:307–318, 2019.

[143] Novak Zagradjanin, Aleksandar Rodic, Dragan Pamucar, and Bojan Pavkovic. Cloud-based multi-robot path planning in complex and crowded environment using fuzzy logic and online learning. *Information Technology and Control*, 50(2):357–374, 2021.

[144] Yujing Chen, Zheng Chai, Yue Cheng, and Huzefa Rangwala. Asynchronous federated learning for sensor data with concept drift. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4822–4831. IEEE, 2021.

[145] Hengjing He, Supun Kamburugamuve, Geoffrey C Fox, and Wei Zhao. Cloud based real-time multi-robot collision avoidance for swarm robotics. *International Journal of Grid and Distributed Computing*, 9(6):339–358, 2016.

[146] Kaiyuan Chen, Ryan Hoque, Karthik Dharmarajan, Edith LLontopl, Simeon Adebola, Jeffrey Ichnowski, John Kubiatowicz, and Ken Goldberg. FogROS2-SGC: A ROS2 cloud robotics platform for secure global connectivity. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2023.

[147] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. Exploring the performance of ROS2. In *Proceedings of the 13th international conference on embedded software*, pages 1–10, 2016.

[148] Jeffrey Ichnowski, Kaiyuan Chen, Karthik Dharmarajan, Simeon Adebola, Michael Danielczuk, Víctor Mayoral-Vilches, Nikhil Jha, Hugo Zhan, Edith LLontop, Derek Xu, et al. Fogros2: An adaptive platform for cloud and fog robotics using ros 2. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 5493–5500. IEEE, 2023.

[149] Kaiyuan Chen, Kush Hari, Trinity Chung, Michael Wang, Nan Tian, Christian Juette, Jeffrey Ichnowski, Liu Ren, John Kubiatowicz, Ion Stoica, et al. FogROS2-FT: Fault Tolerant Cloud Robotics. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1390–1397. IEEE, 2024.

[150] António Ruano, Sérgio Silva, Helder Duarte, and Pedro M Ferreira. Wireless sensors and iot platform for intelligent hvac control. *Applied Sciences*, 8(3):370, 2018.

[151] JJ Fernández-Lozano, Miguel Martín-Guzmán, Juan Martín-Ávila, and Alfonso García-Cerezo. A wireless sensor network for urban traffic characterization and trend monitoring. *Sensors*, 15(10):26143–26169, 2015.

[152] Juan Bravo-Arrabal, Pablo Zambrana, Juan Jesús Fernández-Lozano, José Antonio Gómez-Ruiz, Javier Serón Barba, and Alfonso García-Cerezo. Realistic deployment of hybrid wireless sensor networks based on ZigBee and LoRa for search and Rescue applications. *IEEE Access*, 10:64618–64637, 2022.

[153] Homepage Libelium. *(accessed February, 2021). [Online].* Available: http://www.libelium.com, 2020.

[154] Digi International Inc. Xctu - next-generation configuration platform for xbee/rf solutions, 2024.

[155] Lorenzo Parri, Stefano Parrino, Giacomo Peruzzi, and Alessandro Pozzebon. Low power wide area networks (lpwan) at sea: Performance analysis of offshore data

transmission by means of lorawan connectivity for marine monitoring applications. *Sensors*, 19(14):3239, 2019.

[156] Juan Cantizani-Estepa, Juan Bravo-Arrabal, JJ Fernandez-Lozano, Sergio Fortes, Raquel Barco, and Alfonso García-Cerezo. BLE-based detection system by UAV and UGV: a SAR use case. In *submitted to IEEE International Conference on Robotics and Automation*, Philadelphia (EEUU), 2022.

[157] J. Bravo-Arrabal, C. S. Alvarez-Merino, J. J. Fernández-Lozano, J. A. Gómez-Ruiz, A. Mandow, R. Barco, and A. Garcia-Cerezo. Field Report on Experimental Comparison of a WiFi Mesh Network Against Commercial 5G in an Underground Disaster Environment, booktitle = Proceedings of the 2023 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). pages 186–192. IEEE, November 2023.

[158] Jens-Rainer Ohm, Gary J Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand. Comparison of the coding efficiency of video coding standards—including high efficiency video coding (hevc). *IEEE Transactions on circuits and systems for video technology*, 22(12):1669–1684, 2012.

[159] The Things Network. The things network documentation, 2024. Accessed: December 3, 2024.

[160] Anderson M Da Rocha, Marcos A De Oliveira, Pauletti José FM, and Gerson Geraldo H Cavalheiro. Abp vs. otaa activation of lora devices: an experimental study in a rural context. In *2023 International Conference on Computing, Networking and Communications (ICNC)*, pages 630–634. IEEE, 2023.

[161] The Things Network. Adaptive data rate. https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/, n.d. Accessed: December 10, 2024.

[162] Manuel Toscano-Moreno, Anthony Mandow, María Alcázar Martínez, and Alfonso García-Cerezo. DEM-AIA: Asymmetric inclination-aware trajectory planner for off-road vehicles with digital elevation models. *Engineering Applications of Artificial Intelligence*, 121, 5 2023.

[163] J. Bravo-Arrabal, M. Toscano-Moreno, J. J. Fernandez-Lozano, A. Mandow, J. A. Gomez-Ruiz, and A. García-Cerezo. "The Internet of Cooperative Agents Architecture (X-IoCA) for Robots, Hybrid Sensor Networks, and MEC Centers in Complex

Environments: A Search and Rescue Case Study," Sensors, vol. 21. *no*, 10(3390), 2021.

[164] Manuel Toscano-Moreno, Juan Bravo-Arrabal, Manuel Sanchez-Montero, Javier Seron Barba, Ricardo Vazquez-Martin, J. J. Fernandez-Lozano, Anthony Mandow, and Alfonso Garcia-Cerezo. Integrating ROS and Android for Rescuers in a Cloud Robotics Architecture: Application to a Casualty Evacuation Exercise. *SSRR 2022 - IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 270–276, 2022.

[165] Manuel Córdoba Ramos, Juan Bravo Arrabal, Juan Jesus Fernandez Lozano, Anthony Mandow, and Alfonso García Cerezo. UR2A: Comunicación Bidireccional Android-ROS 2 para Arquitecturas Edge-Cloud en Sistemas Robóticos Conectados. *Jornadas de Automática*, (45), 2024.

[166] Hyeong Ryeol Kam, Sung-Ho Lee, Taejung Park, and Chang-Hun Kim. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60(2):337–345, 2015.

[167] Sami Salama Hussen Hajjaj and Khairul Saleh Mohamed Sahari. Establishing remote networks for ros applications via port forwarding: A detailed tutorial. *International Journal of Advanced Robotic Systems*, 14(3):1729881417703355, 2017.

[168] Silvia Lins, Kleber Vieira Cardoso, Cristiano Bonato Both, Luciano Mendes, José F De Rezende, Antonio Silveira, Neiva Linder, and Aldebaro Klautau. Artificial intelligence for enhanced mobility and 5g connectivity in uav-based critical missions. *IEEE Access*, 9:111792–111801, 2021.

[169] Jesús Morales, Ricardo Vázquez-Martín, Anthony Mandow, David Morilla-Cabello, and Alfonso García-Cerezo. The UMA-SAR Dataset: Multimodal data collection from a ground vehicle during outdoor disaster response training exercises. *The International Journal of Robotics Research*, 40(6-7):835–847, 2021.

[170] Davide Borsatti, Chiara Grasselli, Chiara Contoli, Luigia Micciullo, Luca Spinacci, Marina Settembre, Walter Cerroni, and Franco Callegati. Mission critical communications support with 5g and network slicing. *IEEE Transactions on Network and Service Management*, 20(1):595–607, 2022.

[171] Amartya Mukherjee, Debashis De, Nilanjan Dey, Rubén González Crespo, and Enrique Herrera-Viedma. Disastdrone: A disaster aware consumer internet of drone

things system in ultra-low latent 6g network. *IEEE Transactions on Consumer Electronics*, 69(1):38–48, 2023.

[172] Juan Bravo-Arrabal, Ricardo Vázquez-Martín, JJ Fernández-Lozano, and Alfonso García-Cerezo. Strengthening Multi-Robot Systems for SAR: Co-Designing Robotics and Communication Towards 6G. *arXiv preprint arXiv:2504.01940*, 2025.

[173] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2):1729881419841532, 2019.

[174] Carlos Simón Álvarez-Merino, Hao Qiang Luo Chen, Emil Jatib Khatib, Raquel Barco-Moreno, et al. Aplicación móvil para localización de interior mediante fusión de tecnologías. 2021.

[175] Peter JG Teunissen and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*, volume 10. Springer, 2017.

[176] Google. Google wifi mesh. Wireless mesh networking system, 2023. Accessed: 2024-11-26.

[177] Miguel Grinberg. *Flask web development*. " O'Reilly Media, Inc.", 2018.

[178] Tabea Dreyer, Amir Haluts, Amos Korman, Nir Gov, Ehud Fonio, and Ofer Feinerman. Comparing cooperative geometric puzzle solving in ants versus humans. *Proceedings of the National Academy of Sciences*, 122(1):e2414274121, 2025.

[179] Manuel Toscano-Moreno, Juan Bravo-Arrabal, Manuel Sánchez-Montero, Javier Serón Barba, Ricardo Vázquez-Martín, JJ Fernandez-Lozano, Anthony Mandow, and Alfonso Garcia-Cerezo. Integrating ROS and Android for rescuers in a cloud robotics architecture: application to a casualty evacuation exercise. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 270–276, 2022.

[180] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. ROS: an open-source Robot Operating System. In *IEEE ICRA Workshop on Open Source Software*, volume 3, pages 1–6, Kobe, Japan, 2009.

[181] M. Berrocoso, Raúl Páez, Bismarck Jigena, and C Caturla. The rap net: A geodetic positioning network for andalusia (south spain). *EUREF Publication*, pages 364–368, 01 2006.

[182] G. Ruiz-Mudarra, J. Bravo-Arrabal, and J. J. Fernández-Lozano. UMA-ROS-Android repository. Accessed: Jul 6, 2022.

[183] International Organization for Standardization. EN ISO 9409-1:2004 Manipulating Industrial Robots, Mechanical Interfaces. ISO, 2004.

[184] Oliver Zettinig, Benjamin Frisch, Salvatore Virga, Marco Esposito, Anna Rienmüller, Bernhard Meyer, Christoph Hennersperger, Yu-Mi Ryang, and Nassir Navab. 3d ultrasound registration-based visual servoing for neurosurgical navigation. *International journal of computer assisted radiology and surgery*, 12:1607–1619, 2017.

[185] Francisco Pastor, Francisco J Ruiz-Ruiz, Jesús M Gómez-de Gabriel, and Alfonso J García-Cerezo. Autonomous Wristband Placement in a Moving Hand for Victims in Search and Rescue Scenarios With a Mobile Manipulator. *IEEE Robotics and Automation Letters*, 7(4):11871–11878, 2022.

[186] Sergio Garrido-Jurado, Rafael Munoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern recognition*, 51:481–491, 2016.

[187] Jesús Morales, Isabel Castelo, Rodrigo Serra, Pedro U Lima, and Meysam Basiri. Vision-based autonomous following of a moving platform and landing for an unmanned aerial vehicle. *Sensors*, 23(2):829, 2023.

[188] Nils Rottmann, Nico Studt, Floris Ernst, and Elmar Rueckert. ROS-Mobile: An Android application for the robot operating system. *arXiv preprint arXiv:2011.02781*, 2020.

[189] Alberto Sartori, Ralf Waspe, and Christian Schlette. Mqtt enabled simulation interface for motion execution of industrial robots. In *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*, pages 28–32. IEEE, 2023.

[190] Wenwei Lin, Peidong Liang, Guantai Luo, Ziyang Zhao, and Chentao Zhang. Research of online hand–eye calibration method based on charuco board. *Sensors*, 22(10):3805, 2022.

[191] Lennart Reiher, Bastian Lampe, Timo Woopen, Raphael Van Kempen, Till Beemelmanns, and Lutz Eckstein. Enabling connectivity for automated mobility: a novel MQTT-based interface evaluated in a 5G case study on edge-cloud lidar object detection. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–9. IEEE.

[192] Yi-Chun Du, Jheng-Bang Shih, Ming-Jui Wu, and Chung-Yi Chiou. Development of an AVF stenosis assessment tool for hemodialysis patients using robotic ultrasound system. *Micromachines*, 9(2):51, 2018.

[193] Qinghua Huang, Jiulong Lan, and Xuelong Li. Robotic arm based automatic ultrasound scanning for three-dimensional imaging. *IEEE Transactions on Industrial Informatics*, 15(2):1173–1182, 2018.

[194] Anders Prier Lindvig, Iñigo Iturrate, Uwe Kindler, and Christoffer Sloth. ur_rtde: An Interface for Controlling Universal Robots (UR) using the Real-Time Data Exchange (RTDE). In *2025 IEEE/SICE International Symposium on System Integration (SII)*, pages 1118–1123. IEEE, 2025.

[195] Jürgen Rossmann, Michael Schluse, Christian Schlette, Ralf Waspe, J Van Impe, and F Logist. A new approach to 3D simulation technology as enabling technology for erobotics. In *1st International Simulation Tools Conference & EXPO*, volume 2013, pages 39–46, 2013.

[196] Carlos Baena, Sergio Fortes, Oswaldo Peñaherrera, and Raquel Barco. A framework to boost the potential of network-in-a-box solutions. In *12th International Conference on Network of the Future (NoF)*, pages 1–3. IEEE, 2021.

[197] Yun Liu and Siqing Du. Psychological stress level detection based on electrodermal activity. *Behavioural brain research*, 341:50–53, 2018.

[198] Ravi Bhoja, Oren T Guttman, Amanda A Fox, Emily Melikman, Matthew Kosemund, and Kevin J Gingrich. Psychophysiological stress indicators of heart rate variability and electrodermal activity with application in healthcare simulation research. *Simulation in Healthcare*, 15(1):39–45, 2020.

[199] Luca Menghini, Evelyn Gianfranchi, Nicola Cellini, Elisabetta Patron, Mariaelena Tagliabue, and Michela Sarlo. Stressing the accuracy: Wrist-worn wearable sensor validation over different conditions. *Psychophysiology*, 56(11):e13441, 2019.

[200] Zhuoqi Cheng, Bence Mány, Kasper Balsby Jørgensen, Siheon An, Marcus Leander Jensen, Richard Thulstrup, Habib Frost, Thiusius R. Savarimuthu, and Olof Huldt. Portable robot for needle insertion assistance to femoral artery. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2407–2413, 2024.

Appendix A

Cooperative Agents

The following is a detailed list and description of the robotic and human agents integrated into the experimental part of this PhD thesis through the proposed Internet of Cooperative Agents (IoCA) architecture.

A.1 Robots

A.1.1 Cuadriga

Cuadriga (shown in Figure A.1 is a four-wheeled brushless-motors skid-steering mobile robot developed by our research group from scratch. This robot weighs 83.1 kg with dimensions 0.82 m (length) × 0.64 m (width) × 0.81 m (height) and has a payload of 120 kg. The distance between the front and rear wheel contact points is 0.475 m, while the distance between left and right is half a meter. Regarding the wheels, they have pneumatic 35.5 cm diameter tires with rigid suspension. Motors are controlled by an embedded board that integrates a microcontroller and two independent H-bridge power stages. The maximum linear speed is 1.2 m/s. Power is supplied by a 36V battery pack composed of six 12V lead-acid batteries connected in a series-parallel configuration.

Cuadriga's EDs include an IMU, GNSS module with RTK differential corrections, a pan-tilt-zoom (PTZ) camera, a LiDAR system, and H2WTN nodes, including environmental sensor nodes from the LoRa H-WSN (such as temperature, humidity, or gases probes) and BLE scanners and transmitters. Cuadriga embarks two different types of BLE scanners: one low-cost based on the ESP32 microcontroller and another commercial one (Meshlium Scanner) from Libelium, for short-range event detection.

A compact onboard computer performs data acquisition and high-level motion control using a LabVIEW program.

(a) (b)

Figure A.1 Cuadriga in ROUD scenarios

This computer interfaces with the embedded motor controller and the current sensors through two serial links. Manual operation is possible through a wireless joystick.

Communication with a tele-operation station is done using a wireless router (WiFi, 2.4 GHz) or via the Internet via 5G. In addition, the robot embarks other radios, such as Bluetooth, LoRa, ZigBee, 3G, 4G, and 900 MHz.

Cuadriga embarks the switch Netgear MS510TXPP Netgear 8-Port Multi-Gigabit PoE to interconnect all their elements, using the same 5G CPE (Huawei 5G CPE Pro, model H122-373), from Huawei to access the Internet.

A.1.2 Rambler

The Rambler robot (see Figure A.2) is the natural evolution of Cuadriga robot. This UGV has been designed and developed by the Robotics and Mechatronics Lab of the Univesity of Malaga from scratch. Rambler is a 4-motor-wheel skid-steer mobile robot, with independent pneumatic active suspension provided by double-acting cylinders and pressure controllers, two for each cylinder. Moreover, a pneumatic compressor supplies the required pressure.

Rambler weighs 460 kg with dimensions 1.6 m (length) × 1.2 m (width) × 0.66 m (height) and has a payload of more than 320 kg. The distance between the front and rear wheel contact points goes from 0.95 to 1 m. Distance between left and right wheel contact points is 1.05 m. Regarding the wheels, they have pneumatic 50 cm diameter tyres. Maximum linear speed is 22 m/s (80 km/h).

This UGV, and all its elements, are powered by 64 lithium iron $LiFePO\_4$ batteries forming 16SP4 cells with their corresponding charge, discharge, and temperature monitoring controller.

Figure A.2 Rambler acting as a mobile sink

Rambler's EDs include an IMU, GNSS module with RTK differential corrections, a pan-tilt-zoom (PTZ) camera, a LiDAR system, and H2WTN nodes, including environmental sensor nodes from the LoRa H-WSN (such as temperature, humidity, or gases probes) and BLE scanners and transmitters.

Moreover, Rambler embarks a LoRa concentrator-node based on the Conduit IP67 Base Station (Multitech MTCDTIP 220L), from Multitech, which hosts an MQTT broker, for long-range gathering and event detection (see chapter 3).

Rambler has two PCs with two different operating systems (Windows and Ubuntu) being able to work with both at the same time of it is desired for different purposes. Each CPE operates with a SIM card with a static public IP address in order to facilitate the network configuration, incoming connections, and avoiding Network Address Translation (NAT) problems. This is useful since Rambler makes use of a LabVIEW program (in Windows) for data acquisition (speed, batteries, controller parameters, etc) and high level motion control. The onboard computer interfaces with the embedded motor controller and the current sensors through two serial links. Manual operation is possible through a wireless joystick. In addition, the LabVIEW project includes an MQTT client to connect to the MQTT brokers (hosted in the *Cloud-* and in some *Local-Edge*) devices, allowing the access robot information remotely. Establishing an static IP address for the CPEs of both UGVs it is simple to access to the sockets (IP and port) from the MEC centers (BCC and FCC). In addition, the static IP address simplifies the ROS nodes communication (especially, in ROS 1), since nodes only needs to know where are the rest of ROS nodes in the network (WAN). Moreover, User Datagram

Protocol (UDP) connections was integrated in the LabVIEW project (see Figure 5.2) for real-time needs.

Communication with a tele-operation station is done using a wireless router (via WiFi, 2.4 GHz), or through Internet via 5G. In addition, the robot embarks other radios, such as Bluetooth, LoRa, ZigBee, 3G, 4G, and 902-928 MHz radio link.

Rambler embarks the switch Netgear MS510TXPP Netgear 8-Port Multi-Gigabit PoE to interconnect all their elements, using the same 5G CPE (Huawei 5G CPE Pro, model H122-373), from Huawei to access the Internet.

A.1.3 Rover J8

Rover J8 (see Figure A.3) is a commercial robotic platform developed by Argo. This robot has the same differential-drive locomotion system as Cuadriga and Rambler but is based on eight wheels with rubber tires. Rover J8 is an electric off-road 8×8 UGV designed for outdoor navigation, with amphibious capabilities, a weight of 1090 kg, and can carry up to 600 kilos of payload. The Robotics and Mechatronics Lab has modified the UGV since it was acquired in 2019, including new autonomous features. In addition, Rover J8 can navigate through a follow-me mode or via tele-operation. This UGV is powered by a 46 V battery with an autonomy of several hours, which is ideal for evacuating victims. In addition, two stretchers are available on the robot to support SAR missions.

Rover J8 includes a LiDAR system to enable a follow-me navigation mode. Two PCs with Ubuntu 20.04 and ROS Noetic installed were mounted to perform the processing of heavy tasks, such as acquiring and applying algorithms on the point cloud captured by the LiDAR. Finally, a printed part was created to install a 5G smartphone on the front of this robot to transmit images via 5G during its navigation. In this case, the transmission of images was performed by using the UMA-ROS-Android applications.

(a) (b)

Figure A.3 Rover J8 acting as an evacuation platform.

A.1.4 DJI Matrice 600

The DJI Matrice 600 (see Figure A.4) is equipped with a dedicated triple-modular redundancy system composed of three GNSS modules and IMUs. This UAV is an hexacopter drone assembled with six carbon fiber rotors, quick-release landing gear, and mounted folding arms. This robotic agent wights 9.5 kg, and its maximum takeoff weight is 15.5 kg. The UAV uses six standard intelligent flight batteries with a capability of 4500 mAh, voltage of 22.2 V, and built-in smart charge–discharge function. In this case, the 5G communication is implemented by means of the router Inseego 5G MiFi M2000, from Inseego. In addition, this UAV carried a LoRa ED, included in the H2WTN [64], and a CN (based on the module iC880A) using as host a Raspberry Pi 4B for long-range event detection, as well as an ESP32-based BLE scanner for short-distance detection of victims [156].

As for Rover J8 robot, a 3D-printed piece was also designed to embed a 5G smartphone to the UAV in order to transmit images via 5G during its flights.

The DJI Matrice 600 (M600) is a high-end drone designed primarily for professional applications such as aerial filming, surveying, inspection, and mapping. Below is a detailed description of its main technical features:

The Matrice 600 has a diagonal size of 1133 mm (with propellers) and a height of 527 mm. Its empty weight is approximately 9.5 kg, excluding batteries and payload. The drone can support a maximum takeoff weight of 15.5 kg and carry up to 6.0 kg of payload, making it highly versatile for carrying various professional camera systems and sensors.

When it comes to flight time, the M600 offers different durations based on the payload. It can fly for up to 16 minutes with a full 6 kg payload and up to 35 minutes when carrying no additional weight, using the standard TB47S batteries.

(a) (b)

Figure A.4 DJI Matrice 600 carring an ad-hoc 5G sensor

The drone's propulsion system is powered by DJI 6010 brushless motors paired with 17-inch carbon fiber propellers, which provide reliable power and efficiency during flight.

The M600 uses six intelligent TB47S (4500 mAh) or TB48S (5700 mAh) batteries, providing redundancy for safety in case one battery fails during operation. Charging these batteries takes approximately 92 minutes with the standard charger. The battery management system ensures that all batteries are monitored for performance, balancing power distribution between them.

The M600 can reach a maximum speed of 65 km/h in GPS mode and can ascend at a rate of 5 m/s while descending at 3 m/s. It operates at altitudes of up to 2500 meters above sea level with the TB47S batteries, and this range increases to 4500 meters with the TB48S batteries. The drone has a maximum operational range of up to 5 km, depending on conditions and local regulations, and is equipped with dual GNSS (GPS and GLONASS) for reliable navigation.

The flight control system is handled by either the A3 Pro or the DJI N3 flight controllers, providing enhanced flight precision. Communication with the remote control can operate at both 2.4 GHz and 5.8 GHz frequencies, with an operational range of up to 5 km. Additionally, the M600 supports RTK (Real-Time Kinematic) modules, which are optional but provide centimeter-level precision for navigation and mapping.

The drone supports various camera systems, including DJI's Zenmuse series (such as the X3, X5, X5R, XT, and Z30). It also accommodates third-party cameras with the appropriate support systems. For stabilization, it uses a 3-axis gimbal system, such as the Zenmuse gimbals, ensuring smooth footage and precise control over the camera.

The DJI GO and DJI Assistant 2 software offer full control of the M600's functions, including intelligent flight modes such as Waypoints, Point of Interest, ActiveTrack, TapFly, and Follow Me. The Return-to-Home feature is triggered automatically in case of signal loss or low battery, ensuring safe operation.

In terms of performance, the drone operates in a temperature range of -10°C to 40°C and includes an active cooling system for its electronic controllers, ensuring stable performance in different environmental conditions.

Finally, the M600's foldable design allows for easier transport, making it a convenient choice for professionals working in the field.

These features make the Matrice 600 a robust platform for professional applications that demand high precision, payload capacity, and flexibility in sensor and camera configurations.

A.1.5 Atyges FX8

The Atyges FX8 (see Figure A.5) is a professional-grade drone designed for industrial and technical applications such as aerial mapping, photogrammetry, surveillance, and inspection. Below is a detailed overview of its main technical features:

The Atyges FX8 has a compact folding design, making it highly portable and easy to transport to different locations. When fully deployed, it has a wheelbase of 1050 mm, and its total takeoff weight is approximately 6.6 kg without payload. It supports a maximum takeoff weight of 10 kg, allowing for diverse payload options, including high-resolution cameras and various sensors.

The drone's propulsion is powered by high-efficiency brushless motors paired with large carbon fiber propellers, ensuring stable and smooth flight even in challenging conditions. This combination provides both power and flight efficiency, critical for long-duration industrial tasks.

In terms of flight time, the FX8 can achieve up to 50 minutes of operation under ideal conditions without a payload. This duration decreases based on the weight and type of equipment it is carrying, but it remains an optimal choice for long-duration missions, such as surveying large areas or extended inspections.

The FX8 operates with redundant GNSS systems (GPS and GLONASS) for accurate positioning and stable flight, even in areas with poor satellite coverage. It also integrates RTK (Real-Time Kinematic) technology, which allows for centimeter-level accuracy in positioning, making it particularly useful for precision mapping and photogrammetry applications.

Figure A.5 Atyges FX8 acting as an FTM-based OSA

For communication, the FX8 drone uses 2.4 GHz and 5.8 GHz frequencies with a range of up to 8 km, depending on conditions and local regulations. The remote controller is highly intuitive, with a built-in display providing real-time flight data, live video feed, and system diagnostics. In terms of communication reliability, the FX8 drone has dual redundant communication links, ensuring continuous connection even in case of interference.

The payload capacity of the FX8 allows it to carry a wide range of equipment, including RGB cameras, multispectral sensors, LiDAR, thermal imaging cameras, and other custom sensors based on specific industrial needs. The drone is compatible with both fixed cameras and gimbals for 3-axis stabilization, ensuring that imagery or sensor data is collected with minimal vibration or distortion.

The drone's autopilot system is fully programmable, enabling the user to create and execute waypoint missions, flight paths, and mapping routes. The FX8 is capable of performing autonomous missions, including return-to-home and landing features triggered by low battery or communication loss.

The FX8 uses high-capacity LiPo batteries to achieve its extended flight times, and the battery management system monitors each battery's health, charge level, and performance during operation. Depending on the mission, the battery swapping system allows for quick replacement in the field, minimizing downtime between flights.

For software, the Atyges FX8 is compatible with standard industrial drone operation platforms, including Mission Planner and DJI Ground Station Pro, allowing for a range of mission types, from aerial mapping to autonomous inspection and surveillance.

In terms of environmental adaptability, the Atyges FX8 operates in a wide temperature range, suitable for various climates and field conditions. It includes a sturdy, weather-resistant frame, making it suitable for operations in tough environments such as industrial sites, wind farms, and agricultural fields.

The FX8 also features various safety systems, including obstacle detection and avoidance technology, redundant flight control systems, and emergency landing protocols. This ensures maximum operational safety during critical missions.

Overall, the Atyges FX8 stands out as a versatile and durable drone, capable of carrying advanced payloads for industrial-grade applications such as precision agriculture, topographic surveys, and infrastructure inspections. Its combination of long flight time, high payload capacity, and advanced safety systems makes it an ideal platform for professional and industrial tasks.

A.1.6 UR3 manipulator

The Universal Robots UR3 is a Compact Robotic Arm (CRA) designed for small-scale industrial applications that demand precision, repeatability, and safety. Below is an overview of its key features and technical specifications:

With a payload capacity of 3 kg, the UR3 is ideal for tasks such as light assembly, laboratory automation, and other precision-oriented operations. It offers a reach of up to 500 mm, making it well-suited for tasks within confined or limited workspaces. Despite its small size, the UR3 supports various tools and end-effectors, including grippers and sensors, enhancing its adaptability for multiple applications.

Figure A.6 illustrates the IoRT-in-hand system installed on the UR3's end-effector.

The UR3 collaborative robot (cobot) is known for its 6-axis design, which provides a high degree of freedom of movement. Each joint can rotate ±360 degrees, while the sixth axis can continuously rotate, enabling advanced flexibility and complex movements. With a repeatability of ±0.1 mm, the UR3 is highly accurate, making it well-suited for delicate tasks such as precision assembly, testing, and handling small components.

The UR3 features force sensing in all its joints, allowing it to respond to external forces effectively. This capability makes it ideal for collaborative environments, as it can stop or slow down when encountering resistance or potential collisions, ensuring the safety of human operators. Additionally, its integrated force/torque sensor enhances its performance in delicate operations, such as polishing or fine manipulation.

Weighing just 11 kg, the UR3 is portable and easy to redeploy for different tasks or locations within a facility. This portability is particularly advantageous in industries requiring frequent adjustments to production lines or laboratory automation, where flexibility and rapid redeployment are essential.

Programming the UR3 is straightforward, thanks to its intuitive URCap software and Polyscope graphical user interface, operated via a 12-inch touchscreen. The interface supports programming by demonstration, allowing users to manually guide the robot to teach it tasks, making it accessible even to those without extensive robotics experience. For more advanced users, the UR3 also supports URScript, a proprietary programming language from Universal Robots, providing enhanced flexibility for complex tasks.

The UR3 meets industry standards for collaborative robots, adhering to ISO/TS 15066 safety protocols. It includes built-in safety features such as emergency stops, force-limited joints, and protective stops, which ensure safe operation alongside human workers. Unlike traditional industrial robots, the UR3 can operate without physical barriers, making it ideal for environments where safety and collaboration are paramount.

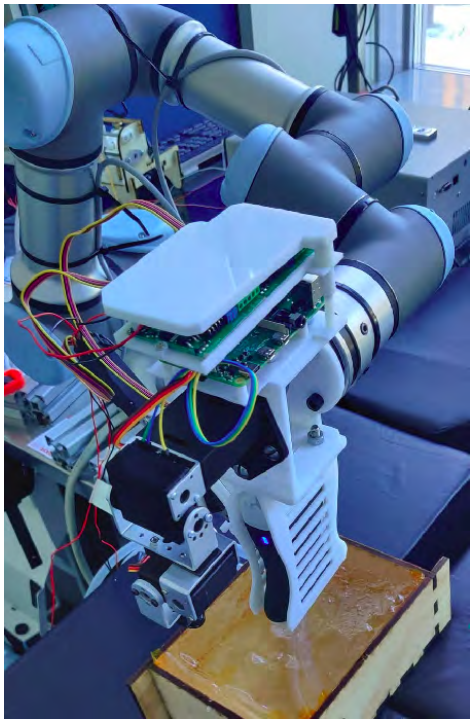


Figure A.6 Prototype of the IoRT-in-hand for the UR3 manipulator.

Energy efficiency is another highlight of the UR3, as it consumes only 350 watts of power under typical operating conditions. This low energy consumption reduces operational costs, making it an economical solution for businesses seeking to automate their processes.

The UR3 is designed for various applications, including assembly, pick and place, screw-driving, quality inspection, testing and laboratory automation, and polishing and deburring. Its compact size and adaptability make it particularly suitable for small workstations and confined spaces where traditional robots may not fit.

Finally, the UR3 can be easily integrated into existing production setups without requiring significant infrastructure changes. This seamless integration allows businesses to automate specific tasks without disrupting their entire production line. The Universal Robots UR3 is a versatile, user-friendly cobot that excels in tasks requiring precision, flexibility, and safety. It is an excellent choice for manufacturing, electronics, healthcare, and research industries, particularly in environments that emphasize collaborative workspaces and human-robot interaction.

Table A.1 Summary of equipment and capabilities of the robotic agents used in this work.

Feature	**Cuadriga**	**Rambler**	**DJI M600**	**Atyges FX8**	**UR3**	**Rover J8**
Platform Type	4-wheel UGV, skid-steer	4-wheel UGV, active suspension	Hexacopter drone	Industrial quad-copter	6-DOF cobot arm	8-wheel UGV, skid-steer, amphibious
5G UE	Huawei 5G CPE Pro (H122-373), static IP, SA/NSA, SIM 100 GB/mo	Huawei 5G CPE Pro (H122-373), static IP, SA/NSA, SIM 100 GB/mo	Google Pixel 7 Pro (5G + GNSS, Android 11–15, SIM 100 GB/mo)	Google Pixel 7 Pro (5G + GNSS, Android 11–15, SIM 100 GB/mo)	No	Teltonika RUTX50 (5G, static IP, SIM 100 GB/mo) + MIMO antenna + GS110MX switch
BLE Scanner	ESP32 + Meshlium	ESP32 + Meshlium	ESP32	ESP32	Not applicable	Not applicable
LoRa SG	Yes (sensor nodes onboard)	Yes (sensor nodes onboard)	Yes (sensor nodes onboard)	Optional	No	Yes (sensor nodes onboard)
LoRa CN	No	Yes (Multitech MTCDTIP 220L + local Mosquitto broker)	Yes (IMST iC880A LoRa concentrator onboard + Mosquitto)	No	No	No
PC	Intel NUC (i7 8th Gen, 16 GB, 256 GB SSD)	2 PCs (Win+Ubuntu)	RPi 4 + Smartphone	Intel NUC (i7 8th Gen, 16 GB, 256 GB SSD)	URCap/Polyscope	1 PC (Ubuntu 22.04)
ROS 1	Kinetic, Melodic	Kinetic, Melodic	Kinetic, Melodic	Melodic, Noetic	Melodic, Noetic	Melodic, Noetic
ROS 2	Not used	Not used	Foxy, Humble (onboard PC)	Foxy, Humble (onboard PC)	Not used	Humble
LabVIEW	Yes	Yes	No	No	Not applicable	Not applicable
Controller	Embedded MCU	Embedded + Joystick	Pixhawk Cube Orange (previously DJI A3)	Pixhawk Cube Orange	URScript/Polyscope	Autonomous + RC
Weight (kg)	83.1 kg	460 kg	9.5 kg	6.6 kg	11 kg	1090 kg
Dimensions	0.82 m×0.64 m×0.81 m	1.6 m×1.2 m×0.66 m	Ø1.13 m×0.53 m	Folded: 1.05 m	Reach: 0.5 m	—
Wheels/Props	35.5 cm tires	50 cm tires	17" props	4 dual-rotors (8 blades total)	—	Rubber (8)
Payload (kg)	120 kg	>320 kg	6.0 kg	3.4 kg	3 kg	600 kg
Autonomy	∼2 h	∼3 h	16 min to 35 min	up to 50 min	Continuous	Indeterminate (> 6 h)
Speed	1.2 m s^{-1}	22 m s^{-1}	18 m s^{-1}	—	±360°, cont. axis 6	—
Power	36 V lead-acid	64× $LiFePO_4$	6× TB47S LiPo 6S (4500 mA h, 22.2 V)	High-capacity LiPo	350 W typical	46 V battery
Sensors	u-blox ZED-F9P GNSS, IMU, PTZ, LiDAR, BLE, LoRa, AXIS camera	u-blox ZED-F9P GNSS, IMU, PTZ, LiDAR, BLE, LoRa, AXIS camera	ZED-F9P GNSS (USB-C powered via smartphone), IMU, BLE, LiDAR, camera	ZED-F9P GNSS (USB-C powered via smartphone), RGB, thermal, LiDAR	End-effector with medical tool, endoscopic RGB camera, F/T sensor, Raspberry Pi 4	ZED-F9P GNSS, LiDAR (Velodyne VLP-16), BLE, AXIS camera
Comms	ZigBee, 802.11, BLE, LoRa, LTE, 5G, Mosquitto MQTT	ZigBee, 802.11, BLE, LoRa, LTE, 5G, Mosquitto MQTT	ZigBee, 802.11, BLE, LoRa, LTE, 5G, Mosquitto MQTT	ZigBee, 802.11, BLE, LoRa, LTE, 5G, Mosquitto MQTT	LTE, 802.11	802.11, BLE, LoRa, LTE, 5G, Mosquitto MQTT
Switch	—	Netgear PoE	—	—	—	Netgear GS110MX
WiFi FTM Anchor	GJ2CQ model (14 V, 1.1 A, barrel, powered by platform)	GJ2CQ model (14 V, 1.1 A, barrel, powered by platform)	GL0102 model (USB-C, 5 V, 3 A, 6.7 h autonomy, 500 g)	GL0102 model (USB-C, 5 V, 3 A, 6.7 h autonomy, 500 g)	Not applicable	GJ2CQ model (14 V, 1.1 A, barrel, powered by platform)

A.2 Entities

Several entities, including companies, military units, and emergency services, have contributed to the exercises described in this thesis. In some instances, their participation was indirect, facilitated through the general organization of the annual International Conference on Security, Emergencies, and Catastrophes (JEMERG), coordinated by the chair, Dr. Jesús Miranda. In some other cases, there was direct collaboration in the experiments, and these entities have been specifically marked in the list.

- **Ejército de Tierra:**
  - X Bandera del 4º Tercio Alejandro Farnesio de la Brigada de la Legión (BRILEG)*: Elite infantry unit with experience in rescue operations.
- **Unidad Militar de Emergencias (UME):**
  - 1º Batallón de Intervención en Emergencias (BIEM I)*: Specialized in rapid interventions during disasters.
  - 2º Batallón de Intervención en Emergencias (BIEM II)*: Complementary unit for emergency operations.
- **Ejército del Aire y del Espacio:**
  - Ala 78*: Training and operational support unit.
  - Ala 49*: Unit specialized in surveillance and aerial operations.
  - Unidad Médica de Aeroevacuación (UMAER): Medical team specializing in aerial evacuations.
- **Diputación de Málaga:**
  - Consorcio Provincial de Bomberos de Málaga*: Firefighting service for provincial emergencies.
  - Protección Civil: Volunteer organization providing disaster assistance.
- **Ayuntamiento de Málaga:**
  - Real Cuerpo de Bomberos de Málaga: Municipal fire department.
  - Grupo Operativo de Apoyo de la Policía Local (GOAS)*: Rapid response unit for local emergencies.

- Alisys Robotics*: They collaborated with its robotics quadruped (from Boston Dynamics), which was used in this thesis to mount LoRa EDs for environmental measuring (see Figure A.7).
- SDLE (currently SDLE NexGen)*: Technology company specializing in drones and advanced systems.
- Vodafone España*: Telecommunications operator providing emergency solutions.
- Vodafone Innovation Hub España*: Innovation center for communication technologies.
- Huawei*: Multinational technology company with expertise in network infrastructure.
- Fundación ONCE*: Organization dedicated to inclusion and social assistance.
- Cruz Roja Española (Comité Provincial de Málaga): Humanitarian organization providing emergency support.
- United Nations Institute for Training and Research (UNITAR): CIFAL Málaga: Training center for emergencies and sustainability.
- The Maersk Mc-Kinney Moller Institute, South Denmark University, Odense (Denmark)*: Advanced research center in robotics.
- Instituto de Investigación en Ingeniería Mecatrónica y Sistemas Ciberfísicos de la Universidad de Málaga (IMECH-UMA)*: Research in robotics applied to emergencies.

Figure A.7 Spot (Boston Dynamics) being tele-operated while carrying a LoRa sensory group

- Instituto de Inve stigación en Ingeniería de Telecomunicaciones de la Universidad de Málaga (TELMA)*: Development of advanced communication technologies.
- Centro de Automática y Robótica (CAR), Centro Mixto UPM-CSIC: Research in robotics and autonomous systems.
- Deveryware: Company specialized in security and localization solutions.
- **SAMUR 112 Madrid:**
  - SAMUR Protección Civil: Emergency medical assistance service.
- **SUMMA 112 Madrid:**
  - SUMMA 112: Emergency and urgent medical service of the Madrid Region.
- **Servicio Andaluz de Salud (SAS):**
  - 061 Emergencias Sanitarias: Medical emergency service in Andalusia.
  - SUAP (Servicio de Urgencias de Atención Primaria): Local urgent medical care service.
- Cuerpo Nacional de Policía: National security and emergency management force.
- Guardia Civil: Security and rescue unit for rural and complex environments.
- 112 Andalucía: Emergency coordination service in Andalusia.
- Metropolitan College of New York: Educational institution offering training in emergency management.
- Instituto Galileo Galilei (Especialidad en Protección Civil y Emergencias)*: Technical training in disaster management.
- Colegio Oficial de Psicólogos de Andalucía Oriental: Psychological support in emergencies.
- SEPADEM (Sociedad Española de Psicología Aplicada a Desastres, Urgencias y Emergencias): Organization specializing in psychological support during disasters.
- ESAD Escuela Superior de Arte Dramático Málaga*: Participation in emergency simulation exercises.
- Bomberos Sin Fronteras ONGD*: Humanitarian organization specializing in international rescue missions.