

Peer-Reviewed Technical Communication

System Architecture and Communication Infrastructure for the RoboVaaS Project

Emanuele Coccolo, Cosmin Delea, Fabian Steinmetz ^{IB}, Roberto Francescon, *Student Member, IEEE*,
Alberto Signori ^{IB}, *Student Member, IEEE*, Ching Nok Au, Filippo Campagnaro ^{IB}, *Member, IEEE*,
Vincent Schneider ^{IB}, Federico Favaro, Johannes Oeffner ^{IB}, Bernd-Christian Renner ^{IB},
and Michele Zorzi ^{IB}, *Fellow, IEEE*

Abstract—Current advancements in waterborne autonomous systems, together with the development of cloud-based service-oriented architectures and the recent availability of low-cost underwater acoustic modems and long-range above-water wireless devices, enable the development of new applications to support ships and port activities. Autonomous surface vehicles can, for instance, be used to perform bathymetry and environmental data collection tasks to ensure under keel clearance and to monitor the quality of the water. Similarly, remotely operated vehicles can be deployed to inspect ship hulls and typical port infrastructure elements, such as quay and sheet piling walls. In this article, we present the complete system deployed for the small-scale demonstrations of the Robotic Vessels as-a-Service project, which introduces an on-demand service-based cloud system that dispatches unmanned vehicles capable of performing the required service either autonomously or piloted. These vessels are able to interact with sensors deployed in the port and with the shore station through an integrated underwater and above-water network. The developed system has been validated through sea trials and showcased through an underwater sensor data collection use case. The results of the test presented in this article provide a proof of concept of the system design and indicate its technical feasibility. It also shows the need for further developments for a mature technology allowing on-demand robotic maritime assistance services in real operational scenarios.

Index Terms—Long-range Wi-Fi, network performance validation, robots as a service, sea experiments, underwater acoustic networks, unmanned vessels.

Manuscript received 15 July 2021; revised 28 April 2022 and 30 September 2022; accepted 13 December 2022. Date of publication 7 March 2023; date of current version 14 July 2023. This work was supported in part by the Italian Ministry of Education, Universities and Research, in part by the German Federal Ministry of Economy, in part by ERA-NET Cofund MarTERA under Contract 728053, and in part by the European Union - FSE REACT EU, PON Research and Innovation 2014-2020 (DM 1062/2021). (*Corresponding author: Filippo Campagnaro.*)

Associate Editor: R. Bachmayer.

Emanuele Coccolo was with the Department of Information Engineering, University of Padova, 35131 Padova, Italy. He is now with EvoLogics GmbH, 13355 Berlin, Germany (e-mail: emanuele.coccolo@evolitics.de).

Alberto Signori, Filippo Campagnaro, Federico Favaro, and Michele Zorzi are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: signoria@dei.unipd.it; campagn1@dei.unipd.it; fedefava86@gmail.com; zorzi@dei.unipd.it).

Cosmin Delea, Ching Nok Au, Vincent Schneider, and Johannes Oeffner are with the Fraunhofer Center for Maritime Logistics and Service, 21073 Hamburg, Germany (e-mail: cdelea@cml.fraunhofer.de; ching.nok.au@cml.fraunhofer.de; vincent.schneider@cml.fraunhofer.de; johannes.oeffner@cml.fraunhofer.de).

Fabian Steinmetz and Bernd-Christian Renner are with the Institute for Autonomous Cyber-Physical Systems, Hamburg University of Technology, 21079 Hamburg, Germany (e-mail: fabian.steinmetz@tuhh.de; christian.renner@tuhh.de).

Roberto Francescon was with the Wireless and More srl, 35129 Padova, Italy. He is now with the University of Padova, 35131 Padova, Italy (e-mail: frances1@dei.unipd.it).

Digital Object Identifier 10.1109/JOE.2023.3234710

I. INTRODUCTION AND MOTIVATION FOR THIS ARTICLE

Applied research in the field of robotics is nowadays trying to find more compelling and user-friendly robotic solutions for enabling various industrial applications in an intelligent manner. The main objectives are increasing efficiency and reducing human hazard. In this context, the Robotic Vessels as-a-Service (RoboVaaS) project [1] aims to revolutionize the shipping and near-shore operations by offering robotic-aided services via interconnected unmanned vehicles (UVs), equipped with specialized sensor technology, a reliable data transfer cloud network for above-water and underwater communication, a monitoring station, and a real-time web-based user interface. The high level of autonomy implied in RoboVaaS is expected to be reached by using unmanned vessels, such as autonomous surface vehicles (ASVs) and remotely operated vehicles (ROVs), to perform both autonomous missions and tasks with a minimal set of commands sent by human operators. For the current work, a path follower controller for both ASVs and ROVs, which follows a 2-D vector to a given waypoint, independent of time, is used throughout the in-lab tests and sea trials. The services envisioned in the RoboVaaS project are a ship hull and quay wall inspection service, an antigrounding service and an environmental and bathymetry data collection service (see Fig. 1).

An envisioned user of the RoboVaaS system (such as a ship owner, a navigator, or a port authority) can use one or more of the aforementioned services by placing specific requests and following their progress through a standard web browser. Once the service request is issued, the service provider will assign the request in the form of a task or mission to a single or several surface and underwater UVs, commanded and supervised from a ground control station. During a mission, its progress and the acquired data are transmitted to the users in real time or—in some cases—after postprocessing. Once the mission is completed, the results of the service can be recalled by the user at any point in time, to visualize them through a web browser or to generate a report, based on the acquired data.

The popularity of drones, be they airborne, land, or waterborne, has exponentially increased in the past few years [2]. This is due to the trend of the scientific community and the industry to find consensus for the tools required for operating and developing such robotic systems. The main difficulties consist in the diversity and complexity of the mobile robots, either from the nonlinearities of their mathematical models or from the loosely confined operational envelope. Furthermore, there are great challenges for fusing real-time data and using it as input for complex navigation algorithms or even for teleoperation. The major concern was to minimize the overhead of the exchanged information, which can be done through both hardware and software. For the first option, one can use common field buses that enable high-frequency

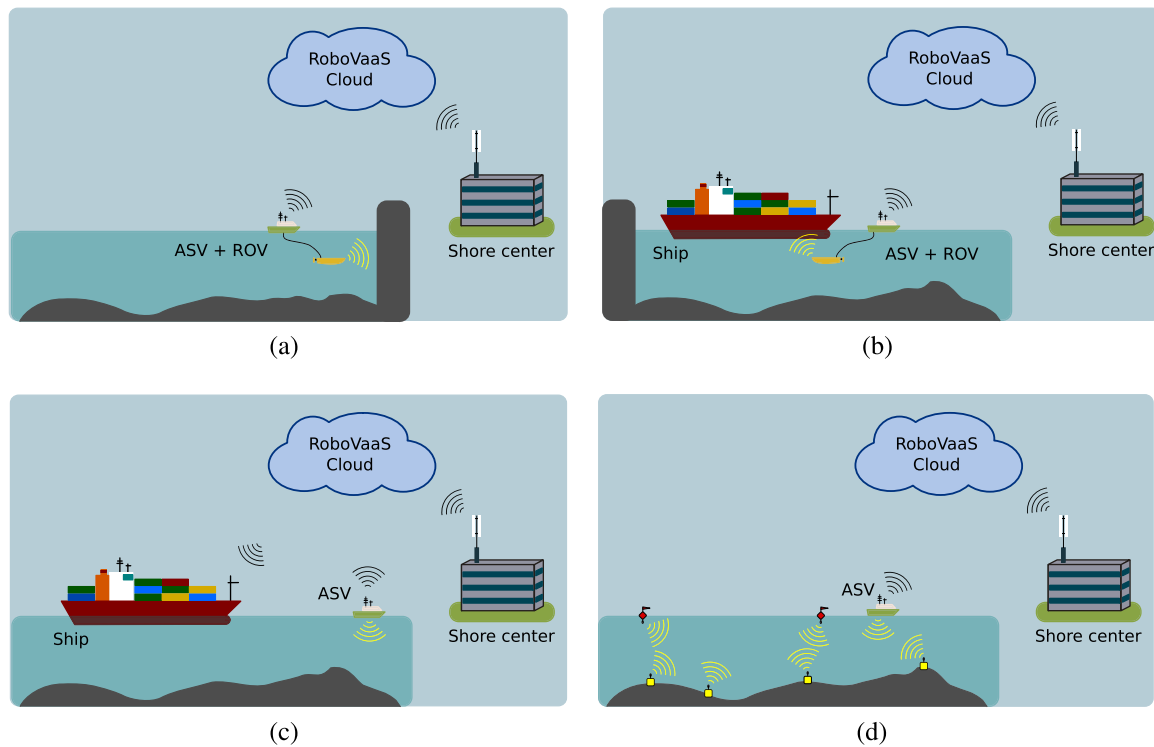


Fig. 1. RoboVaaS services. (a) Quay wall inspection. (b) Ship hull inspection. (c) Antigrounding. (d) Environmental and bathymetry data collection.

control operations, i.e., operations that require to transmit more than 200 control messages per second with reasonable latency [3]. The software, instead, can be optimized by choosing more suitable communication protocols for interfacing the robot's subsystems or by using programming languages that are closer to the hardware and give enough flexibility for modeling the architecture. For example, a typical messaging protocol used by drones is micro aerial vehicle link, which serializes the incoming and outgoing messages into a platform-independent binary format, thus making the communication lightweight and real time [4]. Also, popular robotics frameworks such as robot operating system (ROS) make use of user datagram protocol (UDP)-based protocols, such as UDPROS, to decrease latency between peers communicating with the ROS Master [5]. The drawback is the reliability issues generally related to UDP-based protocols [6], especially for datagram streams with high bandwidth needs. This is particularly important when dealing with high-frequency robots, such as industrial manipulators, that have very strict security requirements and allow a limited number of software interfaces that can be used for interaction with humans. Marine robotics, especially ASVs, can operate at lower data frequencies as they are slower systems. This adds another degree of flexibility that can be used to cope with the complex system logic expected for commanding such systems for diverse industrial applications.

The unmanned system presented in this article—the SeaML ASV—has an operation frequency, defined as the number of JSON-formatted control messages sent every second, of 1–10 Hz, which can be transferred over transmission control protocol (TCP)-based protocols, which ensure reliability and lossless data transfer. From the measurements acquired during tests, the size of each JSON control message is less than 1.2 kB. Usually, such implementations are favored in client–server applications, where the server-side applications are responsible for ensuring a complete and secure data transfer from and to the clients. The information distribution involves packet duplication and routing, which transfers the burden from the unmanned systems to the shore servers, which are easier to scale up. The aim is to maintain the high degree

of flexibility required for operating mobile robots but without compromising the security needed for industrial applications. For maritime systems that are aimed to offer industrial-level services, this aspect is crucial, because there are multiple actors (e.g., port authorities, clients, technical operators, etc.) that need to be involved and require a safe and secure infrastructure for on-demand services performed with unmanned systems.

The assumption in this case is that the low operation frequencies and implicit volume of data considered when designing the system architecture do not affect the safety of operation and monitoring of the unmanned system. These assumptions help us build a service-oriented architecture (SOA) that favors the coupling of the robotic system to an industrial ecosystem, aimed at providing on-demand services to the users. This article proposes a marine robotic solution, whose software design is based on SOA that enables real-time client–server communication to securely fulfill various on-demand services.

The underwater and above-water marine assets used in this project need to exchange information to support the RoboVaaS services. Acoustic communication systems are used for underwater data transfer, as acoustic transmission and signal processing techniques have reached a remarkable level of maturity [7], [8], [9]. New acoustic low-power sensor nodes have been developed [10], and the data acquired by these sensors have been collected from a newly developed ASV equipped with both an acoustic modem and an above-water wireless link: the ASV exploits the latter to upload the collected data to shore.

The contribution of this article is threefold. First, it presents the overall RoboVaaS concept, introducing all robotic-aided services supported by the system. Even though the services provided in RoboVaaS are currently offered by a couple of maritime service providers, there are very few examples of companies providing such a wide spectrum and the technologies chosen for developing them are not publicly accessible. Second, this article describes not only the cloud platform and the user interface employed to require, perform, and monitor the services, but also the communication infrastructure and the data flows and interfaces

used to convey the data between the UVs and the cloud system. As modifications during the deployment phase of a robotic systems can be more expensive than those during the design phase, this article offers a validated solution that can be further replicated or used as a benchmark. Finally, this article provides the complete details of the deployment of a robotic system performing one of the four envisioned use cases, evaluating the whole communication infrastructure, the functionalities of the principal vehicle, and the cloud platform. This article significantly differs from our previous work [11], [12], [13], [14] because previous publications focused on individual aspects of the whole project and evaluated a single component in a simulated or mocked environment, while in this article, all components have been integrated together and tested in the field with an actual deployment.

Two main novel aspects are introduced by this project. The former is the introduction of the “as-a-service” concept to port activities, hence performing on-demand those tasks that are usually executed by port authorities periodically, i.e., upon request by a registered user that can require the services simply using a web interface. The latter main novel aspect is the fact that all services are performed with unmanned surface and underwater vehicles, hence without the need for manned vessels, specialized crews, and/or divers to perform that task, thereby significantly reducing risks for human operators, which increase exponentially with depth. Realizing each service required the implementation of several innovative components, from a surface vessel able to transport an ROV to the integration of underwater and above-water network components and the realization of a very low-cost acoustic modem to retrieve data from a dense underwater sensor network. This article describes in detail the deployment of a first integrated affordable system that can enable monitoring of coasts, rivers, and lakes, at a cost that is an order of magnitude lower than traditional offshore deployments. All components are operational and work in real time, with no need for any further postprocessing, proving the high technology readiness level (TRL) of our system.

The rest of this article is organized as follows. Section II describes the RoboVaaS use cases and their data streaming requirements, and Section III presents the related works of similar projects and solutions to enable the communication of maritime vehicles. Section IV presents the design of each part of the system, from the architecture of the RoboVaaS cloud control system for UVs to the underwater and above-water communication infrastructure and the data format used to support the robotics operations. Section V shows the preliminary evaluation of the system in a simulated environment, while Sections VI and VII present the field test deployment settings and its results, respectively. Finally, Section VIII concludes this article.

II. ROBOVAAS USE CASE DESCRIPTION AND REQUIREMENTS

Within the high-level RoboVaaS vision, a number of services that have a positive impact on near-shore maritime operations have been identified [13], including a quay wall inspection service [see Fig. 1(a)], a ship hull inspection service [see Fig. 1(b)], an antigrounding service [see Fig. 1(c)], and an environmental and bathymetry data collection service [see Fig. 1(d)]. These four services have been identified and defined with the help of the Hamburg Port Authority (HPA), partner of RoboVaaS and crucial stakeholder within the project as a potential end user. In several workshops with an HPA liaison and HPA departments, the consortium was provided the requirements for each service [15]. For the bathymetry data collection service, a workshop with the hydrography department of HPA at an early development stage yielded the necessity of: 1) a cleaner webUI layout; 2) a strict separation of user (job creation) and operator (vehicle control) to avoid uncontrolled vehicle commands; 3) the suggestion of a comment section where users can add job-specific information; and 4) no custom integration of complex equipment such

as multibeam echosounder but the use of proprietary software solutions that can be performed onboard on separate computing unit and can then relay processed data to the RoboVaaS network. Similar feedback loops were performed for the antigrounding service with nautical officers in state-of-the-art Ship Handling Simulators [16]. All these services are performed on-demand upon a request performed by a user registered in the RoboVaaS cloud by means of UVs, such as ASVs and ROVs. The communication link between the deployed ASV and the ground control station can be supported either with a dedicated Wi-Fi or WiMAX deployment or exploiting the cellular coverage provided by a long term evolution (LTE) base station located in the proximity of the area [11], if any.

The information flows needed to support all RoboVaaS services are as follows:

- 1) a request/response flow between the user, the RoboVaaS cloud, and the shore center, which carries the messages for the authentication of the users, the service description, and information about the service availability;
- 2) a mission planning flow between the RoboVaaS cloud and the ASV, in which the information about the working area and the details of the mission are sent to the ASV before the mission starts;
- 3) the control and monitoring flow between the unmanned vessels and the control station, which enables the remote control of the system and the execution of the required task;
- 4) the after-action report flow, used to upload the outcome of the mission and the data collected during the mission to the RoboVaaS cloud.

The control and monitoring flow has different requirements between services, depending on the service target and layout. In the following, we present the details of each service.

A. Quay Wall Inspection

For the quay wall inspection service [see Fig. 1(a)], an ASV-carried ROV assesses the status of the quay walls of an area defined by the user. After the ASV travels to the required area, the ROV is launched and operated along the quay wall while being supported by the ASV, which follows its movements. Both the ASV and the ROV are equipped with high-resolution cameras capturing the video for the operator view.

According to both the simulation study and the analysis performed in [14], and to the data stream measurements performed during the project, to support the quay wall inspection service, the following transmission streams are needed:

- 1) two UDP video streams of 1 Mbit/s each to monitor ASV [17] and ROV [18] operations with high reliability;
- 2) manual command sent over a UDP control stream, with rate 5 kbit/s, to control the movements of a small inspection class ROV [18];
- 3) a manual command sent over a TCP control stream to operate the ASV, with rate 50 kbit/s.

B. Ship Hull Inspection

An ASV-carried ROV performs the inspection of a ship hull to detect the presence of defects that may compromise passenger safety [see Fig. 1(b)]. As for the quay wall inspection service, both the ASV and the ROV are equipped with high-resolution cameras. In addition, the ROV features the Kraken SeaVision system [19], a laser-based sensor for the 3-D imaging of the hull, and multiple sonar cameras.

According to the simulation study and the analysis performed in [14], and according to the measurements performed during the project, to

support the ship hull inspection service, the following transmission streams are needed:

- 1) two UDP video streams of 1 Mbit/s each to monitor ASV and ROV operations with high reliability;
- 2) a TCP control stream to operate the ASV, with rate 50 kbit/s;
- 3) a UDP control stream, with rate 100 kbit/s, to control the movements of a big and stable inspection class ROV able to operate the 3-D Laser scanner with high precision [20];
- 4) a stream for the operational control of Kraken SeaVision 3-D mapping system with rate 3 Mbit/s;
- 5) a stream to upload the 3-D images captured by the Kraken SeaVision 3-D mapping system. This application produces a large amount of data to be transmitted (that varies depending on the desired resolution): these data can be conveyed to the control station at the end of the mission to avoid overloading the network.

C. Antigrounding

The on-demand antigrounding service envisioned in RoboVaaS is meant to deliver real-time bathymetry data to a ship sailing through narrow riverbeds or shallow waters, allowing it to react to threats not marked in conventional navigational charts or to increase sailing intervals in tide-dependent waterways. An ASV traveling ahead of a merchant vessel measures the water depth and, thus, enables real time under keel clearance data [see Fig. 1(c)] as far ahead as the stopping distance, outperforming commercially available forward-looking sonars in shallow waters. The antigrounding service is, thus, envisioned to guarantee safe navigation in areas where bathymetry data are either unavailable (e.g., in some South American port areas) or outdated (e.g., in the case of the port of Hamburg because of frequent changes in the riverbed due to tidal bores and ship passages that require daily measurements for critical regions [21]).

According to the simulation study and the analysis performed in [11], in the antigrounding service, the ASV needs to transmit to the ship a multibeam sonar UDP stream of approximately 0.65 Mbit/s and can be supported by means of a broadband radio link (Wi-Fi, WiMax, or LTE) plus an additional long range low rate (LoRa) backup link used only for alarm messages. The ASV can perform a predefined path with no need for additional data streams. Control and video streams to monitor the mission may be envisioned, but are not strictly required. Within RoboVaaS, not only have the technical data link requirements been evaluated, but a nautical simulation using ship handling simulators has also been implemented and nautical officers have executed an exercise to validate the functionality of the prototype [16].

D. Bathymetry, Environmental Data Collection, and Data Muling

The environmental and bathymetry data collection service is performed directly by an ASV moving along a preloaded path [see Fig. 1(d)]. Both collected data can be either stored in the ASV and uploaded to the ground control station at the end of the mission or transmitted to a server in real time, depending on the network coverage. While the bathymetry data are collected directly by the ASV through sonar systems, the environmental data can be either collected by the ASV sensors or retrieved by the ASV from submerged sensor nodes collecting measurements for a long period of time. If underwater sensors are deployed in the area where the ASV is performing any of the aforementioned services, it can collect their data as a side task. The data are collected from the underwater nodes via acoustic modems, by using a polling protocol. From here onward, in this article, we refer to

the underwater data collection service with the name of data muling, as the ASV, acting as a mule, collects the data from the underwater nodes and then conveys it to the shore server.

The data flow of the data-muling task is related to the sensor data that need to be compressed, sent to the ASV through the underwater network, and finally conveyed to the RoboVaaS server. This data flow does not have stringent requirements in terms of delay or data rate, as each sensor acquires and stores only a few measurements per day (e.g., one sensor measurement every 30 min, with a resulting data generation of few kilobits per day), and the data retrieval operation can last until all data are collected from the sensors.

Being the latter one of the most challenging scenarios for the communication infrastructure, due to the instability of the underwater acoustic channel and the lack of standard devices for underwater communications, in this article, after analyzing the RoboVaaS system architecture, we present the evaluation of the integrated network in a lake trial where the ASV performs the data-muling task. Nevertheless, also, the feasibility of the other services is assessed with the evaluation of the above-water network coverage, throughput, delay, and packet delivery ratio (PDR). The evaluations of the ROV-based use cases have been performed in a separate test and demonstrated with a live demo during the ITS World Congress, available online [22], and is out of the scope of this article.

III. RELATED WORK

During the last decade, several solutions aimed to bring waterborne UVs closer to the robustness and security needs of industrial applications have emerged. Both the vehicles themselves and the infrastructure required for secure operation have gone under thorough analysis. For example, in the context of the SUNRISE project [23], led by the University of Rome La Sapienza, several testbeds have been built and adapted for static and mobile underwater network testing and experimentation. The SUNRISE GATE provides a unified web interface to access every testbed being part of the SUNRISE federation. The LOON testbed [24], implemented by the NATO STO CMRE, for instance, is envisioned to foster a cooperative development of underwater communications and networking with both mobile and static nodes. This testbed is widely used by the scientific community [25], [26], [27] to perform tests on underwater acoustic communication and localization. The University of Porto, instead, built the UPORTO testbed [28] for collaborative experimentation and synergistic operations with harbor systems, including mapping and ship traffic monitoring and control. This paved the way to the advancement of new projects targeting specific applications; for instance, the works of the CRAS laboratory at INESC TEC, which cover areas such as autonomous navigation [29], long-term deployments, data gathering [30], mapping [31], and surveillance.

Other laboratories started projects with analogous aims, like the MORUS project, led by the University of Zagreb, which studied the creation of a complex system with aerial and underwater autonomous vehicles [32] for security and environmental monitoring. Similarly, in the SWARMS project [33], several companies and research institutes combined their efforts to make surface and underwater UVs cooperate with each other to facilitate offshore operations. Another similar project is OceanRINGS [34], where the University of Limerick developed an interface for offshore commercial operations for ROVs, providing both a real-world and a virtual environment.

Using as-a-service concepts for ports and coastal areas is a new trend offering a certain service on demand without the need to purchase the whole product entities involved. A first draft of the services that then became the core of the RoboVaaS project had first been presented in [35] by the HPA and Fraunhofer CML, to obtain a consistent vision

of interconnected smart ports. This concept was extended with the four scenarios described in Section II and three main development pillars—a small-scale demonstration platform to demonstrate scaled-down versions of the services to the projects stakeholders, the web-based service application design (incorporating the above-water and underwater communication), and feedback-based development with stakeholders of potential customers involved [15]. This article validates the core architecture for performing the aforementioned services that was comprehensively described in [13] and seeks to evaluate the scalability of the solution on a global level.

The concept of robotic devices granting services through a centralized system controlled through a web interface was first introduced in [36] for a use case in computer science education. In the context of the Ocean Technology Campus Rostock project, the new offshore infrastructure of the Digital Ocean Lab will set up an undersea site for testing ideas and simulations under controlled conditions in a real-world environment, where an efficient connection of different robotic entities will be demonstrated [37]. The Robot Web Tools project [38] created an open-source framework for communicating with ROS compliant robotic systems over web sockets, using *rosbridge* servers.

A web-based virtual machine for clients interacting with various ROS tools has been developed in [39] and evaluated during a robotics competition. The successful implementation of this robots-as-a-service system has led to the development of ROS Development Studio [5] that offers users on-demand ROS tools, such as the Gazebo Simulator, performing the computational workload on the server-side application. Like in RoboVaaS, a web interface allows the end user to access the whole array of services offered by the service provider. The concept of service robots can also be applied to different domains, such as health care. In [40], for instance, the concept of cloud robotics is extended to service robots to assist the elderly and the disabled in their daily activities. The robots are permanently connected to the cloud; thus, their operations are monitored by developers that can perform maintenance tasks and provide assistance, when required. The main differences between the past projects and RoboVaaS are the use cases that have been developed with port-specific operations in mind and a focus on waterborne robotics.

In the context of data collection from sensor nodes, and more in general in the acoustic underwater environment, the design of the medium access control (MAC) protocol plays an important role, since packet collisions and subsequent retransmissions may have a strong impact on the network. Indeed, due to the typical low bit rate of the acoustic modems and the high propagation delay, each retransmission significantly increases the channel occupancy.

In such an environment, Favaro et al. [41] compare two MAC random access protocols against a polling-based MAC protocol in the data-muling scenario. Specifically, the first random access protocol is a carrier-sense multiple access (CSMA)-based protocol called CSMA-Aloha-Trigger, the second, called distance-aware collision avoidance protocol (DACAP), is based on request-to-send (RTS) and clear-to-send (CTS) signaling. In the CSMA-based protocol, the nodes transmit their packets to the autonomous underwater vehicle (AUV) in a CSMA-like fashion only after the vehicle notifies its presence with the transmission of a trigger packet. In DACAP, sensor nodes are allowed to transmit their data only after the transmission of an RTS followed by the correct reception of a CTS packet. The polling-based protocol, instead, is a former version of the protocol used in this article and presented in [12]. The authors show that the polling-based MAC protocol in the considered scenario always outperforms the analyzed random access protocols in terms of throughput and PDR and is the protocol used in RoboVaaS for the data-muling use case.

Liu et al. [42] describe APOLL, a polling-based MAC protocol that groups all the nodes of the network in two classes: the APOLL controller (AC) and the mobile unit (MU). In APOLL, the AC can decide to schedule a REGISTRATION period in which the MU randomly selects a transmission opportunity to transmit its REGISTRATION packet to the AC. The number of opportunities and the length of each opportunity in the REGISTRATION period are decided by the AC. After the registration, the POLL-REPORT phase starts. In this phase, the AC sends a poll to the intended MU followed by data for the MU, if any, and the MU sends a report back to the AC. The POLL-REPORT phase goes on until the AC decides to schedule another REGISTRATION phase.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The RoboVaaS system enables users, e.g., port authorities, ship owners, or other port clients, to book different services, monitor their progress, and analyze the results through a web-based user interface. This means that as long as the client has access to the Internet, the user can access these services by a standard web browser. To enforce the business logic, users who have broker access rights can assign the tasks to operators, which are technicians responsible for deploying the UVs. At its core, the RoboVaaS system business logic contains a non-Structured Query Language (SQL) database, with different models developed for each of the four use cases. Whenever a new mission of any of the developed use cases is created, it receives a unique identifier, onto which mission critical data are stored. Missions can be created by any type of user, but the real-time connections that allow the flow of data into the database is controlled by users with higher ranking rights, such as operators. The logic enforcement is handled by the controllers of the web applications back end, which, in turn, are the single database clients that modify the content of the database, through specific queries. The queries are modeled based on the data inserted by the registered users in the templates handled by the front-end services directly on the web user interface. The content of the template is fed into the controllers of the back end through specific application programming interface (API) calls. For each call, the access rights of the user requesting the data are compared to the access right table in the database. The latter can be modified in the same manner, but solely by the users with administrative rights. While the structure itself is not entirely a novelty, the key elements that differentiate the RoboVaaS business logic have been explained in [13] and mainly refer to its applicability to waterborne robotics deployed in industrial sites, where little to no drone-specific infrastructure is provided. The presented solution can be first of all deployed in a local server set onshore, next to the place where the port authorities are performing their daily duties. Moreover, through the usage of broadband network connectivity, the solution can be upscaled with dedicated data centers, where a data warehouse system can gather and analyze data from various RoboVaaS units deployed on site. Currently, the RoboVaaS service architecture is not integrated into the operational structure of any port, as it is under development. However, it can be imagined that, once the system is approved, it can be easily integrated into the authorities' control environment by accessing the webUI as an administrator. This way, possible users inside the port administration, e.g., the bathymetry department, can be assigned permanent roles, while temporary users, such as owners of visiting ships, can request a time-restricted access to book an antigrounding service or an inspection service. The operators are also responsible for choosing the UV available to perform the service and ensuring that the legal and safety procedures for performing such operations are met. A communication channel between the UV and the control infrastructure is established only if the following procedures

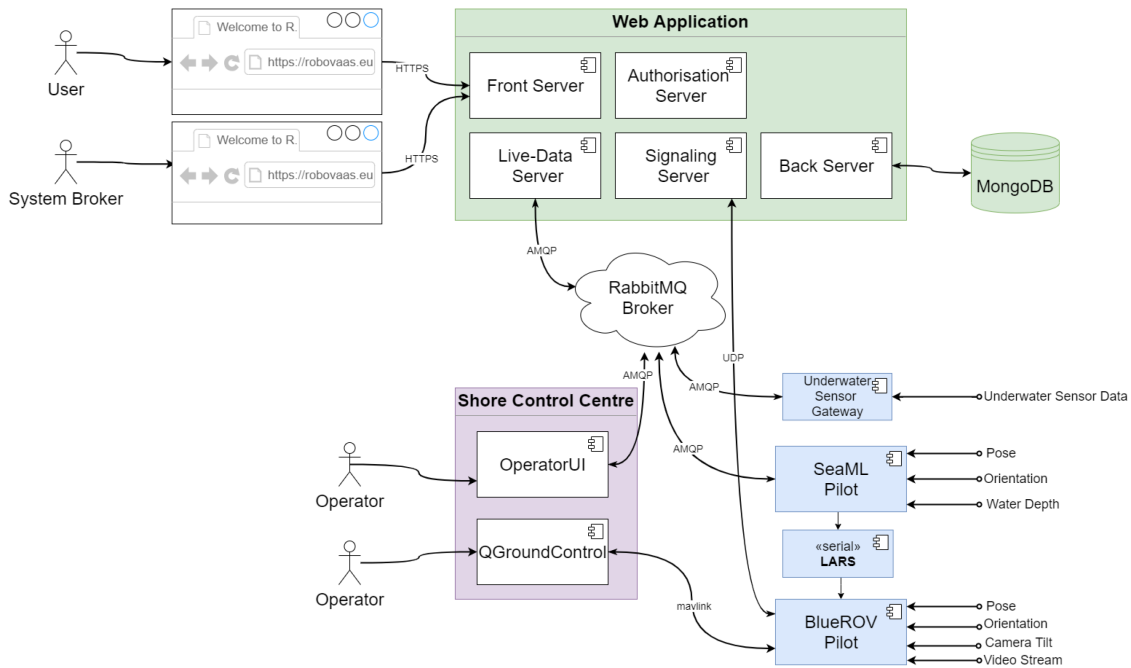


Fig. 2. RoboVaaS System Architecture: Users and System Brokers use the web interface to connect to the Front Server, which, in turn, redirects all requests, made through interaction with the user interface, to the responsible server. Operators use dedicated software for a direct command of the robotic system. The live data are exchanged through RMQ, which regulates the access rights to the queues. The continuously changing access rights to the live data are managed by a dedicated server that oversees the ongoing missions and actors.

are carried out: an operator has been assigned to perform the task, a free UV ready to be deployed is selected by the assigned operator, and the operator has confirmed starting the task.

A. System Architecture

The overall system, depicted in Fig. 2, is decomposed into multiple client applications and server-side microservices. The core subsystems that are used for performing any of the proposed on-demand services are the following.

- 1) the web application: a collection of Node.js web servers that authorize users to insert or retrieve relevant information into or from a database. Moreover, they implement asynchronous messaging protocols such as SocketIO or Advanced Message Queuing Protocol (AMQP) not only for enabling the client-server API, but also for consuming live data coming from the UVs;
- 2) RabbitMQ (RMQ) Broker for efficient distribution of real-time data between the robotic system and the human operator or web-based clients;
- 3) a web-real-time-communication-based server for real-time delivery of video packets from UVs to web-browser clients;
- 4) a file server for reviewing high-quality video footage;
- 5) stand-alone applications, distributed along some single-board computers (SBCs) and comprising the control system of the UVs and their clients having direct access to their operators' inputs.

B. System Functionality

The RoboVaaS system is intended to be commanded within a local area network, through operator applications offering a remote human-machine interface to the deployed robotic system. Remote users can interact with the RoboVaaS system by generating requests to handle

specific missions on a web-based user interface. The latter requires nothing else than a standard internet browser to use the application. Provided that their user account exists and has the permissions to generate requests for one or more use cases, they will automatically create an entry in the job lists of the central database. Each user will register and authenticate through a stand-alone application or through the browser-based client, which will send a request to the Back Server, which has exclusive reading and writing access to the central database. The request is executed by using a Representational State Transfer API. Once authenticated, the user receives a unique user identification number and can generate any number of task requests for the service types that have been granted. In the RoboVaaS vision, specific service types can be granted to users based on the type of subscription they possess. Information such as task descriptions, user roles, and the acquired data is safely logged into a MongoDB, which is a non-SQL database. The Back Server enforces business logic and delegates the job of verifying authentication tokens to the authentication server. MongoDB was preferred to more standard SQL databases due to its flexibility for storing data and, hence, shorter development time. All data are encoded in JSON format, which allowed a loose definition of the table structure and easy modification of the expected input based on the use case, without redesigning the relational diagrams or refactoring the database structure. The drawback is the limitation in terms of querying data belonging to various entities (e.g., multiple robots) and lumping the information efficiently. This could become a problem if more complex relationships between tables were expected, in which case relational databases would be preferable. In addition, live data are handled through an RMQ Broker, which binds the queues of the parties for packet multiplication and routing. While some user types, such as the ASV or the operator, have for safety reasons permanent access rights to the RMQ exchanges and queues, others, such as web-based clients, have their account dynamically generated by the Back Server and only consume live data. The business logic forces each client to

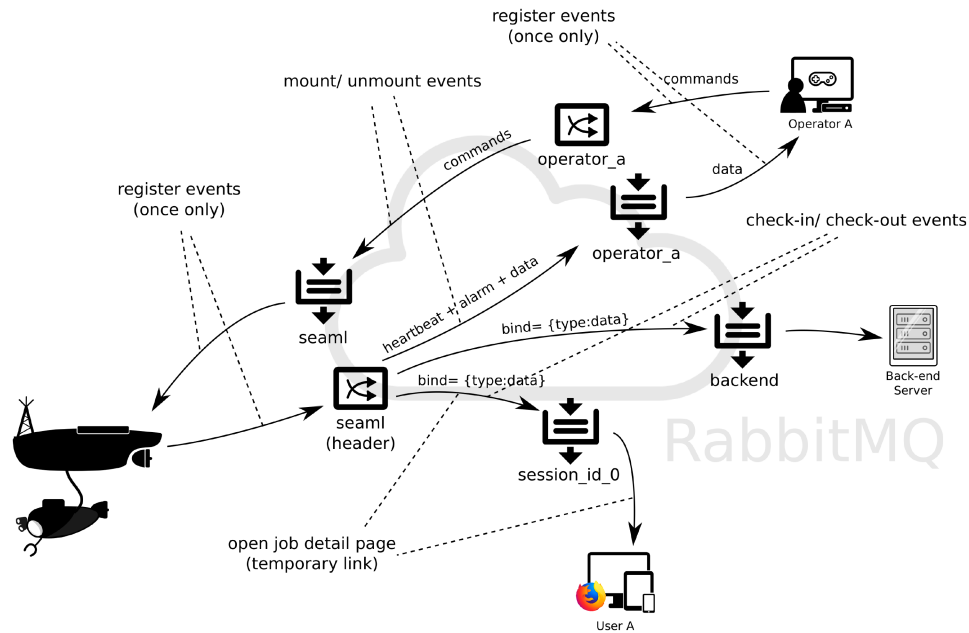


Fig. 3. Access control scheme for RoboVaaS Application: users interacting from the web user interface are allocated temporary queues by the RMQ server for listening to live data. In parallel, queues are generated for the operators, who are also given access keys to write into the robotic system queues, where commands are submitted.

generate its queues following a specific format and to bind them to the exchanges only when the task is started by the operator and confirmed by the Back Server. Should a third party try to listen to the exchange handling the communication between the operator and the robot, its access would be immediately denied by the RMQ Broker, as it had not received permission for its user from the Back Server. In that sense, the Back Server makes use of the RMQ's Hypertext Transfer Protocol API to set the access rights accordingly.

The summary of the access rights control is depicted in Fig. 3.

C. Underwater Network

An underwater network enables the communication between ASVs and battery-powered underwater sensors used to collect environmental data to monitor a certain area. Indeed, electromagnetic signals propagate only up to a few meters under water, while optical communications are affected by sunlight noise and water turbidity, providing a very short range in typical shallow water scenarios [43]. Acoustic signals, instead, can propagate up to several kilometers under the sea, at the price of a low data rate, large propagation delay, and poor performance in the case of multipath, high shipping activities, or noise caused by wind waves [7] or snapping shrimps [44]. In addition, acoustic communication with mobile nodes is strongly affected by the Doppler effect [10].

In this project, the underwater communication is set up with the *ahoi* low-cost and low-power acoustic modems [10] and a complete communication stack implemented using the DESERT Underwater framework [45]. DESERT is a network simulation and emulation tool, based on NS-MIRACLE [46], which implements a complete ISO/OSI modular stack for underwater communications on top of Network Simulator 2 (ns2). An important feature of this framework is the *RealTime scheduler*, which overwrites the existing ns2 scheduler and synchronizes the program with the machine internal processor, so that the simulator can operate in real time with other programs or external devices while using the same protocols implemented for simulations.

Concerning the transmission of packets through the medium, a physical module was implemented to act as driver with the *ahoi* modems: more details on the implementation can be found in [47].

1) *Uwpolling Protocol*: As previously stated, the underwater network is used to collect data from sensor nodes. Specifically, during the underwater data collection service, an ASV patrols the selected area to gather the data from the underwater sensors deployed in the network. For this purpose, *UWPOLLING* [12], a polling-based MAC protocol, has been designed to guarantee fair channel access to all nodes. The protocol assumes up to three types of nodes in the network:

- 1) the sensor nodes, which are equipped with sensors to collect data from the surrounding environment;
- 2) the vehicle acting as a mule, that can be either an ASV or an AUV, which is in charge of moving around in the network and collecting the data from the sensors;
- 3) the sink node, which collects data from the vehicle and forwards it to the shore through an above-water radio frequency (RF) link.

Depending on the scenario, the vehicle can also act as the sink of the underwater network, e.g., an ASV equipped with both acoustic and RF modems capable of collecting data from underwater sensors and directly forwarding it to the shore. Being this the configuration used in the lake test, in Fig. 4, we present the state machine of the underwater sensor node and of the mobile node, i.e., the ASV. The complete protocol state machine when the system is used with an external sink is presented in [12]. The polling protocol works in two subsequent phases: the discovery phase and the polling phase. The first is used by the ASV to let the other nodes know about its presence; the second is employed for the data collection from the sensors and eventually to forward the data to the sink.

The discovery phase starts when the vehicle sends a TRIGGER packet to the surrounding nodes. The sensor nodes and the sink that correctly received the packet from the ASV can decide whether to reply with a PROBE packet, based on their availability of data to transmit. If the node has data to send to the vehicle, it transmits the PROBE after a random backoff uniformly chosen between a minimum (T_{bmin}) and a

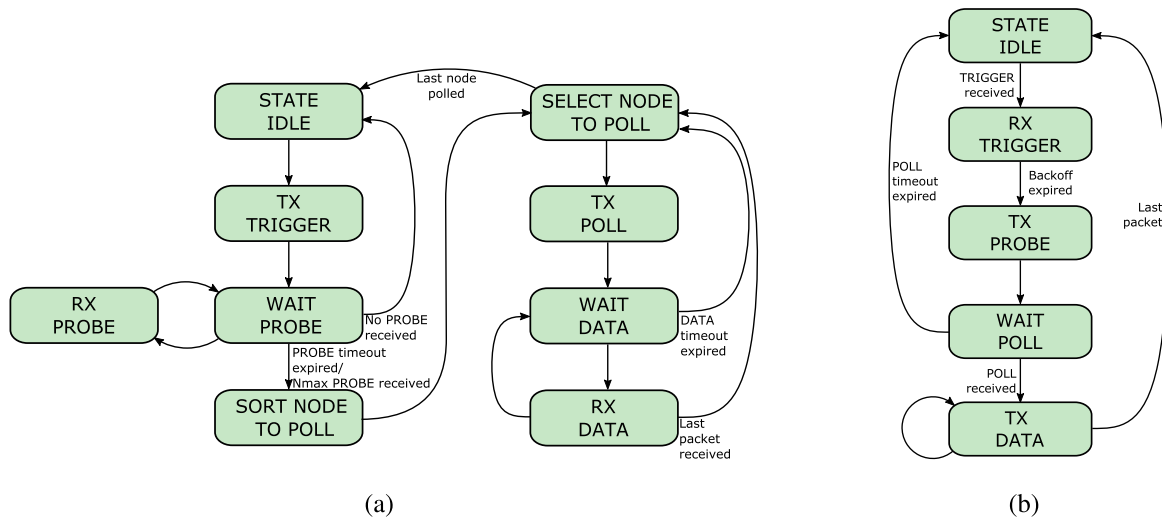


Fig. 4. State machine of (a) the vehicle and (b) sensor node for the *UWPOLLING* MAC protocol implemented in the DESERT Underwater framework.

maximum value ($T_{b_{max}}$). The upper and lower limits of the backoff value are inserted in the TRIGGER packet and can optionally be adapted by the ASV based on the estimated node density [12]. To let the ASV know the amount of time to allocate for each of the nodes, the number of packets the node is going to transmit in the polling phase is inserted in the PROBE packet.

After the reception of the PROBE packets from the surrounding nodes, the ASV starts the polling phase by creating the poll list. The list provides the order in which the nodes will be polled by the ASV, i.e., the order in which each node will transmit its data. The poll list is created based on proportional fair scheduling, which aims to provide almost the same possibility of transmission to all nodes, but without penalizing too much the overall throughput of the network [48]. Indeed, proportional fair scheduling decides the node order based on both the packets transmitted so far by each node and the number of packets a node is going to transmit (more details can be found in [12]). Once the poll list is created, the vehicle starts to poll one by one all the nodes in the list. Specifically, the ASV sends a POLL packet to the first node in the list and waits for data from that node. The intended node, once it has received the POLL, sends the data packets to the ASV. The vehicle waits until either all the intended packets have been received or a timeout related to the amount of time allocated for that node expires. Then, the ASV starts to poll the next node in the list up to the last one. Once all the nodes have been polled, the discovery phase starts again.

2) *Ahoi Modem*: To perform the underwater acoustic transmissions, we selected the *ahoi* modem, a low-cost and low-power acoustic underwater modem [10]. The modem was developed for the integration into micro AUVs and underwater wireless sensor networks. It consists of three stacked printed circuit boards (PCBs) with a size of $50 \times 50 \times 25$ mm and an external commercial hydrophone. For the underwater nodes, the Aquarian Audio AS-1 hydrophone [49] was used, which is the default hydrophone for the *ahoi* modem. The price of a single *ahoi* modem is around € 600 (€ 200 for the PCB and component costs and € 400 for the hydrophone). Furthermore, the *ahoi* modem uses a frequency-shift-keying-based data transmission with a 25-kHz bandwidth centered at 62.5 kHz. The transmission range between 50 and 75 kHz was selected based on the hydrophone characteristics. In addition, the communication band is above the frequencies that contain most of the acoustic noise produced by near vessels and AUVs [50].

D. Above-Water Network

The above-water section of the network has been designed and implemented using equipment manufactured by Mikrotik [51]. Hereafter, we give a brief description of the topology of the above-water network deployed. On the ASV, a Mikrotik Metal 52AC Wi-Fi CPE [52] has been adopted. The Metal 52AC device is a router, mounting a Gigabit Ethernet port, a Wi-Fi 802.11b/g/n and 802.11ac compliant Wi-Fi radio (configured to act as a client), and a 6-dBi omnidirectional antenna. The Metal device has been connected with an Ethernet cable to the ASV's hardware used to forward the data acquired from the underwater network to the above-water network. On the pier, we used mANTBox 2 12s [53] as the Wi-Fi AP, which mounts a 12-dBi 120° directional antenna and an 802.11b/g/n Wi-Fi modem, able to cover long distances and challenging radio channels. We employed a five-Ethernet-port Mikrotik hEX [54] as the core router, which connects the mANTBox, the RabbitMQ servers, and, optionally, laptops for troubleshooting and monitoring, thus allowing the end-to-end connectivity between the Tinkerboard and the servers. We preferred to design a flat network, with Mikrotik hEX acting as the core router and providing a dynamic host configuration protocol (DHCP) server for the entire network, assigning a /24 Classless Interdomain Routing for the entire network. The entire network is, then, on the same "layer 2 domain" avoiding the burden of static routing configuration on hEX. The assigned subnet is large enough (254 addresses available) to be scalable in this scenario and to allow many laptops to connect for monitoring and troubleshooting, using both the Wi-Fi network created by mANTBox (in-band with data transmission from Metal AC) and dedicated Ethernet ports on hEX. To extend the number of available Ethernet ports in this case, an unmanaged switch connected with Mikrotik hEX can be used.

E. Data Compression and Live Data Generation

In addition to DESERT, other two programs have been implemented for the underwater telecommunication pipeline:

- 1) DATA_SENS, an application that acquires data from either a real or a mocked sensor, formats it to a compressed string, and sends it through the DESERT application layer;
- 2) NET_BRIDGE, the application used in the ASV to forward the collected sensor data from the underwater network to the RoboVaaS Cloud.

TABLE I
DATA_SENS STRING FORMAT

| Parameter | ID | TIMESTAMP | DATATYPE | VALUE |
|---------------------|----|-----------|----------------|----------------|
| Format | XX | HHmmss | <type initial> | <value> |
| Size [bytes] | 2 | 6 | 1 | 4 ^a |

^a4 bytes are used for a temperature sensor measurement; other sizes might be used for other data types.

DATA_SENS interfaces with the DESERT framework at the application level through a module that generates a socket connection, aiming to communicate with an external program that generates the data to be transmitted following the protocols in DESERT.

The main reason DATA_SENS sends the data through the acoustic channel with a formatted and compressed string with a fixed size rather than standard JSON formatted strings is due to the limitation of the acoustic channel, which allows only the transmission of small data packets.

DATA_SENS, therefore, creates a string for each sensor measurement formatted, as presented in Table I. The first 2 bytes of the string compose the ID of the sensor node that generated the data, the following 6 bytes the time stamp at which the data was generated, formatted in hours minutes seconds, one byte is used to represent the data type (temperature, salinity, etc.) acquired by the sensor, and, finally, the remaining bytes are used to represent the data value. For instance, in the case of the temperature data type, 4 bytes are used for the value: with this data compression, the underwater nodes transmit temperature data with a packet of 25 bytes, including the header introduced by DESERT. For other data types, the length depends on the number of bytes used to represent the value.

NET_BRIDGE, instead, parses the sensor packets received by the ASV and converts them into a JSON file that is then sent to the RoboVaaS cloud by using an AMQP client that transmits the JSON file by using the above-water network.

For example, the string sent by node 16 through the underwater network at time 14:23:45, with temperature data with value 18.5 °C, would be 16142345T18.5, which is translated at the ASV into this JSON file, where the date is either the current date or the day before, as we assume that the data has been generated no more than 24 h before the current time: {

```

    "buoy_id" : "16,"
    "data_type" : "temperature,"
    "recorded_at" : "2021-03-
02T14:23:45Z,"
    "value" : 18.5
}
```

V. PRELIMINARY IN-LAB TRIALS

To perform some preliminary tests of the system and evaluate its functionality before the actual sea trial, we performed some in-lab hardware-in-the-loop simulations using a digital twin of the ASV. This methodology allowed us not only to test each component of the system separately, but also to perform an integration test of all software and hardware components not directly related to the vehicle's control and navigation system, with consequent reduction of testing costs and time, thereby significantly reducing the probability of failure during the final sea trial demonstration. The model-based designed techniques make use of simulators for low-cost design validation. In addition, communication between different subsystems is abstracted within the simulator environment, allowing for realistic testing and optimization of the system. Consequently, we developed a digital twin of the real system for both the carrier ASV and the underwater equipment.

The Gazebo [55] simulator was used to simulate the full-body dynamics of the ASV employed to perform the data-muling exercise, while DESERT was used to emulate the exact behavior of the underwater network that acts as in a real exercise. For the ASV model, we opted for an open-source Heron [56] model from ClearPath Robotics. The model, depicted in Fig. 5, makes use of Gazebo's UUVSim [57] plug-in for rendering the hydrodynamic forces exerted upon the ASV. Given that the physical properties of Heron are reasonably close to the SeaML ASV, as they have very similar size and shape, the Heron model was used without any adaptation. The development of a digital twin of the SeaML ASV is out of scope and will be sought in further works, by changing the model definition in the ASV description package. The tradeoff is needed to tune the weights of the course controller to the Heron ASV while not having any guarantee of similar performance for the SeaML ASV. The controller output is redirected to the motor set points of the Heron ASV, and the outputs of the emulated sensors, specifically the Global Navigation and Satellite System positions and filtered inertia measuring unit values, are read by the position and course controller, respectively.

Because the low-level controller is platform independent, the companion software as well as the operator application could be tested on standard laptops and/or computers, provided that the connection to the RoboVaaS server was available. After following the standard ASV booking procedure, the companion software (where the low-level controller is running) opens an RMQ queue and binds it to the exchange where the output of the underwater communication network is broadcast by the emulated sensor node.

To test the signal chain from the underwater sensing equipment, which was hosted in Italy, and communication to the radio network and finally to the online server, hosted in Germany, we developed an in-laboratory test facility. At the SIGNET laboratory (University of Padova), we developed a small tank, with a capacity of about 200 L, which has sound absorption capabilities through its internal coating [58]; this tank does not allow one to test topologies but is very useful to test all the hardware and configurations before the sea trial, without stressing the devices with in-air transmission. Fig. 6 shows the setup: the *ahoi* transducers inside the tank are presented in Fig. 6(a), while the scheme of the tested setup is depicted in Fig. 6(b). The underwater network was composed by two nodes: one emulating a sensor node and the other emulating the ASV communication system (we remark that the ASV motion and control system, instead, is emulated by the aforementioned digital twin). Both nodes were composed of an *ahoi* modem and an ASUS Tinkerboard S SBC, where the whole communication framework was running. In addition, the node emulating the ASV was equipped with a Metal 52AC antenna to transmit the data to the emulated shore station. In this way, we could test and debug inside our laboratory the same equipment and tools used in the trial.

In the first test, the shore station and its AMQP server were installed in a Raspberry Pi 4 SBC directly connected to the emulated ASV through a Wi-Fi link established connecting the Raspberry Ethernet port to the hEX router that, using the mANTBox antenna, reached the node connected to the Metal 52AC.

In the second test, the Raspberry Pi was removed, and the hEX was connected to the Internet so the synthetic data were able to reach the public RoboVaaS RMQ server in Germany. Because the underwater network and the above-water link were simulated in Italy, while the Gazebo simulation and the RoboVaaS service platform were deployed in Germany, a publicly available RMQ server was deployed and the underwater network and Gazebo simulations were synchronously started.

On the user and operator endpoints, there is no need for any change, as the entities of the RoboVaaS cloud infrastructure continue to listen to the same AMQP queues. The RMQ clients were configured to

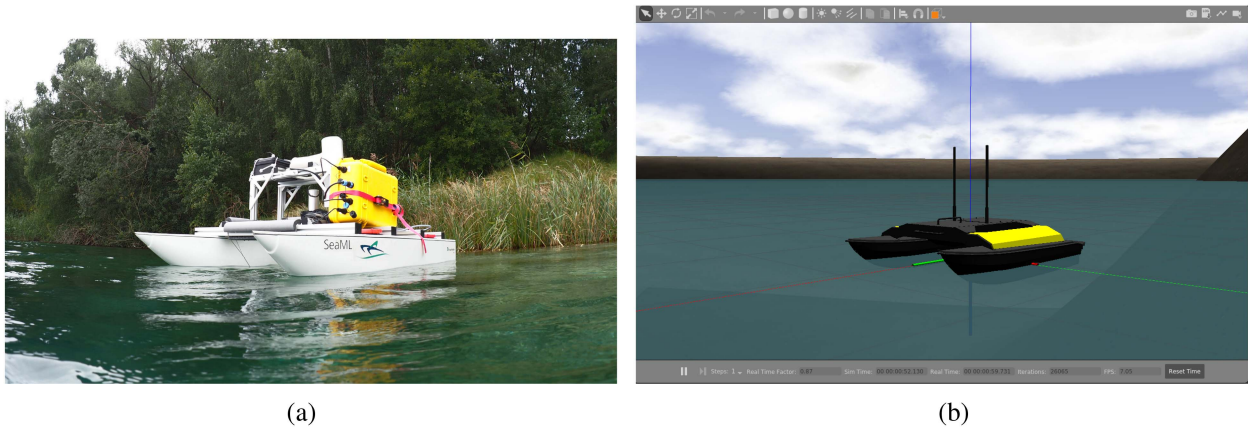


Fig. 5. ASV simulation setup: thanks to the very similar physical properties, the Heron ASV has been used for in-lab tests. By abstracting the hardware of the ship, most of the RoboVaaS system components could be cost-effectively tested. Underwater communication is separately simulated and connected to the RoboVaaS system via publicly accessible RMQ server. (a) SeaML ASV. (b) Heron digital twin.

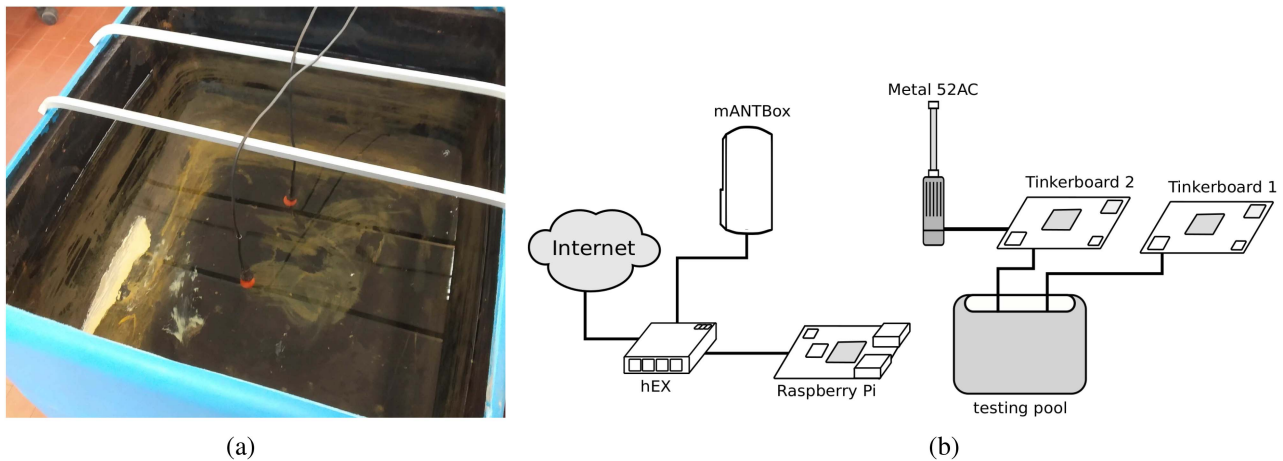


Fig. 6. *ahoi* laboratory setup: while the two underwater nodes and the long range above-water network were installed in Padova, the AMQP server was deployed in Germany and reached through the Internet. (a) Phono-absorbent testing tank. (b) In-lab setup scheme.

communicate over the same publicly available RMQ server, using the predefined queues and message structure. Fig. 7 presents the web interface observed by the operator to monitor the mission status: the red markings on the map represent the path followed by the ASV, each point being a *heartbeat* message successfully received by the server. The green markings are also successfully acknowledged messages, but they also indicate the presence of a sensor reading coming from the underwater equipment. We underline that from the operator's perspective, there is no difference between the emulation and the actual sea trial.

VI. FIELD TESTS: SETUP AND EXERCISE DEFINITION

Field tests were performed at lake Kreidensee in Hemmoor, Lower-Saxony, Germany during a ten-day campaign. The setup used the same software tools and applications to gather sensor data and send control commands, but distributed over the multiple companion computers and/or laptops. As observed in Fig. 2, the onshore equipment included a mobile station acting as a server, supporting the main above-water communication brokers and saving the critical mission information into the database. Other mobile stations were used to run the applications that were directly commanding the UVs. For the offshore counterparts, which were previously either simulated or

run in confined environments, besides the ASV, five buoys with ballast to avoid drifting were manually deployed in the different topologies. Regarding the mobile node, the SeaML ASV carried all the necessary equipment to perform all the services, which varied from a single-beam echosounder for the environmental data collection use case to the offshore underwater communication station for the data-muling use case and, lastly, to a mini-ROV for the quay wall inspection use case. Nevertheless, only the data-muling use case is detailed in this article. Regarding the specific equipment used for the different use cases, the SeaML ASV used an EchoLogger EU D24 dual-frequency single-beam echosounder for gathering depth data in the antigrounding and the bathymetry data collection use cases, an acoustic modem to retrieve the data from the sensors in the data-muling use case, and a commercial ROV from BlueRobotics for the quay wall inspection use case. Conversely, the ship hull inspection use case was demonstrated with a big inspection-class ROV provided by the University of Limerick and equipped with the Kraken SeaVision system. With data muling being the most challenging use case from the communication perspective, in this article, we focus on this specific use case. The main challenges in the other use cases were on the hardware integration. Specifically, in the inspections use case, the mini-ROV had to be deployed from the SeaML ASV with a miniaturized crane: the SeaML could support this operation in calm water, but in the presence of waves and strong

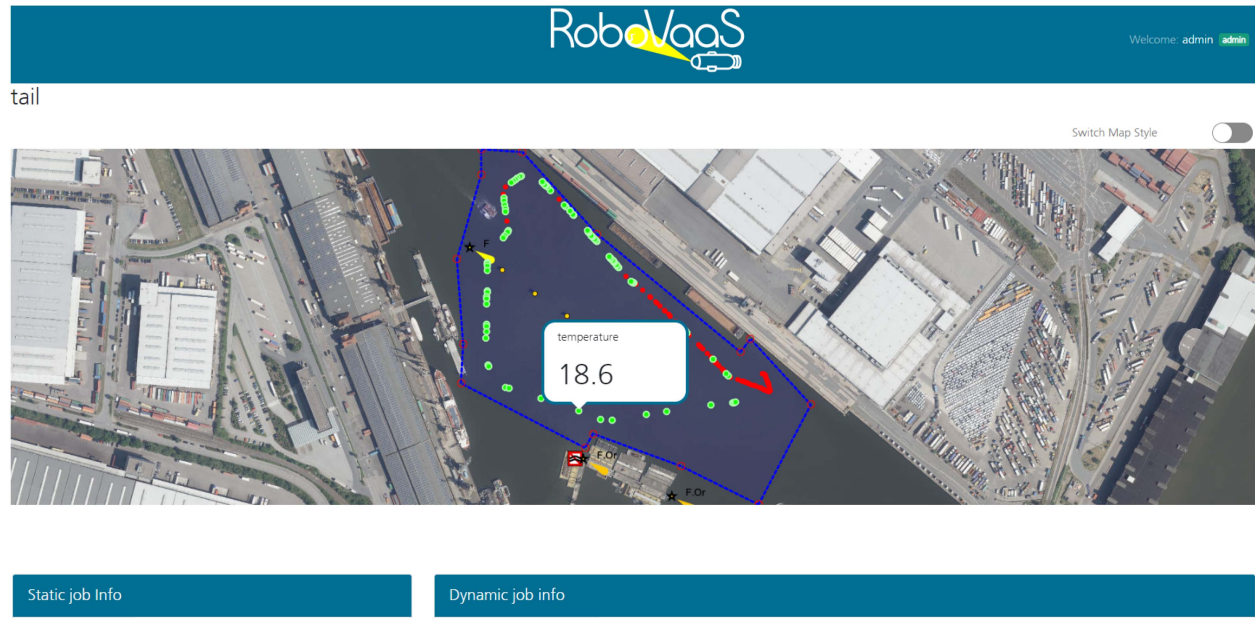


Fig. 7. Simulation results.

current, a larger ASV had to be used: for this reason, a new and larger ASV, called SeaLion [59], was developed by Fraunhofer CML.

The weather conditions were quite stable during the whole test campaign. Specifically, during the day, the temperature ranged between 15 and 25 °C and the weather was sunny or cloudy all days. The water temperature at the surface was 19 °C and on the lake bottom was 16 °C. Almost no wind was observed, and there were no waves in the lake.

A. Shore Operation Center

On the shore, the following hardware components were used to deploy the software solution.

- 1) The server-side application was deployed on a Dell Latitude 5411 laptop running Ubuntu 18.04.
- 2) The operator application, shown in Fig. 8, was deployed on a Dell Latitude 7480 laptop running Windows 10.
- 3) The underwater network was controlled and monitored from a Panasonic Toughbook CF-53, running Linux Mint 18.1 and connected via SSH to the buoys.
- 4) The above-water communication network was deployed on a Mikrotik hEX PoE lite, which had a 4G modem attached for internet connectivity and Mikrotik mANTBox 2 12S Wi-Fi antenna.

B. Robotic System

The carrier board was the already mentioned SeaML ASV, which was designed to fit different measurement equipment and even a mini-ROV. The absolute maximum payload is 25 kg, which was never attained during the tests for all use cases mentioned. The main components of the barebone ASV are as follows:

- 1) four T200 propulsion units from BlueRobotics (two on each side);
- 2) two Raspberry Pi 4 SBCs: one for position estimation and one for computing the command primitives from the high-level commands received from the shore. The first made use of Emlid Navio2 and Reach M+ boards as data acquisition boards and

made use of its own low-level controller for fusing sensor data (instead of using an onboard sensor fusion module);

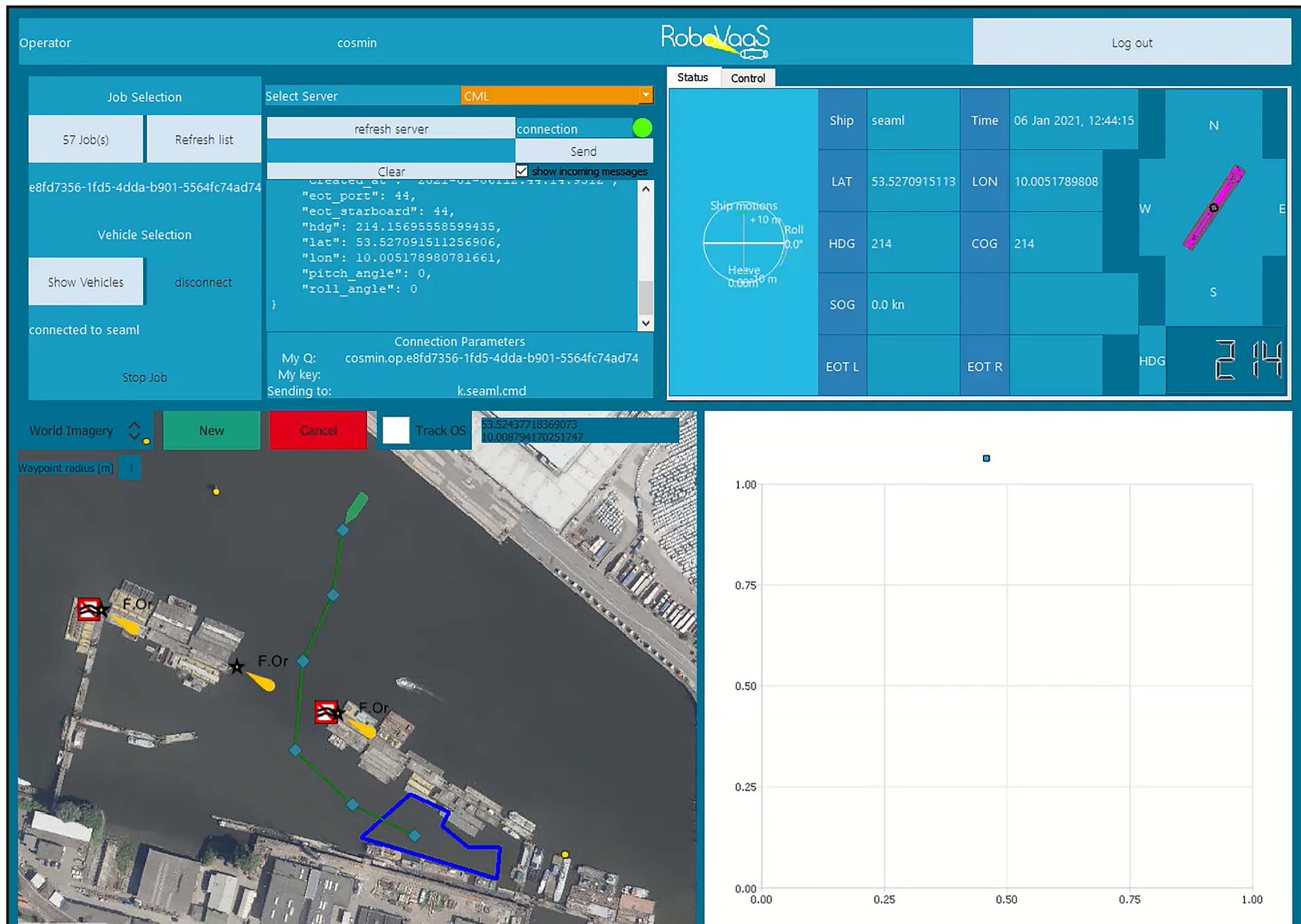
- 3) one Netgear GS305 switch for connecting all clients to the above-water network through the Mikrotik Metal 52AC;
- 4) one FrSky X8R RC receiver coupled to the propulsion unit for emergency takeovers.

Saving a total of nominal 1184 Wh for the power transmission and another 148 Wh for the data and logic, the SeaML ASV was able to perform the tasks without battery exchange for 8–12 h, a time interval that was sufficient to perform four to eight exercises. The obtained autonomy cannot be considered as absolute, because the time intervals also included equipment exchange, calibration, and on-site debugging time in which the propulsion units were either little used or not at all.

All additional components could be connected to the control unit via watertight RJ45 connectors that forwarded the connections to the (unmanaged) switch. In the next subsections, the layout and interfaces of the components specifically aimed for the data-muling use case are presented.

C. Above-Water Network Setup

The underwater communication and all the offshore devices were supported by a radio network operating in the 2.4-GHz Wi-Fi band and implemented using Mikrotik devices. The IP address pool used by the DHCP server was ranging from 192.168.2.55 to 192.168.2.100, while the address of the DHCP server was 192.168.2.1, and the address of the wired interface of the long range shore Wi-Fi antenna was 192.168.2.2: the whole network connections and addressing are depicted in Fig. 9. The 12-dBi 120° aperture antenna was deployed in the shore at a height of approximately 15 m from the water level, pointing the area where the SeaML ASV was moving. The omnidirectional antenna in the ASV was installed on top of the vehicle at a height of approximately 0.75 m. Given the long range requirement, the 20-MHz subchannels of the 802.11n protocol were used. The effective radiated power (ERP) of the antennas was set to 100 mW, according to the European restrictions [60]. A detailed description of the above-water network topology and the devices used in the network have been described in Section IV-D.



(a)



(b)

Fig. 8. Operator Console contains two stand-alone applications that are used to command and control one ASV and one ROV, respectively. The ASV console is the in-house developed *OperatorUI*, while for the ROV the open-source *QGroundControl* application is used. (a) *OperatorUI* App for high-level commanding SeaML ASV. (b) *QGroundControl* App for ROV teleoperation.

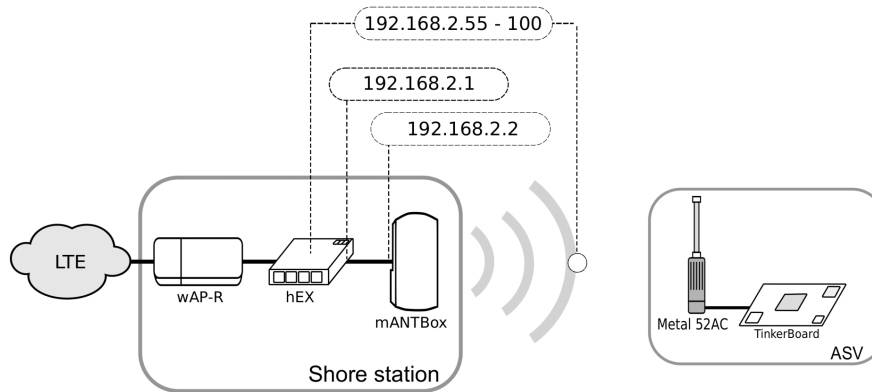


Fig. 9. Above-water network architecture. The shore station (left) exchanges data with the ASV (right) using a long range Wi-Fi link and acts as the gateway of the network providing all nodes Internet connectivity through LTE.

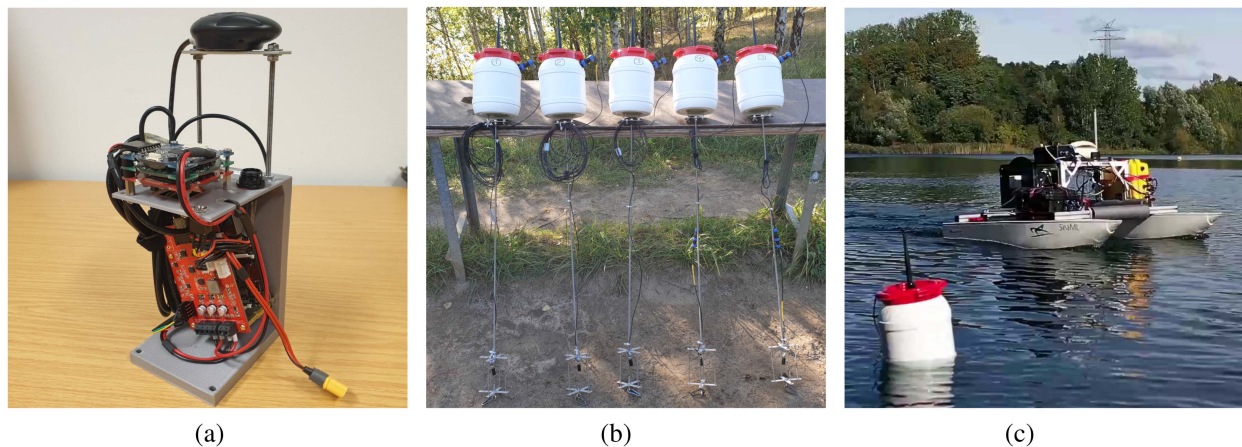


Fig. 10. For the evaluation and demonstration of the data collection service, a prototype node based on a buoy was built. (a) Electronic components. (b) Buoys with hydrophones. (c) Deployed buoy and SeaML.

D. Underwater Nodes

The underwater nodes for the data-muling service are equipped with sensors to measure environmental data, a power supply, a processing unit, and the *ahoi* acoustic underwater modem. Usually, the nodes are submerged in an area of interest and do not have any cabled connectivity or above-water communication link. In a preliminary test performed to evaluate the underwater node functionality in combination with the *ahoi* modems, we experienced a communication range between 150 and 250 m in static scenarios and up to 150 m in mobile scenarios.

To evaluate the data collection service, we decided to use buoys instead of fully submerged nodes to simplify the deployment. A buoy consists of an ASUS TinkerBoard S used to execute the DESERT Underwater framework, an *ahoi* modem, a Navilock NL-6002U GPS receiver, and a power supply. An external Wi-Fi antenna connected to the TinkerBoard enables debugging and monitoring during the evaluation. However, Wi-Fi connection and GPS receiver are just part of the prototype for the service demonstration with buoys and would be removed from the actual nodes in the case of a long-term deployment. For power supply, a battery (11.1 V, 2400 mAh) and a power management board are used. The power management board provides a stabilized 5-V supply to the TinkerBoard, measures the battery voltage and current, and protects the components against undervoltage, reverse polarity, and short circuits. Depending on the computational load of the TinkerBoard and the transmission intervals of the *ahoi* modem, the battery allows an

operation time between 6 and 10 h. The external hydrophone is placed about 1.1 m under the buoy to avoid the region directly under the water surface that is strongly affected by acoustic reflections. Furthermore, a cage protects the hydrophone against physical damage. Finally, a rope is connected to the protection cage to fix the buoys in the water with an anchor.

For the deployment, five buoys were constructed. Fig. 10 depicts electronic components, the buoys before the deployment, and a single buoy with the SeaML ASV. In addition, a sixth node was prepared, using a Raspberry Pi 4 board connected to an *ahoi* modem. The electronic components were installed in a box to place the node on a jetty and submerge the hydrophone at 1-m depth.

To collect environmental data from the sensor nodes, the SeaML was equipped with an *ahoi* modem and a TinkerBoard as well. Similar electronic components from buoys were installed in a waterproofed case on top of the SeaML. Furthermore, a hydrophone in a protection case was mounted 0.9 m under the SeaML. To measure the position of the mobile and the fixed node on a jetty, a Navilock NL-8001U GPS receiver was used.

E. Underwater Network Settings

The underwater network is composed of one mobile node and a number of static nodes that ranges from four to six, depending on the

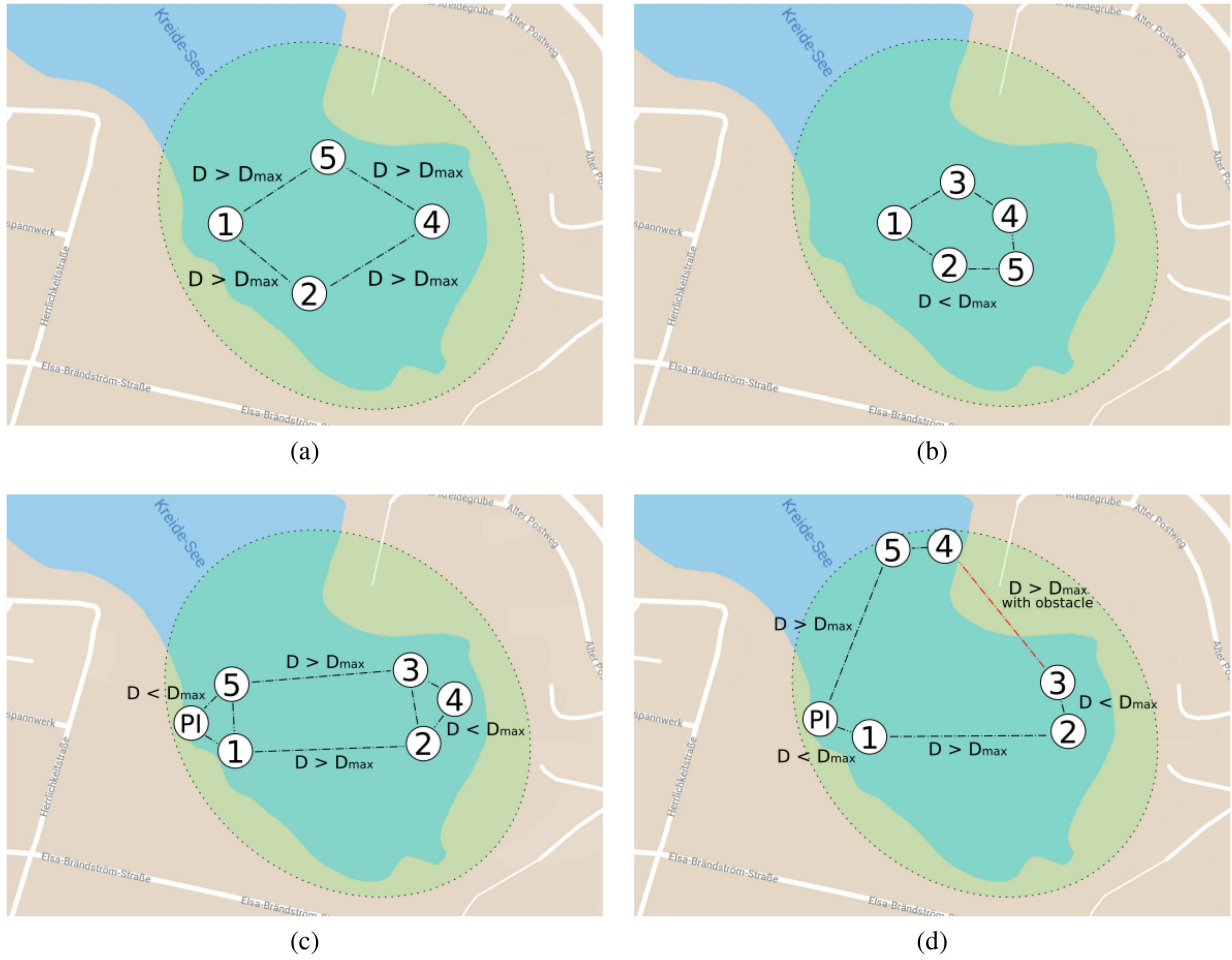


Fig. 11. (a)–(d) Topologies represented with the required minimum distances.

considered network topology. In our scenario, every 60 s, each static node generates a packet with a payload of 13 B (if we consider the packet headers introduced by the communication protocols, the size increases up to 25 B). The transmission bit rate used by the *ahoi* modems is 200 bit/s.

To collect the data from the sensor nodes, the *UWPOLLING* MAC protocol has been used. An important parameter of this protocol is the maximum backoff time $T_{b_{max}}$ used in the discovery phase by the sensor nodes to randomize the channel access trying to reduce the collisions (more details are provided in Section IV-C1). In our tests, we set $T_{b_{max}} = 15$ s. In addition, each node can transmit in each protocol cycle (i.e., every polling phase) a burst of up to five consecutive packets, to limit the number of packets sent by a node in each cycle and, thus, reduce the probability that the ASV moves out of range during a packet burst transmission.

F. Data-Muling Topologies

As previously mentioned, the underwater network relied on the DESERT Underwater framework to accomplish the network protocol management. The communication is performed with acoustic signals, so that the sensors can be placed even on the seabed and still transmit wirelessly to the surface. Moreover, the data are transmitted using the DESERT Framework, whose packet header contains the required information for the routing and its packet payload is formatted to include

all the needed information for the data acquisition, as explained in Section IV-E.

To better understand the behavior of the *UWPOLLING* MAC protocol, different network topologies have been tested to analyze how the protocol behaves under different network conditions, such as node density and distances. The performance has been assessed in terms of fairness and packet delay, and the overall underwater network has been evaluated taking into consideration the PDR of the data packets.

To perform this analysis, we designed four different topologies to reproduce four networks with different node densities. In particular, we designed the networks to obtain clusters with a different number of nodes to test the behavior of the discovery phase of the *UWPOLLING* protocols. Based on the maximum range D_{MAX} of the *ahoi* modem (about 150–250 m) and the area of operations in Fig. 11, we depicted the following topologies.

- 1) *Equally distanced nodes with $D > D_{MAX}$:* In this topology, all the adjacent nodes have a distance greater than the transmission range; thus, the ASV would be in range with no more than two sensor nodes at the same time. This topology is shown in Fig. 11(a).
- 2) *Equally distanced nodes with $D < D_{MAX}$:* In this topology, the nodes are placed close enough to obtain a single cluster with all the available nodes, such that the ASV can be in range with all the sensor nodes during the entire test. The goal is to let all nodes

participate to the same channel contention during a discovery phase. This topology is shown in Fig. 11(b).

- 3) *Two clusters at $D > D_{MAX}$* : In this topology, there are two main clusters, composed of three sensor nodes each and placed at a distance such that when the AUV is close to one cluster, it is not in range with the other one. With this topology, we obtain a network in which one of the clusters is not visited for a while and accumulates packets in the internal queue until the next visit, when the muling node returns in range. This topology is shown in Fig. 11(c).
- 4) *Three clusters at $D > D_{MAX}$ and physical obstacle*: In this topology, there are three clusters, composed of two sensor nodes each and placed at a distance larger than the communication range; in addition to the previous case, two clusters are divided also by a physical obstacle (in the Hemmoor test, it was a small headland), so that there is a certain communications shielding between the ASV and the sensor nodes in the other clusters, even while traveling to them. This topology is shown in Fig. 11(d).

VII. FIELD TEST: RESULTS

The analysis of the results is covered in this section, considering the underwater and above-water networks separately. The performance of the motion controller of the SeaML ASV is not evaluated and is considered out of scope. The data-muling exercises are fully dependent on the carrier platform and its ability to integrate the incoming data from the underwater network with those from navigation, and on the flawless correlation of real-time data with the rest of the exercise data contained in the database. Indeed, the parameters to assess the performance differ since the underwater and above-water channels are different by nature, and the protocols are built taking into account the characteristics of the two channels, leading to lower throughput and higher delays for the underwater network.

A. Service Application Performance Analysis

In the process of collecting environmental data from the underwater nodes, several points that define the vision of RoboVaaS have been achieved. First of all, the ASV has proven to be a reliable platform carrier for all the services we considered. Although within the scope of the project only one ASV for the small-scale evaluations has been developed and, subsequently, the equipment had to be adapted for each use case, the changes could be rapidly made thanks to the modular design.

There were in total 16 complete test runs for the data-muling use case, where each time the area of interest was selected by a web-browser client and approved by the system broker, while the path computed by the ASV to patrol the area was finally set by the operator. Physically, the roles of the web-browser client and of the system broker were played by the same person, because both demanded little interaction and access to a web browser. The operator role was played by another person, who constantly monitored the job execution and focused on avoiding collisions or loss of signal. In the case of emergency, the operator was equipped with an RC remote, which could take over control from the autopilot and sail the ASV by manually steering it. This was, however, seldom used because the above-water network proved to be very reliable even around corners where some natural elements, such as trees, were blocking the direct contact to the ASV. Second, the chosen location had no ship traffic, so most of the obstacles were static, which allowed precise path planning. Each time, the operator chose the appropriate path for the ASV to follow, so that all sensor buoys were slalomed before it returned to shore. To evaluate the performance of the system in a mission that is as close as possible to reality, the path was changed

even for the same topology; the distances from the ASV and the sensor buoys were always longer than 2 m to avoid collisions due to sway.

The average speed of the ASV was around 1.5 kn (0.78 m/s), and the paths ranged from 700 m to 1.2 km. Because a waypoint-following algorithm was used and the catamaran-shaped ASV is an underactuated system, the paths could not be perfectly followed. The effects of velocity perpendicular to the direction of motion (known as sway) increased the total distance followed, but this additional distance was, hence, neglected. As observed in Fig. 12, the packets from the underwater nodes have been constantly collected during the exercise, with increasing frequencies when the ASV was within a 60-m radius from the nodes.

Each packet of data received from the underwater network was incorporated in the messages delivered to the shore over AMQP via RMQ. The ASV incorporated the JSON-encoded messages into a JSON object containing the keyword *data*. The front-end server handling the web user interface of the web-browser client then handled the object as a sensor point and marked it accordingly on the map shown to the user. Third, the contents were saved by the web application back end into the database with the appropriate exercise-specific identification number; further metrics could be shown, as depicted in Fig. 13. Finally, the web-browser clients could download the plotted data in comma-separated values (CSV) format for further analysis.

B. Above-Water Network: Performance Analysis

An evaluation of the above-water network was taken separately during the last days of tests and were performed, at first, considering only the actual coverage of the mANTBox Access Point. We recall that the 2.4-GHz Wi-Fi channel was used with a 20-MHz bandwidth, and the transmission power was configured to the maximum allowed by the European Wi-Fi regulations (100 mW) [60]. To evaluate the Wi-Fi coverage, we set a continuous PING command from the SeaML ASV to the mANTBox and manually drove the SeaML in different places of the lake and saved the data in local log files.

We use iperf3 [61] software to measure the performance of the network. Specifically, we use iperf3 to transmit a TCP and a UDP stream between the two antennas and measure the maximum achievable throughput, computed as the number of payload bits received at the application layer per second, and the jitter that provides the variation of the packet delivery delay (PDD) and can be computed as [62]

$$\text{jitter} = \sum_{i=1}^{N_v-1} \frac{|d(i+1) - d(i)|}{N_v - 1} \quad (1)$$

where N_v is the number of packets received for the considered video stream and $d(i)$ is the delay of the i th packet. This metric is very important for some applications such as video streaming: specifically, a video is considered to be smooth enough to remotely control a UV when the packet jitter is below 2 ms [63].

The obtained outcome is shown in Fig. 14, where different zones are defined based on the delay/jitter retrieved by the PING command with respect to the positions of the ASV. The antenna is 120° directional, but includes secondary lobes that permit transmissions at a wider aperture. The green area represents the zone where the link is stable and expands up to a range of approximately 350 m. In the yellow area, the link is unstable, while in the red area, the link is disrupted. Specifically, the average throughput available in the green zone is 19.1 Mbit/s, and the jitter measured with iperf3 is less than 0.3 ms, and therefore, it is small enough to permit a real-time stream video with standard resolution. Instead, in the border zones (yellow and red), the average throughput available is 17.3 Mbit/s, but the connection is really unstable with many disruptions (half of the packets were lost in the yellow area and most of the packets were lost in the red area) and the jitter oscillates between

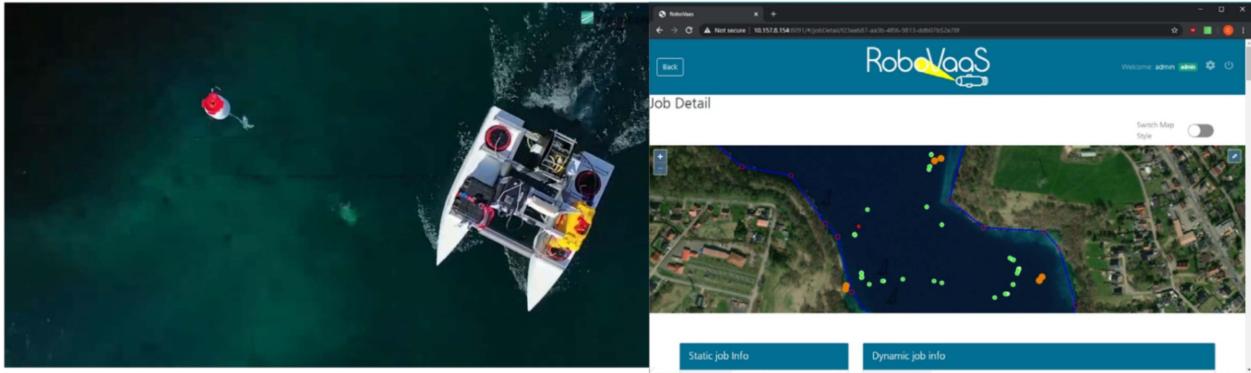


Fig. 12. Underwater data collection service results visualization obtained in real time from the messages acquired by the ASV from the underwater wireless nodes and sent to the RMQ queues generated for this exercise. Points without sensor data, previously marked with red dots, have been removed from visualization.

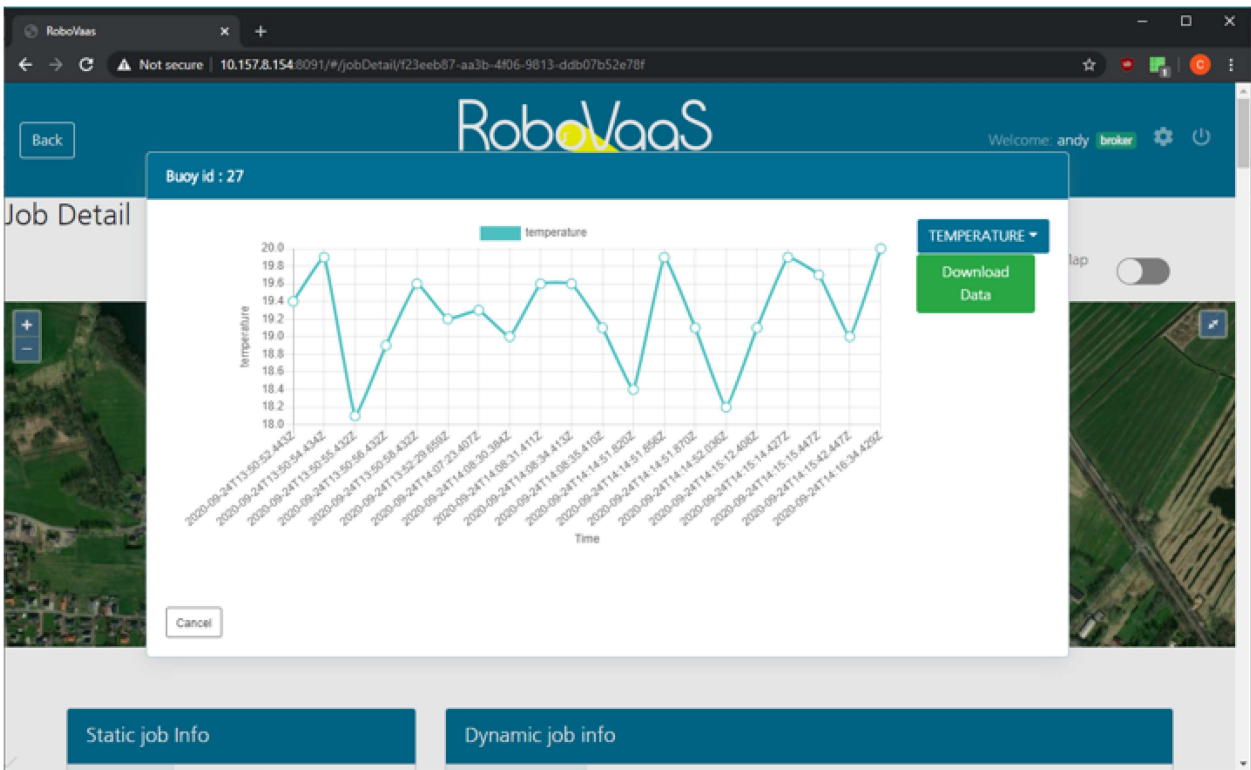


Fig. 13. Graphical representation of underwater data collection service on the web user interface. Visualization generated online for each buoy and updated in real time.

0.4 ms, when close to the green area, and 1.9 s, when the link is disrupted and then reestablished, depending on the position of the mobile antenna, which means, in quality-of-experience terms, that several packets could be lost during the transmission and a video stream would be laggy. Furthermore, in the red zone, the connection appears discontinuous, which leads to long periods where the devices try to establish a new connection.

In terms of coverage, which depends on the transmission power, the RoboVaaS scenario is restricted by the European ETSI regulation [60] that imposes a maximum ERP = 100 mW. With this configuration, the use of a 40-MHz channel, that provides a higher throughput,¹ did

¹We measured a throughput of 50 Mbit/s for 40 MHz with the antennas placed at a distance of up to 100 m between each other.

not allow us to cover the required range for RoboVaaS (at least a few hundreds of meters). Therefore, due to the throughput requirements of less than 10 Mbit/s (see Section II), the whole evaluation was performed only with the 20-MHz channel that matched the RoboVaaS needs.

C. Underwater Physical Layer: Performance Analysis

Before the underwater network trials with the DESERT Underwater framework, preliminary tests were done to analyze the physical transmission with the *ahoi* modems. During the tests, all modems were connected via Wi-Fi to a central computer. The central computer saves all received acoustic packets and triggers the packet transmissions. In all cases, packets with an overall size of 6 B each were used. Previous

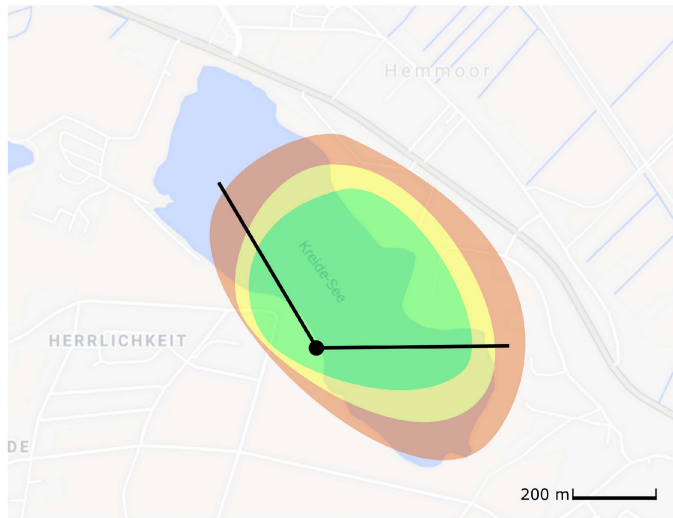


Fig. 14. Coverage map of mANTBox Wi-Fi over Hemmoor Lake. A stable connection was achieved up to a maximum range of 350 m from the shore station.

evaluations pointed out that the bottleneck of *ahoi* modems is the synchronization phase at the beginning of the data transmission [50]. Based on that, short packets were used to analyze the communication links.

The involved modems transmit in a loop. Every modem transmits a single packet, and each following transmission starts 500 ms after the end of the previous transmission due to the long propagation time of acoustic signals and long channel delay spread. For each data-muling topology (see Section VI-F), the transmission links between the static nodes are tested. During the evaluation, high PDRs for ranges up to 150 m are observed. On average, the PDR is 90.1% (3605 received packets out of 4000) in this region. Also, for longer static communication links up to 350 m, a high reliability is measured, e.g., 66% (2377 received packets out of 3600) for communication distances between 300 and 375 m. However, for long distances (≥ 150 m), the PDR has a strong dependence on position. At some positions, PDRs below 15% are observed.

Furthermore, the communication link between static nodes (buoys and node on the jetty) and the mobile node (SeaML) is analyzed. For the test, the SeaML travels a route of 743 m in the area of the static nodes. The trial takes 16 min and each node transmits 171 acoustic packets with 6 B header, 0 B payload, and 500-ms delay after each transmission. In sum, 1197 packets are transmitted. Fig. 15 displays the PDR between SeaML and static nodes for different distances. The numbers on top of the bars indicate the number of received (rx) and transmitted (tx) packets. Static communication links between the static nodes are not counted in the bar plot.

The communication from SeaML to the static nodes is better than the communication from the nodes to the SeaML. In both cases, the PDRs for distances from 0 to 75 m are between 90% and 98%. Conversely, the PDR for distances from 75 to 100 m is 74% in the first case (SeaML to static nodes) and 53% in the second case (static nodes to SeaML). For longer distances from 100 to 300 m, the PDRs are between 17% and 40% (SeaML to static nodes) and between 0% and 17% (static nodes to SeaML).

In summary, the communication range in static scenarios is longer than in the mobile case. The relative velocity between the transmitter and the receiver produces a frequency shift at the receiver side (Doppler effect) and a synchronization error. The synchronization error increases with the packet duration.

The reason for this asymmetrical behavior in the mobile scenario is the acoustic noise produced by the thrusters of the SeaML; hence, the transducer onboard the vehicle experienced a higher noise than the transducers deployed in the static nodes. The SeaML uses four BlueRobotics T200 thrusters, which are mounted on the left- and right-hand sides of the SeaML at a distance of about 1.1 m from the hydrophone. Fig. 16 depicts the received power spectral densities (PSDs) for different scenarios. In all cases, the *ahoi* modem recorded the signal over 163.84 ms (32 768 samples with 200-kHz sampling rate) with a fixed gain. It is important to note that the signal is recorded at the analog-to-digital converter after the analog signal processing chain. Based on that, the signal is affected by the hydrophone characteristic, a preamplifier, a bandpass filter, and a second amplifier. For more information about the analog signal processing, the reader is referred to [10]. At first, the received signal with disabled thrusters is measured [see Fig. 16(a)]. The yellow area indicates the transmission bandwidth between 50 and 75 kHz. All analog components are tuned to provide a flat transmission characteristic and a low-noise spectrum in the transmission bandwidth. Therefore, other regions are not optimized due to the low-cost and low-power requirements. Afterward, the second diagram shows the recorded signal, while the SeaML is traveling at maximum speed (during the experiments). Fig. 16(b) depicts a higher noise level compared to the stopped SeaML. In the communication bandwidth from 50 to 75 kHz, the average noise level increases from -34.5 to -18.2 dB/Hz. In addition, the thruster noise produces a large peak around 72 kHz. The acoustic noise from the thrusters decreases the signal-to-noise ratio (SNR) between acoustic communication signal and background noise. Therefore, the communication reliability from the SeaML to the static nodes (without thruster noise at the receiver side) is better than the communication from the static nodes to the SeaML. For the sake of completeness, Fig. 16(c) and (d) represents PSD during a packet reception without and with thruster noise. The communication distances are 70 and 68 m, and the average sound levels in the communication bandwidth are -12.8 dB/Hz (without thruster noise) and -10.3 dB/Hz (with thruster noise). In this example, the SNR (w.r.t. the average values in the communication bandwidth) is 21.7 dB without thruster and 7.1 dB with thruster noise. The communication distance of about 70 m is still in an area with high PDR [see Fig. 15(b)]. However, longer communication distances would result in lower SNR and, therefore, a lower PDR. Other conditions that usually impact the acoustic communications, such as transducers depth, motion, changes in wind speed, and water salinity, can be excluded from the explanation for the asymmetrical behavior. The depth of the transducers was fixed to approximately 1 m below the surface thanks to the metal frame used for the deployment. Motion-based Doppler effects affected the communication link in both directions (static nodes to ASV and ASV to static nodes) and does not explain the asymmetric behavior. In addition, wind and waves were rarely noted and constant in the evaluation area. In addition, water temperature and salinity were measured with a professional CTD-48 probe manufactured by Sea&Sun Technologies near to the node PI (see Fig. 11). At 1-m depth, the water temperature was 18.1 °C and the salinity was 0.24 ppt, resulting in a speed of sound of 1476.6 m/s. Based on the narrow evaluation area and small fluctuations (no water inflow or outflow), temperature, salinity, and speed of sound can be assumed to be constant and to affect the communication link in a symmetric way.

D. Underwater Network: Performance Analysis

The underwater scenario is characterized by high propagation delay and bit error rate, combined with a low bit rate determined by the frequencies available for communications. Thus, on the one hand, the overall throughput is orders of magnitude lower than that observed

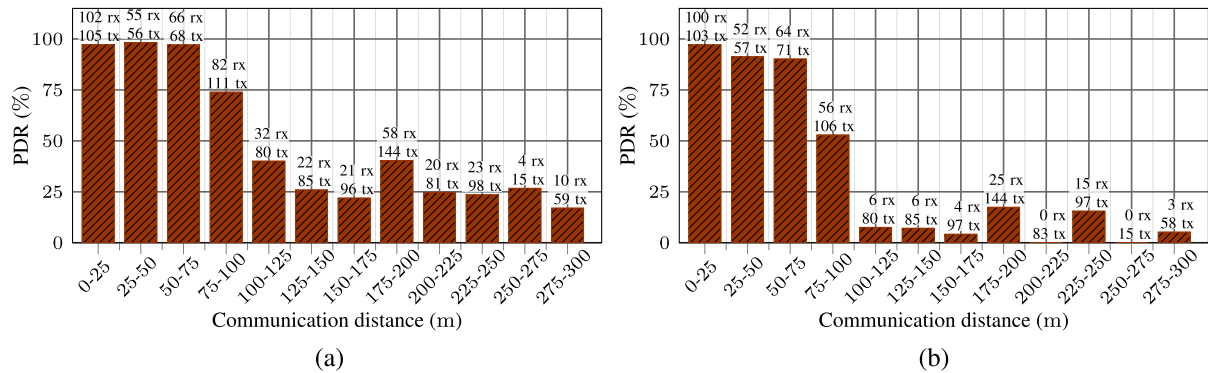


Fig. 15. PDRs between a mobile node (SeaML) and static nodes (five buoys and node on a jetty). (a) SeaML (transmitter) \Rightarrow static nodes (receiver). (b) Static nodes (transmitter) \Rightarrow SeaML (receiver).

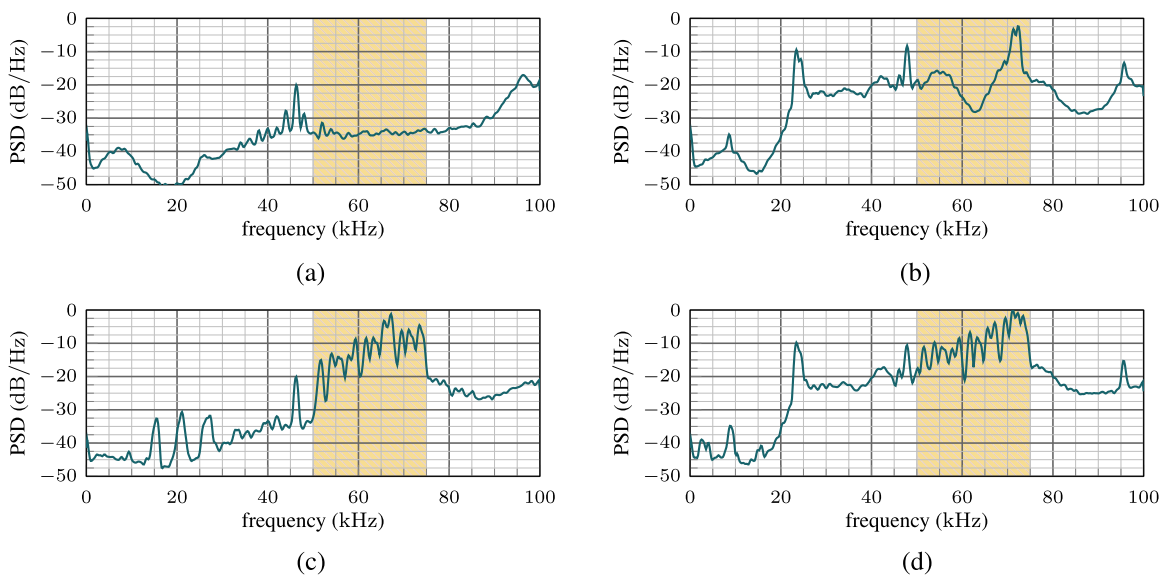


Fig. 16. Thruster noise and communication sampled by the *ahoi* modem. The yellow area indicates the transmission bandwidth, and the plots are normalized to a maximum of 0 dB/Hz. (a) Stopped SeaML (without thruster noise). (b) Driving SeaML (with thruster noise). (c) Stopped SeaML and communication (70-m dist.). (d) Driving SeaML and communication (68-m dist.).

in wireless terrestrial networks. On the other hand, the PDD, defined as the time elapsed from the packet generation to the correct packet reception at the destination, in underwater acoustic networks is in the order of several seconds or minutes, not of milliseconds like in terrestrial networks. These phenomena, typical of delay-tolerant networks such as satellite and underwater acoustic networks [64], are even more evident in a data-muling scenario, where a mobile node collects the data from submerged sensors when it enters their communication range, as the link between a sensor and the mobile node is systematically disrupted as soon as the mobile node moves out of the sensor's coverage area. Another important metric that shows how the network worked during the tests is Jain's *fairness* index (JFI) [65], which indicates whether all the nodes were treated fairly by the network protocol stack. Specifically, JFI is computed as

$$\text{JFI} = \frac{\left(\sum_{i=1}^{N_{\text{nodes}}} P_{rx,i} \right)^2}{N_{\text{nodes}} \sum_{i=1}^{N_{\text{nodes}}} P_{rx,i}^2} \quad (2)$$

where N_{nodes} is the number of nodes in the network and $P_{rx,i}$ is the number of data packets received by the ASV from node i .

TABLE II
METRICS FOR TOPOLOGY 1

| Topology 1 | Sent | Received | PDR [%] | PDD [s] |
|------------|------|----------|---------|---------|
| Node 1 | 48 | 38 | 0.791 | 501.46 |
| Node 2 | 63 | 55 | 0.873 | 326.55 |
| Node 4 | 65 | 60 | 0.923 | 438.97 |
| Node 5 | 70 | 58 | 0.828 | 417.30 |

In Tables II–V, we present the network performance for the four different topologies.

As defined above, the PDD is the delay after which a packet is correctly received and is computed starting from the packet generation time. Thus, for the topologies where the nodes are not always in the ASV range, the delay also includes the time the vehicle takes to visit all

TABLE III
METRICS FOR TOPOLOGY 2

| Topology 2 | Sent | Received | PDR [%] | PDD [s] |
|------------|------|----------|---------|---------|
| Node 1 | 60 | 48 | 0.8 | 81.16 |
| Node 2 | 59 | 39 | 0.661 | 74.49 |
| Node 3 | 60 | 48 | 0.8 | 54.7 |
| Node 4 | 60 | 37 | 0.616 | 81.08 |
| Node 5 | 59 | 46 | 0.779 | 74.41 |

TABLE IV
METRICS FOR TOPOLOGY 3

| Topology 3 | Sent | Received | PDR [%] | PDD [s] |
|------------|------|----------|---------|---------|
| Node 1 | 57 | 42 | 0.737 | 331.39 |
| Node 2 | 59 | 46 | 0.779 | 187.57 |
| Node 3 | 60 | 34 | 0.566 | 97.42 |
| Node 4 | 60 | 44 | 0.733 | 237.95 |
| Node 5 | 59 | 37 | 0.616 | 296.81 |
| Node PI | 60 | 32 | 0.533 | 296.81 |

TABLE V
METRICS FOR TOPOLOGY 4

| Topology 4 | Sent | Received | PDR [%] | PDD [s] |
|------------|------|----------|---------|---------|
| Node 1 | 50 | 28 | 0.560 | 233.89 |
| Node 2 | 50 | 41 | 0.820 | 571.40 |
| Node 3 | 60 | 37 | 0.616 | 267.01 |
| Node 4 | 57 | 38 | 0.666 | 438.29 |
| Node 5 | 56 | 41 | 0.732 | 255.55 |
| Node PI | 51 | 37 | 0.725 | 255.55 |

the other nodes and to come back. This can be observed for topologies 1, 3, and 4, where the average PDDs of the networks (i.e., the PDD averaged over the nodes) are 421.07, 241.33, and 336.95 s, respectively, against the average network PDD of Topology 2 equal to 73.17 s. Indeed, in Topology 2, the mean delays per node are far smaller than in the other topologies, because all nodes are in range of the ASV approximately during the whole test; thus, the reception time for the packets is determined mainly by the time the *UWPOLLING* protocol takes to poll every node that successfully takes part in the discovery phase.

From the analysis of the log files obtained during the test campaign, we observe that the PDRs change for the different topologies and depend on both preamble synchronization problems (as stated in Section VII-C) and asymmetry of the acoustic link due to the ASV movement and the

noise produced by its thrusters. Indeed, in some cases, a few packets are lost inside the burst of five packets sent by a node in its polling phase, while in other cases the burst of packets is completely (or almost completely) lost. In the last case, most of the control packets exchanged by the vehicles and the nodes (i.e., TRIGGER, PROBE, and POLL) are correctly received, but then the link quality drops faster due to the movement and the asymmetry mainly caused by the ASV propellers' noise; therefore, the data packets are lost. While in the first case, packet loss happens independently in one or more packets inside the burst of five data packets, in the second case, the whole burst of five packets (or at least the last part of a burst) is lost. Depending on the topologies and on the specific node position, the prevalent cause of packet loss can be different. In Topology 1, the PDR at the MAC layer (i.e., without considering the packets that remained in the queue at the end of the simulation) is higher than 80% for all the nodes. In this case, except for node 5, the main cause of packet loss is a failure in the preamble synchronization. Even in Topology 2, the main cause of packet loss is the failure in the synchronization. Indeed, in this case, the ASV is in range with all the nodes most of the time, and therefore, the ASV movement does not have much of an impact on the reception rate. On the other hand, the channel asymmetry and the ASV movement become the prevalent cause of packet loss in Topologies 3 and 4, where the ASV moves between clusters of nodes that are not within range of each other or are even separated by a physical obstacle (as in Topology 4).

Fig. 17 digs deeper into the relation between the ASV positions and the received packets, showing the actual ASV position for each received packet from each node in all the four analyzed topologies. In the figures, the packets received by the ASV have a different marker for each transmitting node, i.e., yellow diamonds for the packets received from Node 1, green stars for those received from Node 2, blue hexagons for the packets received from Node 3, purple circles for the packets received from Node 4, black dots for the packets received from Node 5, and orange triangles for the packets received from Node 6. In topologies where the distance between nodes or between clusters of nodes is greater than the maximum transmission range, the cluster formation is well defined in proximity of the interested nodes, like in Topologies 1, 3, and 4. Specifically, in Fig. 17(a), we can observe the network behavior in Topology 1, where the ASV can receive the packets only when it is in the proximity of a single node; hence, the derived figure shows a strong clustering of the packet receptions. Instead, in Topology 2 depicted in Fig. 17(b), the reception of packets by the sink node can happen even when the ASV was on the opposite side of the network, meaning that all the nodes are in range during most of the experiment, depending on the instantaneous channel conditions. In Topology 3, presented in Fig. 17(c), the cluster formation is divided into two regions that correspond to the two main groups of nodes at the opposite sides of the lake. Some of the packets are received even in the middle of the lake since the ASV was still in range during their transmission. Finally, Fig. 17(d) shows the received packets in Topology 4. Also, in this case, the three clusters are well defined. Still, there are some packets received even during the travel of the ASV from one cluster to another, as in Topology 3, but this occurs less often than in the previous case, especially between groups 4-5 and 2-3, since the ASV was traveling in a clockwise path between the different groups. Thus, group 2-3 is shielded by the natural headland until the ASV reached at least the middle of that path segment.

Finally, the JFI for each topology is reported in Table VI: the closer the index to one, the more fairly the protocol treats the nodes of the network. The results show that all nodes are treated fairly by the polling protocol, since the JFI ranges from 0.976 to 0.987 for all network topologies.

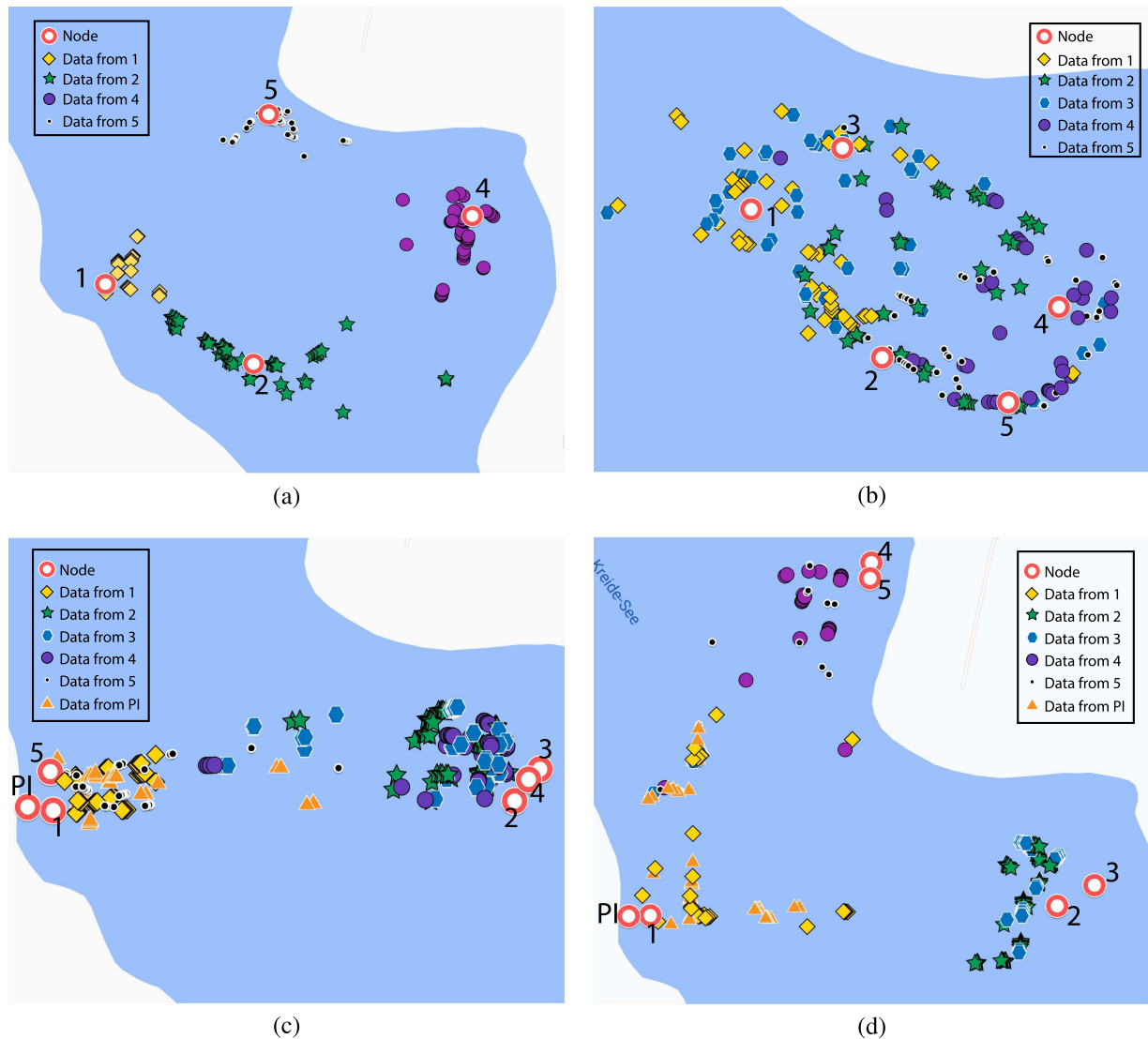


Fig. 17. (a)–(d) ASV’s positions for each received packet for all topologies.

TABLE VI
FAIRNESS INDEX FOR EVERY TOPOLOGY

| | Topology 1 | Topology 2 | Topology 3 | Topology 4 |
|-----|------------|------------|------------|------------|
| JFI | 0.9755 | 0.9869 | 0.9874 | 0.9842 |

VIII. CONCLUSION

This article presents a detailed procedure for enabling the usage of robotic systems, such as, but not limited to, ASVs or ROVs in industrial applications with modern web-based visualization tools and a robust communication infrastructure. Starting from the conclusions obtained in [13] in the design and implementation phases, the overall communication infrastructure system has been tested in simulation and deployed for lake trials. This has first of all validated the communication infrastructure concept for TRLs 4 and 5 and enabled proceeding to the final stage for obtaining TRL 5 and 6 through a public demonstration

in an industrial environment, a task that was accomplished with the live demo performed during the ITS World Congress in October 2021 in the Port of Hamburg [22].

While the above-water and underwater communications have been discussed in detail as the theme of this article, the presented results can and should be considered in the overall context of the RoboVaaS project. The experimental results presented were all collected using the service architecture successfully integrated for the four use cases presented in Section II. This includes the web user interface, or webUI, where each topology run was set up by a job request and then processed via the back-end applications supporting the RoboVaaS service and assigned to the SeaML as the ASV to perform the job. All the data were saved in the RoboVaaS database and is available for postprocessing. The SeaML ASV, developed and intended as the modular demonstration platform, was successfully deployed for the ten days of the test campaign and performed all the required tasks.

The introduction of web services enabled rapid access to experimental data by multiple devices, regardless of their operating system or processor architecture, while still maintaining control over the access rights into the database and keeping the robotic systems secure from

hacking. The security is obtained, first, through logic enforcement done by the back-end services, which authenticate each request via a secret token generated at login and through exposing the front-end server port over secured web sockets. With the exception of the Front Server, all other server-side components are accessible solely within local area network, where trusted clients are assumed. The clients within the local area network are assumed to be given access by the services provider, while the Wi-Fi Protected Access II (WPA2) encrypted wireless link that connects the robotic system being monitored only allows a limited number of devices to simultaneously access the system (in the current sea trials, this number was limited to 20). To further enhance the security aspects of the wireless network, we envision the possibility to use Extensible Authentication Protocol, adopting the so-called WPA2 Enterprise, using, for example, the IEEE 802.1x implementation [66]. The RMQ broker applied the access rights to the exchanges and queues dynamically enforced by the central server. This procedure shifted the focus to the roles of each user, because any mission required to have an operator and a robot assigned in order for the service to be started. This, in turn, enabled the rapid development of all use cases within the RoboVaaS project. The use-case definition has to be incorporated into the high-level control system of the robotic system, inside the central server's logic and inside the operator's stand-alone application. This process could be automated through message templates, similar to the message-type primitives characteristic of ROS-based systems.

By using a non-SQL database, such as MongoDB, the advantage in terms of flexibility of data insertion, migration, or export was counterbalanced by the lower speed of the queries, which needed to parse several thousand JSON objects for each request for historic data. This flexibility allowed loosely defining the database models and relying more on data parsing. This is particularly useful when complex structures, with multiple parameters in exchanged messages, are nested within one another. This has the drawback that, by increasing the complexity of the data that need to be parsed, the performance of the queries decreases significantly: other solutions based, for instance, on SQL databases, where models are defined *a priori*, can overcome this issue at the price of a lower flexibility.

On the quality of service performed with UVs, the data-muling use case successfully combined underwater and above-water communication technologies deployed on an ASV. In comparison to deployment on AUVs, the surface vehicles can cover large areas in short amounts of time like AUVs, while still maintaining reasonably high runtime autonomy.

Having benchmarked the performance of the communication system and validated the overall system through sea trials, the proposed solution reached TRLs 5 and 6. A future goal is to validate the system in an industrial environment and, therefore, reach TRLs 6 and 7. In parallel, the high-level control system of the ASV will be restructured to be ROS compliant and augment different control strategies to enhance the autonomy of the robot. Future work will also focus on improving the PDR of the acoustic modems in a mobile network, where the propellers noise causes a significant performance drop. To this aim, different modulation and synchronization techniques will be investigated to obtain a higher resilience against propellers noise. Finally, additional relevant use cases, such as marine litter cleaning or infrastructure inspection, will be investigated to provide robotic assistance solutions with the help of the modular platform developed and with the above-water and underwater communication system infrastructure.

ACKNOWLEDGMENT

The authors would like to thank Christian Busse (Hamburg University of Technology) for his support to build the underwater nodes.

REFERENCES

- [1] *Robotic Vessels as-a-Service*, RoboVaaS. Accessed: Apr. 2022. [Online]. Available: <https://www.martera.eu/projects/robovaas>
- [2] *Drone Technol. Uses and Appl. for Commercial, Ind. and Mil. Drones in 2021 and the Future*. Accessed: Apr. 2022. [Online]. Available: <https://www.businessinsider.com/drone-technology-uses-applications?r=US&IR=T>
- [3] S. Cavalieri, A. Di Stefano, and O. Mirabella, "Impact of fieldbus on communication in robotic systems," *IEEE Trans. Robot. Autom.*, vol. 13, no. 1, pp. 30–48, Feb. 1997.
- [4] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (MAVlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87658–87680, 2019.
- [5] *Robot Ignite Academy*. Accessed: Apr. 2022. [Online]. Available: https://www.theconstructsim.com/robotigniteacademy_learnros/ros-courses-library/
- [6] V. Markovski, F. Xue, and L. Trajković, "Simulation and analysis of packet loss in user datagram protocol transfers," *J. Supercomput.*, vol. 20, no. 2, pp. 175–196, 2001.
- [7] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM Mobile Comput. Commun. Rev.*, vol. 11, no. 4, pp. 34–43, 2007.
- [8] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Philos. Trans. Roy. Soc. A*, vol. 370, pp. 158–175, 2012.
- [9] J. Alves and J. C. Fricke, "Analysis of JANUS and underwater telephone capabilities and co-existence," in *Proc. IEEE 3rd Underwater Commun. Netw. Conf.*, 2016, pp. 1–5.
- [10] B.-C. Renner, J. Heitmann, and F. Steinmetz, "ahoi: Inexpensive, low-power communication and localization for underwater sensor networks and AUVs," *ACM Trans. Sens. Netw.*, vol. 16, no. 2, 2020, Art. no. 18.
- [11] D. Zordan, F. Campagnaro, and M. Zorzi, "On the feasibility of an anti-grounding service with autonomous surface vessels," in *Proc. IEEE OCEANS Conf.*, 2019, pp. 1–8.
- [12] A. Signori, F. Campagnaro, F. Steinmetz, B.-C. Renner, and M. Zorzi, "Data gathering from a multimodal dense underwater acoustic sensor network deployed in shallow fresh water scenarios," *J. Sens. Actuator Netw.*, vol. 8, no. 4, 2019, Art. no. 55.
- [13] C. Delea et al., "Communication infrastructure and cloud computing in robotic vessel as-a-service application," in *Proc. IEEE/MTS OCEANS, Virtual, Glob. OCEANS Conf.*, 2020, pp. 1–7.
- [14] T. Zugno, F. Campagnaro, and M. Zorzi, "Controlling in real-time an ASV-carried ROV for quay wall and ship hull inspection through wireless links in harbor environments," in *Proc. IEEE/MTS OCEANS Virtual, Global OCEANS Conf.*, 2020, pp. 1–9.
- [15] V. E. Schneider, C. Delea, J. Oeffner, B. Sarpong, H. -C. Burmeister, and C. Jahn, "Robotic service concepts for the port of tomorrow: Developed via a small-scale demonstration testbed," in *Proc. Eur. Navigat. Conf.*, 2020, pp. 1–8.
- [16] V. E. Schneider, M. Constapel, J. Riordan, F. Flannery, H.-C. Burmeister, and C. Jahn, "Towards an investigation of a mass-assisted anti-grounding service through simulated nautical scenarios in a ship handling simulator," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 929, 2020, Art. no. 0 12031.
- [17] *SEAML—Compact, Versatile Autonomous Craft for German Research Body*. Accessed: Sep. 2022. [Online]. Available: <https://www.bairdmaritime.com/work-boat-work/seaml/specialised-fields/unmanned-craft-non-naval/vessel-review-seaml-compact-versatile-autonomous-craft-for-german-research-body/>
- [18] *BlueROV2*. Accessed: Sep. 2022. [Online]. Available: <https://bluerobotics.com/store/rov/bluerov2/>
- [19] *3D Laser Imaging—SEAVISION*. Accessed: Apr. 2022. [Online]. Available: <https://krakenrobotics.com/products/subsea-inspection-tools/>
- [20] *Univ. of Limerick Launches ETAIN ROV*. Accessed: Apr. 2022. [Online]. Available: <https://www.smartbay.ie/article/university-limerick-launches-etain-rov>
- [21] *Water Depth for Ships in the Port of Hamburg*. Accessed: Apr. 2022. [Online]. Available: <https://www.hamburg-port-authority.de/en/waterway/waterside-access>
- [22] *RoboVaaS*. Accessed: Apr. 2022. [Online]. Available: <https://www.youtube.com/watch?v=HbJJC0-dJQ&t=588s>
- [23] C. Petrioli et al., "The SUNRISE GATE: Accessing the SUNRISE federation of facilities to test solutions for the Internet of Underwater Things," in *Proc. Underwater Commun. Netw. Conf.*, 2014, pp. 1–4.

- [24] J. Alves, J. Potter, P. Guerrini, G. Zappa, and K. LePage, "The LOON in 2014: Test bed description," in *Proc. Underwater Commun. Netw. Conf.*, 2014, pp. 1–4.
- [25] X. Zhao, D. Pompili, and J. Alves, "Energy-efficient OFDM bandwidth selection for underwater acoustic carrier aggregation systems," in *Proc. IEEE 3rd Underwater Commun. Netw. Conf.*, 2016, pp. 1–5.
- [26] R. Petroccia et al., "Deployment of a persistent underwater acoustic sensor network: The CommsNet17 experience," in *Proc. MTS/IEEE OCEANS Conf., Kobe Techno-Oceans*, 2018, pp. 1–9.
- [27] R. Costanzi, N. Monnini, A. Ridolfi, B. Allotta, and A. Caiti, "On field experience on underwater acoustic localization through USBL modems," in *Proc. IEEE OCEANS Conf.*, 2017, pp. 1–5.
- [28] R. Martins, J. B. de Sousa, R. Caldas, C. Petrioli, and J. Potter, "SUNRISE project: Porto university testbed," in *Proc. Underwater Commun. Netw.*, 2014, pp. 1–5.
- [29] D. Fernandes, F. Pinheiro, A. Dias, A. Martins, J. Almeida, and E. Silva, "Teaching robotics with a simulator environment developed for the autonomous driving competition," in *Proc. Int. Conf. Robot. Educ.*, 2019, pp. 387–399.
- [30] J. Barbosa, A. Dias, J. Almeida, and E. Silva, "Evaluation of lightweight convolutional neural networks for real-time electrical assets detection," in *Proc. 4th Iberian Robot. Conf.*, 2019, pp. 99–112.
- [31] A. M. Pinto and A. C. Matos, "MARESyE: A hybrid imaging system for underwater robotic applications," *Inf. Fusion*, vol. 55, pp. 16–29, 2020.
- [32] MORUS Project—Unmanned System for Maritime Security and Environment Monitoring. Accessed: Apr. 2022. [Online]. Available: <https://www.researchgate.net/project/MORUS-Unmanned-System-for-Maritime-Security-and-Environmental-Monitoring>
- [33] Smart and Netw. Underwater Robots in Cooperation Meshes. Accessed: Sep. 2022. [Online]. Available: <http://swarms.eu/approach.html#technicalapproach>
- [34] E. Omerdic and D. Toal, "OceanRINGS: System concept applications," in *Proc. 20th Mediterranean Conf. Control Autom.*, 2012, pp. 1391–1396.
- [35] C. Jahn and S. Saxe, *Digitalization of Seaports—Visions of the Future*. Hamburg, Germany: Fraunhofer Center for Maritime Logistics and Services, 2017.
- [36] Y. Chen and H. Hu, "Internet of intelligent things and robot as a service," *Simul. Model. Pract. Theory*, vol. 34, pp. 159–171, 2013.
- [37] Ocean Technol. Campus Rostock. Accessed: Apr. 2022. [Online]. Available: <https://www.igd.fraunhofer.de/en/projects/ocean-technology-campus-rostock>
- [38] R. Toris et al., "Robot web tools: Efficient messaging for cloud robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4530–4537.
- [39] E. Cervera, G. Casañ, and R. Tellez, "Cloud simulations for RoboCup," in *Robot World Cup*. New York, NY, USA: Springer, 2017, pp. 180–189.
- [40] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *IEEE Netw.*, vol. 26, no. 3, pp. 28–34, Jun. 2012.
- [41] F. Favaro, P. Casari, F. Guerra, and M. Zorzi, "Data upload from a static underwater network to an AUV: Polling or random access?," in *Proc. IEEE EOS OCEANS Conf.*, 2012, pp. 1–6.
- [42] W. Liu et al., "APOLL: Adaptive polling for reconfigurable underwater data collection systems," in *Proc. IEEE/MTS Techno-OCEANS Conf.*, 2018, pp. 1–9.
- [43] F. Campagnaro, F. Favaro, F. Guerra, V. Sanjuan, M. Zorzi, and P. Casari, "Simulation of multimodal optical and acoustic communications in underwater networks," in *Proc. IEEE OCEANS Conf.*, 2015, pp. 1–7.
- [44] A. Mahmood, M. Chitre, and H. Vishnu, "Locally optimal inspired detection in snapping shrimp noise," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 1049–1062, Oct. 2017.
- [45] F. Campagnaro et al., "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. IEEE 3rd Underwater Commun. Netw. Conf.*, 2016, pp. 1–5.
- [46] N. Baldo, M. Miozzo, F. Guerra, M. Rossi, and M. Zorzi, "Miracle: The multi-interface cross-layer extension of ns2," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, 2010, Art. no. 761792.
- [47] R. Francescon et al., "An event-based stack for data transmission through underwater multimodal networks," in *Proc. 5th Underwater Commun. Netw. Conf.*, 2021, pp. 1–5.
- [48] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [49] *Aquarian as-1 Hydrophone*, Aquarian Audio & Scientific, 2021. Accessed: Sep. 2022. [Online]. Available: <http://www.aquarianaudio.com/as-1-hydrophone.html>
- [50] F. Steinmetz and C. Renner, "Resilience against shipping noise and interference in low-power acoustic underwater communication," in *Proc. IEEE/MTS OCEANS Conf.*, 2019, pp. 1–10.
- [51] Mikrotik. Accessed: Apr. 2022. [Online]. Available: <https://mikrotik.com/>
- [52] Mikrotik Metal 52AC. Accessed: Apr. 2022. [Online]. Available: <https://mikrotik.com/product/RBMetalG-52SHPacn>
- [53] Mikrotik mANTBox. Accessed: Apr. 2022. [Online]. Available: https://mikrotik.com/product/mantbox_2_12_s
- [54] Mikrotik hEX POE Lite. Accessed: Apr. 2022. [Online]. Available: <https://mikrotik.com/product/RB750UPr2>
- [55] GAZEBO—Robot Simul. Made Easy. Accessed: Apr. 2022. [Online]. Available: <http://gazebo.org/>
- [56] Heron. Accessed: Apr. 2022. [Online]. Available: <https://github.com/heron>
- [57] Unmanned Underwater Vehicle Simulator Documentation. Accessed: Apr. 2022. [Online]. Available: <https://uuvsimulator.github.io/>
- [58] F. Meneghello, F. Campagnaro, R. Diamant, P. Casari, and M. Zorzi, "Design and evaluation of a low-cost acoustic chamber for underwater networking experiments," in *Proc. 11th ACM Int. Conf. Underwater Netw. Syst.*, 2016, pp. 1–8.
- [59] SEAML: SeaLion—Providing Unmanned Robotic Services. Accessed: Sep. 2022. [Online]. Available: https://www.cml.fraunhofer.de/content/dam/cml/en/documents/Produktbلاetter/Produktblatt_SealLion.pdf
- [60] Data Transmiss. Equip. Operating in the 2,4 GHz ISM Band and Using Spread Spectr. Modulation Techn.. Accessed: Apr. 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300300_300399/300328/02.02.01_30/en_300328v020201v.pdf
- [61] Iperf3. Accessed: Apr. 2022. [Online]. Available: <https://software.es.net/iperf/>
- [62] C. Demichelis and P. Chimento, "IP packet delay variation metric for IP performance metrics (IPPM)," *Internet Request for Comments*, RFC Editor, RFC 3393, Nov. 2002.
- [63] T. N. Minhas, O. G. Lagunas, P. Arlos, and M. Fiedler, "Mobile video sensitivity to packet loss and packet delay variation in terms of QoE," in *Proc. 19th Int. Packet Video Workshop*, 2012, pp. 83–88.
- [64] V. Kebkal, K. Kebkal, O. Kebkal, and M. Komar, "Experimental results of delay-tolerant networking in underwater acoustic channel using S2C modems with embedded sandbox on-board," in *Proc. IEEE/MTS OCEANS Conf.*, 2015, pp. 1–6.
- [65] J.-Y. Le Boudec, *Performance Evaluation of Computer and Communication Systems*. Lausanne, Switzerland: EPFL Press, 2011.
- [66] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowitz, "Extensible authentication protocol (EAP), RFC 3748," *Internet Requests for Comments*, RFC Editor, RFC 3748, Jun. 2004, Accessed: Sep. 2022.



Emanuele Cocco received the master's degree in ICT for Internet and multimedia (cybersystem) from the University of Padova, Padova, Italy, in 2020.

From 2020 to 2022, he was a Researcher Engineer with the SIGNET Lab, University of Padova and participated in the RoboVaas project, integrating different networks and protocols in a single framework. He is currently a Research Engineer with EvoLogics GmbH, Berlin, Germany, focusing on protocol development and system integration.



Cosmin Delea received the master's degree in control theory from the Hamburg University of Technology, Hamburg, Germany, in 2018.

He is currently a Research Associate with the Fraunhofer Centre for Maritime Logistics and Services, Hamburg. His current work focuses on the development of control system for waterborne robotics (unmanned surface vessels, remotely operated vehicles, and autonomous underwater vehicles) and software systems for supporting their operation. His main research interests include mobile underactuated

robotics, linear control systems, navigation and control of waterborne robots, and underwater sensing.



Fabian Steinmetz received the B.Sc. and M.Sc. degrees in electrical engineering in 2015 and 2018, respectively, from the Hamburg University of Technology, Hamburg, Germany, where he is currently working toward the Ph.D. degree.

He is also a Research Assistant with the Hamburg University of Technology, where he is with the Institute for Autonomous Cyber-Physical System. From 2021 to 2022, he was with the Institute of Computer Science, University of Koblenz and Landau. Since 2022, he has been with the Institute for Autonomous

Cyber-Physical Systems, Hamburg University of Technology. Furthermore, he was involved in the MarTERA RoboVaaS project. His current research interests include acoustic underwater communication and localization for low-power applications, for example, underwater wireless sensor networks or micro autonomous underwater vehicles.

Mr. Steinmetz received the Best Paper Award at International Conference on Underwater Networks and Systems in 2018.



Roberto Francescon (Student Member, IEEE) received the M.Sc. degree in telecommunications engineering in 2015 from the University of Padova, Padova, Italy, where he is currently working toward the Ph.D. degree.

He joined many sea trials with the NATO STO CMRE, La Spezia, Italy, and EvoLogics GmbH, Berlin, Germany. From 2017 to October 2022, he was a Software Developer with Wireless and More srl, a spin-off company of the University of Padova. His research interests include the development of

software supporting and managing new communication protocols and schemes for wireless underwater and terrestrial networks.



Alberto Signori (Student Member, IEEE) received the B.Sc. degree in information engineering and the M.Sc. degree in telecommunication engineering from the University of Padova, Padua, Italy, in 2015 and 2017, respectively, and the Ph.D. degree in information engineering from the University of Padova under the supervision of Prof. M. Zorzi in 2021.

In January 2018, he joined the SIGNET Research Group, Department of Information Engineering, University of Padova, as a Research Engineer. His studies during the Ph.D. research include the design, analysis,

and implementation of novel communication protocols and architectures for underwater networks and network security for underwater communications. He has been involved in the software development of the DESERT Underwater framework. He collaborated with the MarTERA RoboVaaS project and with the European Defence Agency project SALSA.

Dr. Signori received the Best Student Paper Award at International Conference on Underwater Networks and Systems in 2019 and the Best Paper Award at IEEE INFOCOM International Workshop on Wireless Communications and Networking in Extreme Environments in 2020.



Ching Nok Au received the M.Sc. degree in information and communication systems from the Hamburg University of Technology, Hamburg, Germany, in 2020.

He is currently a Scientific Researcher with the Fraunhofer Center for Maritime Logistics and Service, Hamburg, Germany, where he focuses on the design and implementation of web-based solutions. He has accumulated full-stack experience in setting up microservices, network engineering, and handling web protocols according to varying throughput and

latency needs. He has been involved in the implementation of various projects such as MUH, RoboVaaS, SCIPPER, and RAPID, where he is responsible for web development, including mission planning of multiple unmanned vehicles as well as visualization of air quality and emission.



Filippo Campagnaro (Member, IEEE) received the B.Sc. degree in information engineering and the M.Sc. degree in telecommunication engineering, in 2012 and 2014, respectively, and the Ph.D. degree in information engineering in 2019, all from the University of Padova, Padova, Italy.

In 2014, he joined the SIGNET Group, Department of Information Engineering, University of Padova, as a Research Engineer. In October 2019, he joined the Department of Information Engineering, University of Padova, as a Postdoctoral Researcher and Lecturer of ICT master courses. He is currently a Research Assistant Professor with the Department of Information Engineering, University of Padova. His Ph.D. thesis involves the design and the evaluation of multimodal underwater optical and acoustic networks, with a particular focus on simulation and field experimentation (defense scheduled for February 2020). He has joined many sea trials with the NATO STO CMRE, La Spezia, Italy; EvoLogics GmbH, Berlin, Germany; ENEA, Rome, Italy; the IMDEA Network Institute, Madrid, Spain; and the University of Haifa, Haifa, Israel. He is the Technical Manager of the MarTERA RoboVaaS project and collaborates with the European Defence Agency project SALSA. Since 2017, he has been collaborating with Wireless and More srl, a spin-off company of the University of Padova. His current work involves software development for the DESERT Underwater framework as well as the design of novel communication stacks and architectures, with special attention to cross-layer and multitechnology designs. His research interests mainly include the design, analysis, implementation, and field evaluation of multimodal underwater networks.



Vincent Schneider received the bachelor's and master's degrees in biomimetics from the University of Applied Sciences, Bremen, Germany, in 2015 and 2017, respectively.

While focusing on computational fluid dynamic analysis of flying fish's gliding flight for wing in ground effect craft in his bachelor, he developed a fish-inspired fin propulsion system for a micro remotely operated vehicle using additive manufacturing. Within five years with the Fraunhofer Center for Maritime Logistics and Service (Fraunhofer CML),

Hamburg, Germany, he gained valuable experience in coordinating scientific projects such as RoboVaaS and strengthened his hardware developing skills by leading the design, integration, and validation of two custom-built ASVs. Within this work, he deepened his knowledge about additive manufacturing and was leading the installation of a laboratory for autonomous systems at Fraunhofer CML's new facilities and is currently the head of that laboratory. His focus is to spark innovative ideas and take the lead in research projects while improving innovation speed and reduce development cycles for prototype development with the initiated facilities.



Federico Favaro received the B.Sc. and M.Sc. degrees in telecommunication engineering from the University of Padova, Padova, Italy, in 2008 and 2011, respectively.

In 2012, he joined the SIGNET Group, Department of Information Engineering, University of Padova, as a Research Engineer on underwater networks. He is working as a network architect in the field of mobile networks. He is also consultant at SIGNET Group and one of the maintainers of the DESERT Underwater framework. His research interests include the implementation and testing of medium access control protocols and software interfaces between simulation software and real modems, as well as design and implementation of above-water wireless networks aimed to interface the underwater sensors networks and Internet of Things above-water world.



Johannes Oeffner received the B.Sc. degree in biology from the University of Münster, Münster, Germany, in 2007 and the M.Sc. degree in biomimetics/locomotion in fluids from the University of Applied Sciences Bremen, Bremen, Germany, in 2009.

He is currently leading the team “Maritime Technologies and Biomimetics” with the Fraunhofer Center for Maritime Logistics and Service (Fraunhofer CML), Hamburg, Germany. His team focuses on developing green technologies to increase energy efficiency, sustainability, and environmental impact

and on prototyping innovative biomimetic solutions and robotic applications. Before joining the Fraunhofer CML, he was a Hydroacoustics Scientist with a research institute in New Zealand, participating in experimental research surveys and being involved in signal processing, data handling, and analysis. He has coauthored several scientific articles on biomimetics—i.e., on sensing and friction reduction—fluid dynamics, hydroacoustics, and autonomous navigation systems. He initiated and coordinated the EU-Project AIRCOAT with the aim to develop a biomimetic air lubricating hull coating that reduce hull friction to reduce energy use and emissions of ships.



Bernd-Christian Renner received the Diploma and Ph.D. degrees in computer science and engineering from the Hamburg University of Technology, Hamburg, Germany, in 2008 and 2013, respectively.

He is currently a Full Professor and Head of the Institute for Autonomous Cyber-Physical Systems, Hamburg University of Technology. From 2020 to 2022, he was an Associate Professor with the University of Koblenz and Landau, Mainz, Germany. From 2016 to 2020, he was an Assistant Professor with the Hamburg University of Technology. From

2012 to 2016, he was a Postdoctoral Researcher with the University of Lübeck, Lübeck, Germany. He has authored or coauthored articles in several international journals and conferences and reviewed for and served on several technical program committees. He has been serving the International Workshop on Energy Harvesting and Energy-Neutral Systems at ACM Conference on Embedded Networked Sensor Systems in several roles since 2014, and he is a member of its steering committee. His research interests include applications of networked low-power sensing and cyber-physical systems running on energy harvested from the environment. His research interests also include networking protocols for networks of embedded sensor, active and passive backscatter acoustic communication, localization, and navigation in mobile low-power underwater networks.



Michele Zorzi (Fellow, IEEE) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padova, Padova, Italy, in 1990 and 1994, respectively.

During the academic year 1992–1993, he was on leave with the University of California at San Diego (UCSD), La Jolla, CA, USA. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy. After spending three years with the Center for Wireless Communications, UCSD, in 1998, he joined the

School of Engineering, University of Ferrara, Ferrara, Italy, where he became a Professor in 2000. Since November 2003, he has been with the Department of Information Engineering, University of Padova. His current research interests include performance evaluation in mobile communications systems, wireless sensor networks, Internet of Things, cognitive communications and networking, 5G millimeter-wave cellular systems, vehicular networks, and underwater communications and networks.

Dr. Zorzi received several awards from the IEEE Communications Society, including the Best Tutorial Paper Award in 2008 and 2019, the Education Award in 2016, the Stephen O. Rice Best Paper Award in 2018, and the Joseph LoCicero Award for Exemplary Service to Publications in 2020. He was the Editor-in-Chief for *IEEE Wireless Communications Magazine* from 2003 to 2005, *IEEE TRANSACTIONS ON COMMUNICATIONS* from 2008 to 2011, and *IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING* from 2014 to 2018. He has served the IEEE Communications Society as a Member-at-Large of the Board of Governors from 2009 to 2011 and from 2021 to 2023, as the Director of Education from 2014 to 2015, and as the Director of Journals from 2020 to 2021.