E-Navigation: a Distributed Decision Support System with Extended Reality for Bridge and Ashore Seafarers

Pietro Cassará[1] *IEEE Member*, Maria Di Summa[2], Alberto Gotta[1] *IEEE Member*, Michele Martelli[3],

***Abstract*—A distributed decision support system has been developed to assist seafarers during several navigation tasks, for instance, in avoiding a collision with a detected obstacle in the sea and envisioning a future autonomous navigation system. In this paper, the decision support system is based on the results of a customized simulation model representing the ship's behavior, including hydrodynamics, propulsion, and control effects. Sensors monitor and collect the parameters of the environment and the ship onboard. The telemetry and the calculated route are visualized on a wearable visor exploiting augmented reality. Such context information is also replicated ashore through a narrow-band satellite link using an IoT publish-subscribe communication paradigm to allow one or more remote seafarers to supervise the situation in a virtual reality environment. Overall, the potential of the proposed system is presented and discussed for application in the context of autonomous navigation.**

***Index Terms*—Autonomous Ship; Extended Reality; Situation Awareness; Maritime IoT-based Mobile Communications; Decision Support System; Digital Twin; Virtual Bridge.**

I. INTRODUCTION

Recent years are witnessing the advent of Remotely Piloted Systems in aerial and terrestrial. Maritime applications and the prototypes of Unmanned Autonomous Vehicles leverage novel learning and control techniques [1]–[3]. Since the maritime industry is showing increasing interest in autonomous surface vessels (ASVs), the Maritime Safety Committee (MSC) of the International Maritime Organization (IMO) approved June 2019 interim guidelines for Maritime Autonomous Surface Ships (MASS) trials. These guidelines identify four levels of autonomy degree (D_x), which are:

- D_1: ships with automated processes and decision support, which includes the automation of some unsupervised operations but with a seafarer ready to take control;
- D_2: remotely controlled ships with seafarers on board, where the ships are operated from a remote location, but the seafarers can be available on board to take control;
- D_3: Remotely controlled ship without seafarers on board; hence, the ship is controlled and operated from another location;
- D_4: fully autonomous ship controlled by an operative system.

The current trend of maritime new buildings is a digitalization of the majority of the systems installed onboard, in particular for the navigation decision support tool. Promising technologies are Augmented Reality (AR) and Virtual Reality (VR), especially in D_2 and D_3 for autonomous ships. The development of tools based on both AR and VR opens scientific challenges when dealing specifically with the sea environment with the need to investigate how ICT can be applied in maritime domains with the requirement of a massive amount of data transfer over oceanic distance. For the above-mentioned motivations, this work aims at integrating different enabling technologies to build a cyber-physical system of systems (CPSOS) supporting different levels of autonomy according to the specific needs of the reference scenario. The presence of VR and AR technologies allows the system to take advantage of the real and virtual worlds in a continuous exchange of information that allows for improving navigation safety.

This paper presents the studies and the performance analysis conducted to develop a "Virtual Bridge Viewer", which implements an Extended Reality (XR) to display information appropriately selected and processed to enrich the perceived reality on-board and, throughout a VR environment, ashore as well. The system was developed in the framework of the e-Navigation Project for developing a prototype of a simulation-based Decision Support System (DSS), according to the scenarios envisioned for D_1-D_3. The first use case addresses a D_1 scenario, where the end-user of the selected application is the bridge seafarer, and the information presented by the VR regards collision avoidance. The DSS is based on two modules: the former is devoted to identifying the target, while the latter will calculate the evasive course. The suggested route and some cruising parameters are shown on VR-googles on board. The second use case addresses D_2 and looks forward to D_3 scenarios that, leveraging a scalable communication framework, allow for mirroring the cruising parameters on multiple remote screens exploiting VR techniques. We also propose an IoT communication framework for allowing the bidirectional exchange of information and commands between

---

This study is part of the Project e-Navigation, a Research and Innovation project coordinated by Fincantieri S.p.A. with the participation of the National Research Council (CNR); the project receives grants from the Italian Ministry of Infrastructures and Transport (MIT). Part of the work addressed in this paper is under the National Patent n. 102020000010105 of 06/05/2020.

[1] Pietro Cassará and Alberto Gotta are with the ISTI, CNR of Pisa, Italy, and with CNIT Parma Italy, e-mail: pietro.cassara@isti.cnr.it, alberto.gotta@isti.cnr.it.

[2] Maria Di Summa is with the STIIMA, CNR of Bari, Italy, e-mail: maria.disumma@stiima.cnr.it.

[3] Michele Martelli is with the DITEN, University of Genova, Italy, e-mail: michele.martelli@unige.it).

ships and the relative remote control operators. Several issues arise in complex scenarios, such as those addressed in this paper, including the dependability of concurrent remote control systems that interact simultaneously. Therefore, this work limits the investigation of a system constrained to a single ship and a single remote control center, looking at the building blocks of a system that integrates: the ship simulator, which acts as a digital twin of the real ship, the communication paradigm, which enables the onboard automation as well as the remote supervision, and the Human Machine Interfaces (HMIs) to offer to the bridge seafarer for safe navigation as well as the VR environment to remote ones. A realistic navigation scenario has been simulated to test the DSS system. In contrast, the ship's and the shore's communication has been tested through real traffic data via an emulated satellite channel. The proposed DSS integrates state-of-the-art simulation techniques, communications, and virtual reality to obtain a new novel CPSoS for safer navigation.

II. RELATED WORK

In recent years, many studies highlighted the advantages deriving from navigation through the three-dimensional visualization of cartography [4]. The scientific community has shown particular interest in XR applications in the maritime sector [5] by overlaying AR information on either real or VR environments [6]. Grabowski et al. [7] have also investigated the new challenges relative to the impact of such new technologies in the maritime field and which contribution the XR may provide in safety-critical systems [8].

Interesting insights for designing communication and computing infrastructures underlying autonomous systems can be found in the literature concerning autonomous cars and aerial systems. Regarding autonomous cars, the realization of full-autonomous driving boosted research, showing that autonomous driving requires massive data from other vehicles and roadside units (RSUs) or ground control stations (GCS). This use of massive data puts forward new vehicle-infrastructure collaborative autonomous driving networking requirements such as sensing-communication-computing infrastructure with low latency and high reliability. In this case, the Edge framework can give the advantage of reducing the latency for the communication between the end user and the server that provides information and functionalities for the autonomous vehicle. In this scenario, authors in [9] propose a dynamic bandwidth adjustment and Time-Sensitive-Network(TSN) channel division method based on a 5G+TSN network and applies this method to the cloud collaborative vehicle networking system. The system aims to achieve the L3 level and above of autonomous vehicle networking and promote the system's development, deployment, and operation in multiple scenarios.

The authors in [10] address the issues of reliable, efficient, and low latency infrastructure exploiting the advantages of Software Define Networking (SDN), fog computing, and 5G mobile communication technologies. The authors present a new architecture for 5G-VANETs based on cloud computing, fog framework, and SDN. Using the SDN, the authors separate the control plane from the data plane to optimize the computing load and reduce the latency. Further, the authors split the control plane into two sub-layers: temporarily in the fog and permanently in the cloud. They install the local SDN controllers inside the fog cell head nodes (fog head vehicles) to make local decisions more rapid and efficient. For resiliency purposes, the authors also propose to elect a backup local SDN controller in the dynamic fog in case of dysfunction.

In [11] authors address the problem of supporting the autonomous vehicle infrastructure for generating massive data and exploiting the Artificial Intelligence (AI) algorithm underlying the control policies. For this reason, the authors propose C-Continuum, a Computing Continuum framework targeting distributed AI in UAV scenarios. C-Continuum aims to define a new generation of tools and mechanisms to enable fine-granularity computation, coordination, and mobility management across the mobile-computing spectrum from the edge to the core. C- Continuum embraces Named Data Networking (NDN), making a case for naming any computational entity and using those names for resource location, data transfers, and computing functions. The aim of developing a computational environment that is distributed, edge-oriented, and federated is also introduced in the Teaching Project [12], [13], which allows for the smooth integration of various heterogeneous resources such as specialized edge devices, general-purpose nodes, and cloud resources to efficiently run AI, cybersecurity, and dependability components of autonomous driving applications.

New communication and computing infrastructures capable of supporting low-latency massive traffic are the basis of the new multi-ship and cooperative collision avoidance system, as the system proposed in [14] that uses machine learning techniques. Again, Martelli et al. introduced the Internet of Ships (IoS) in [15] by hinting at a distributed computing platform to provide automatic control for maritime services.

In [16], the IoS concept has been expanded, looking at the shipping digitalisation process with the advent of a wholly autonomous and, at the same time, safe and reliable ship. The authors sustain that full autonomy could be obtained by two linked AI systems representing the ship navigator and the ship engineer that possess sensing and analysis skills, situational awareness, planning, and control capabilities. The exchange of information, models, controls, etc., needs an overlay network that supports it. In [17] Song et al. show the development and implementation of the Internet of Maritime Things (IoMT) platform to support long-range and high-rate communication for remote and online marine water quality monitoring. The proposed platform allows it to reach up to 10 Mbps but with a maximum coverage limited to 25 Km, which is remarkable for sensing applications but does not apply to the IoS context. Yet the proposed IoMT implementation is designed to support direct long-range communications between sensors and data collectors (gateways), which is suitable for sensing operations. Bacco et al. investigated the usage of satellite terminals with random access (RA) media [18]–[20] for Machine to Machine (M2M) and Internet of Things (IoT) applications. Networking with multi-service GEO satellites [21] imposes a fine-tuning for elastic traffic like short TCP connections or TFRC ones as studied and tested in [22] and [23]. In these papers, authors assume to adopt RA techniques over satellite, which have been proven more suitable with M2M/IoT traffic [24]. Cassará et al. provided a framework to evaluate the performance of RA with interference cancellation [25] to improve the efficiency of IoT traffic in satellite-maritime scenarios with global coverage. Recently, Luglio et al. provided a preliminary study [26] on the under-development VHF Data Exchange System (VDES) standard for both messaging capabilities and system flexibility via satellite for application messages concerning Automatic Identification System (AIS) and Application-Specific Messages (ASMs). In that paper, the authors reviewed the main characteristics of VDES in terms of channel compositions, supported rates, access schemes and latency, showing limited scalability and very low throughput compared to other Slotted Aloha [25] or Spread Aloha [27] techniques. This work tested a full IoT communication stack ranging from the satellite access system to the IoT overlay network. We also consider a network infrastructure with application proxies, data ingestion brokers, and bridges loaded by the real traffic generated by an accurate DSS simulator that mimics a collision avoidance maneuver of a ship, including all data generated by the ship's sensors. The telemetry flow is consumed onboard through a local ingestion broker by an AR visor and ashore by a VR application. The VR application runs a virtual reality engine to play the ship's maneuvers and another AR visor that shows the same situation as being onboard. The idea presented in this paper is a fluid interaction between the real world and the digital world so that it is possible to experience what is put into practice in the other and vice versa. For us, this is the metaverse concept, which Facebook has announced [28]. However, the metaverse does not yet have a univocal definition. In the maritime field, the metaverse represents the possibility of planning, monitoring, controlling, and managing highly complex situations that can occur at sea without being physically present on the ship. The maritime field is historically the most skeptical one and has more significant reserves in the adoption of new technologies, yet the metaverse tools are beginning to find their place in this sector as well, as highlighted in [29]

According to the literature review, to the best of our knowledge, we can assert that developing a CPSoS that integrates different enabling technologies to build a versatile and scalable decision support tool for autonomous maritime navigation hereafter proposed is the first example of such a kind.

III. SYSTEM ARCHITECTURE

This section describes the three building blocks that characterize the development of the E-Navigation platform. These blocks are the *ship simulator*, which acts as the digital twin of a real ship, *the communication infrastructure*, which provides the distributed fashion of the DSS, and *the virtual bridge* that implements the human-centric interface. The ship simulator provides the time series of the main variables needed to control the ship (i.e. speed, course, shaft revolution, etc.). The collision detection and avoidance modules work in real-time inside the ship simulator. The suggested outcomes of the ship simulator are delivered to the HMI systems to visualize the ship motions in a 3D environment. It is possible to customize the scenario regarding the manoeuvre, the main elements (fixed and moving obstacles), and the geographical information, i.e. the bathymetry and weather conditions. The virtual bridge is a hardware device, such as a screen. It receives the information from both the scenario and the ship simulator. After processing this information, it visualizes the information used to perform an evasive manoeuvre, working as a tool to support the user in taking decisions to enhance his situational awareness. All the data are managed and transmitted by the communication infrastructure. The three modules are explained in detail in the following sections.The three blocks interact with each other by means of the scenario builder as depicted in Figure 1.

TABLE I
SIMULATOR SUB-SYSTEM

MACRO - SYSTEMS	SUB-SYSTEMS
Manoeuvrability	Hull Rudder Appendages Propeller
Propulsion Plant	Main Engine Gearbox Shaft line
Control & Guidance System	Propulsion Control System Collision Detection Collision Avoidance

A. Ship Simulator

Time-domain simulation represents one of the most valuable techniques to predict the dynamic behaviour of a ship system by using a virtual environment [30]. Dynamic simulation allows for investigating the transient behaviour of the ship during manoeuvres at the design stage, i.e., without the need to have the real ship available. In addition, it allows for testing and optimising several ship parameters and developing control and decision support systems with a high grade of reliability [31]. Therefore, to study the proposed DSS's effectiveness, a multi-domain simulation platform has been developed that represents the dynamic behaviour of a ship and the communications infrastructure in the time domain. The simulator is a software platform including all the ship macro systems linked together to adequately describe the global ship behaviour: the ship manoeuvrability, the propulsion plant, and the control system. Each macro system has different elements, each schematized and modelled using the differential equations that govern their physical behaviour and represent their functions [32].

The simplest way to schematize a system is by using a table of parameters and algebraic equations that identify the system behaviour in steady-state conditions (i.e. propeller). On the contrary, detailed analysis requires computational efforts that do not fit the real-time requirement. The complex and more realistic approach used for most elements is to model

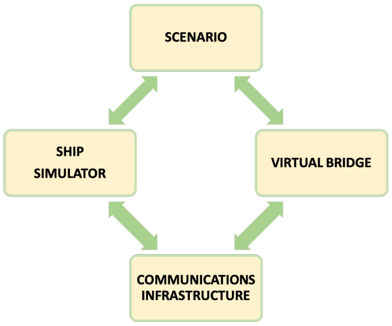


Fig. 1. System Layout.

the system with its physical equations, both algebraic and differential (i.e. ship hull force, control system, shaft line dynamics). In pursuing this, have been considered a large set of parameters increases the model complexity. This approach was validated during ship sea trials, as reported in [33]. Moreover, real hardware or software can be included in the simulation loop, i.e. the ”virtual bridge”. These four elements together create the possibility to enhance situation awareness during ship operations. The peculiarity of this work is a system engineering approach to bringing together all disciplines involved to represent a unified view of the system.

B. Communication Infrastructure

In this section, we describe the logical architecture of communication infrastructure based on both the paradigms of Internet of Things (IoT) [17], [34], and publish/subscribe (PUB/SUB) [35], [36]. The proposed infrastructure supports the virtual bridge service, and the relative data flows between the vessels' Command and Control (C2) systems and the ashore control station. Figure 2 shows a scheme of the logical architecture of the PUB/SUB communication infrastructure: such a paradigm allows for independent operation of the nodes that produce data from those that consume them, called publishers and subscribers [35]. Such an asynchronous communication mechanism between publishers and subscribers is implemented through the *broker* entity, which allows for receiving and storing the message queue containing the measurements sent by publishers and dispatching them to subscribers listening for the desired *topics*. In addition, the PUB/SUB paradigm also allows for decoupling the timings between message producing and consuming, respectively, and, in particular, according to the relative access bandwidth. Finally, this paradigm encapsulates heterogeneous types of information, such as data from high-level applications, sensors or actuators, or data from other logical entities within the same communication infrastructure. Figure 2 shows the high-level description of the proposed communication architecture. Data generated onboard related to C2 functionalities (e.g. engines, rudders, navigation systems, etc...) are sent to the Publisher Proxy Module (PPM). The PPM is in charge of encapsulating the data into IoT-compliant messages of the adopted communication protocol and then sending them to the broker. Differently from a direct IoT communication as in [17], the PPM has been designed to multiplex the payloads of the measurements and frame them into the messages of the adopted communication protocol, thus achieving higher efficiency, being able to fill the payload of an entire application packet. According to the topic's label, the broker forwards the messages to the topic's subscribers. In the PUB/SUB paradigm, the topic is a string of alphanumeric characters used by the publisher and subscriber at message publication and registration, respectively. Instead, the Subscriber Proxy Module (SPM) receives the messages from the broker, extracts the data, and forwards it to the actual recipients (e.g., the ashore stations). PPM and SPM are the software interfaces that convert raw sensor payloads into IoT-compliant messages and vice-versa into application data, respectively.

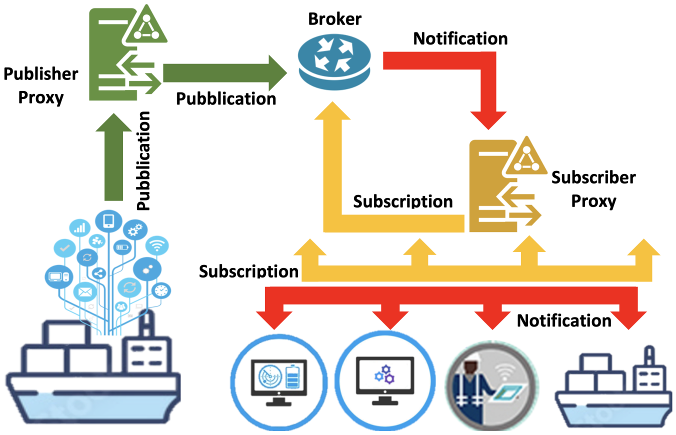


Fig. 2. Publish/subscribe communication paradigm for IoS.

Figure 3 shows a detailed scheme of the reference communication architecture: we can identify two macro communication entities, vessels and the Ashore Command and Control (ACC) station (red dashed line), connected by a bidirectional satellite link, to provide reliable global coverage. Note that the blue dashed blue lines are the virtual counterpart of the vessels available at the shore station. The medium access model has been implemented according to what is presented in the following papers [20], [25], [37].

The proposed architecture foresees that the data generated by the publishers, such as sensors, virtual bridges, or other service entities, are sent to the PPM. The use of such a module provides twice the advantages: all the entities, services, or systems involved in the communications may not necessarily use a standard communication protocol. We can interface these devices through a standard protocol through the PPM, minimizing hardware/software modifications. The PPM can be used as an aggregator for data packets generated by the devices. These packets frequently hold limited size (e.g. a few bytes), so their transmission is inefficient for channel utilization, mainly because of the overhead due to the protocol stack. By exploiting an appropriate data aggregation function, this issue can be reduced.

The proposed architecture in Figure 3 allows communication between vessel entities or toward one or more remote ashore centers in the simplest way possible to avoid any increase in delays or transmission errors. The information

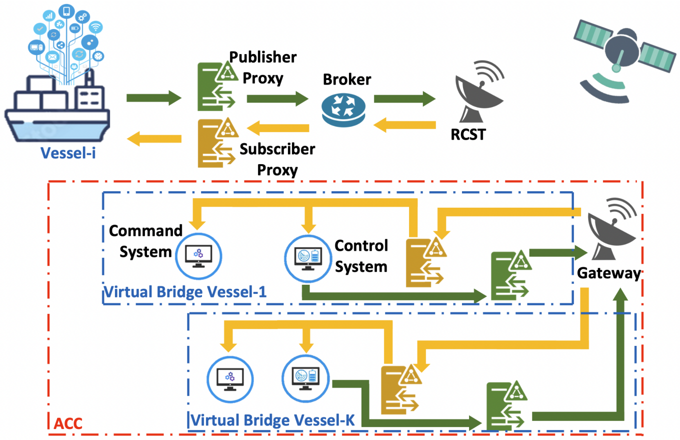


Fig. 3. Communication scenario: Vessel i allocates the *topics* to message with an ACC ashore, which manages K Virtual Bridges of the relative vessels $(1, \ldots, i, \ldots, K)$.

generated by an entity or service of the vessel is received by the PPM, encapsulated in suitable messages, and sent to the broker responsible for forwarding the message to the destinations that can still be the vessel itself, one or more ACCs, or both. The SPM at the destination was introduced for the same reasons as the PPMs, i.e., to facilitate interfacing with modules interested in using information generated by a vessel, such as a control system or the virtual bridge service. The ACC manages out of K Virtual Bridges of the relative vessels that have subscribed to the topics to exchange the control messages.

PPMs and SPMs give a high degree of reconfigurability to our communication architecture since they allow the integration among source-sensor entities that may be heterogeneous regarding communication protocols. The interaction between the ship's digital twin and the HMI of the virtual bridge imposes a broker to be located on board the vessel, as shown in Figure 3, to implement a reliable DSS. Each fleet vessel must have a specific broker module to minimize the latency between the communication entities and services, such as the virtual bridge, which needs the typical constraints of real-time systems.

The performance of the proposed infrastructure can be improved by introducing a local broker, as shown in Figure 4. The local broker can provide advantages when we address the scenario in which a fleet of vessels equipped with a broker has to communicate with a shore station; hence at the station, SPMs have to be available for each service or application sending the data. Consequently, each broker-subscriber pair establishes a satellite connection through the gateway, with latency and packet loss specific to the adopted physical layer technology. This high number of connections is generated using the most common communication protocols on which pub/sub paradigms are based, such as TCP. Thus, specific fault-tolerance techniques need to be provided for each subscriber module that at least guarantees the success of the transmitted data even when the connection fails. In addition, the amount of load on the channel can increase considerably with the number of subscribers because protocols such as TCP can involve a lot of acknowledgment messages. Using the local broker can limit the load and reliability issues mentioned above. Implementing such an infrastructure does not require any publisher on the shipboard side of any changes. At the same time, the shore station subscribers will have to establish a session with the local broker and no longer with the gateway. In doing so, the local broker must interface with the satellite gateway and handle any disconnections with one or more remote brokers. In addition, the only TCP acknowledgments transmitted over the satellite channel and intended for the remote brokers will be those generated by the local broker, thus ensuring better channel utilization.

A second potential advantage of using a local broker is reducing subscribers' response time. Indeed, the subscriber's attempt to connect to the broker requires at least three Round-Trip-Times (RTT): one for establishing the connection through TCP, one for creating the Pub/Sub session, and one for sending the subscribing message. Hence, in the case of geostationary satellites, we start to receive useful data after at least 1.5 seconds. Instead, we can offer caching capabilities for data messages using the local broker, allowing a new subscriber to receive them once the connection ends. In addition, the subscriber communicating with the broker and no longer with the satellite gateway would reduce its waiting time to receive data. Implementing the caching service on the remote broker also increases the infrastructure's reliability. In fact, in case of a temporary disconnection with the local broker, the remote broker can forward the whole data stream to the latter once the connection is re-established, making the entire operation transparent to subscribers and publishers.

The communication efficiency in terms of the delivery time of IoT messages through random access (RA) satellite links was introduced in [38]: Bacco et al. achieved that a PUB/SUB communication architecture can be beneficial in satellite-based architectures. They compared the two most used protocols for IoT, i.e., CoAP and MQTT, and experimented that CoAP was more efficient than MQTT with a low volume of data. However, CoAP didn't natively offer congestion control (CC) mechanisms to face higher data rates or user density, w.r.t. MQTT. Thus the latter was more flexible for operating in different traffic conditions. Hence, Bacco et al. addressed this issue by developing the CC feature, namely TFRC-s, for CoAP to offer the same facilities and general use as MQTT with the underlying TCP in [20]. In addition, they validate the performance of the CC protocol by deriving an analytical framework that models the PUB/SUB protocol's working point versus the satellite link's loss rate and the traffic load G_n, expressed by the number n of users attempting to channel in a given number M of time slot per datalink superframe.

The effective throughput in case of congestion loss only or also with fading is derived in [20] in terms of successful transmission probability p_{tx} given that a message is ready to be transmitted as:

$$p_{tx} = \Gamma^{-1}_{N_{rep}}(p)/G_n, \tag{1}$$

where $\Gamma_{N_{rep}}$ is the empirical model that accounts for the datalink losses, the interference cancellation mitigation tech-

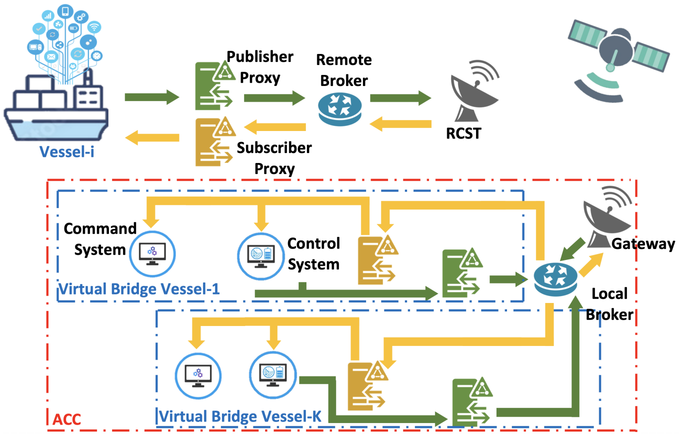


Fig. 4. Communication scenario multi-broker.

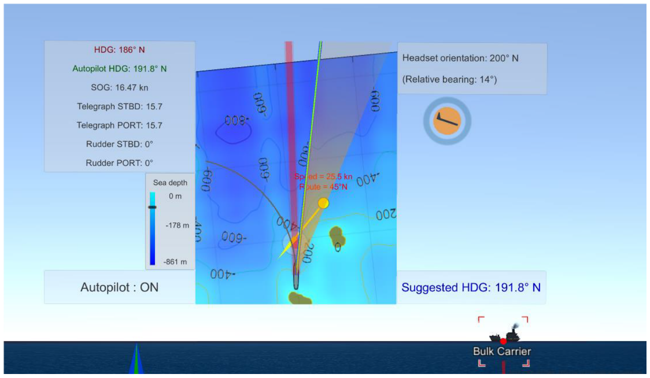

HDG: 186° N
Autopilot HDG: 191.8° N
SOG: 16.47 kn
Telegraph STBD: 15.7
Telegraph PORT: 15.7
Rudder STBD: 0°
Rudder PORT: 0°
Headset orientation: 200° N
(Relative bearing: 14°)
Speed = 25.5 kn
Route = 45°N
Autopilot : ON
Suggested HDG: 191.8° N

Fig. 5. Virtual Bridge Viewer.

nique applied on RA satellite links, and the loss probability p of datalink frame due to fading. It must be noted that such a model accounts for the datalink layer mitigation technique and the retransmission mechanism implemented at the transport layer by TFRC-s or TCP for CoAP and MQTT, respectively. Therefore, (1) provides the effective throughput of PUB/SUB message delivery through a satellite RA link.

However, other issues still arise in establishing the broker-broker communication channel. Firstly, many publishing/subscribing-based protocols do not provide broker-broker communication as a standard. Therefore, it is necessary to rely on tools that allow for ad-hoc solutions, which leads to the disadvantage of having an infrastructure component that is not supported by standards and, therefore, cannot necessarily be easily interfaced with commercial systems. Then, the local broker node acts as a centralized entity for the remote brokers. So it represents a single point of failure on which the system's proper functioning depends. Therefore, it is necessary to ensure a degree of resilience to failures. Geographical redundancy is the commonly adopted solution, but this involves managing the synchronization and switching of redundant units.

C. Virtual Bridge

Virtual Bridge Viewer and Ashore Control Centre are the two XR applications designed for the E-navigation system. The former is the application that supports the seafarer on board, and the latter is usable inside the ashore station that monitors the flow of ships in one or more specific areas. We developed the two applications for Microsoft HoloLens and a desktop workstation, but they could be installed on other devices that support the development environment Unity 3D [39]. The main software development platform is Unity, with the integrated development environment Microsoft Visual Studio, which must also be used for installation on HoloLens. The development language is $C\#$. The programming interface of HoloLens is based on the UWP platform (Universal Windows Platform), supported by Windows 10. During the development, packages for Unity were used and freely available on the net. The other used packages are available as open-source software on the GitHub platform. The choice of display supports is aimed at offering greater comfort to operators during their task execution. The two applications are very similar even if they have specific peculiarities to support the seafarers in their particular and different tasks. These applications display navigation information and indications of possible obstacles overlaid onto the real view and assist in evasive maneuvers to avoid collisions. The digital data is rendered in textual, graphic, or both formats, depending on the most effective way so that the operator can receive the information more quickly and comprehensively. The ACC application presents a video stream of the view of the operator wearing the HoloLens in a room inside the ship without windows for tactical reasons, i.e., in the absence of a direct line of sight with the external surrounding seascape. So, the ACC operator can perceive the situation on the outer bridge. AR applications represent the front end of information directly coming from AIS, ECDIS, and ARPA onboard systems and a simulator developed in the Marine Engineering Department of the University of Genoa laboratory. Based on the data collected by the onboard systems, the simulator detects a possible danger of collision. It always processes the data foreseeing a possible evasive maneuver displayed in AR. The view of an operator using the Virtual Bridge Viewer is shown in Figures 5, 6, and the ACC in Figure 7, respectively. In Figure 6 shows the application features intended for the operator on board: the environment has been synthetically reconstructed in VR since there were no clips of the application tested during a real cruise.

Figure 5 reports the information visible to the onboard operator when he is wearing the Hololens. The left panel shows:

- information relating to heading, speed (SOG), left and right telegraph notch, and left and right rudder angle;
- textual information relating to the activation of the autopilot;
- the depth at the ship's position is displayed as an indicator and numerically.

On the central panel:

- bathymetric profile: represents the depth of the seabed in an area that is normally a few kilometers;
- current route (red line), a red band around the current route highlights the tracking error (30m);

- autopilot route (green line);
- trajectory corresponding to the tactical diameter (grey curve)
- orientation of the viewer (orange), shows the direction of observation with respect to the ship. Right panel:
- orientation of the viewer: indicates the direction of observation both absolute (with respect to the North) and with respect to the ship;
- graphic representation of the wind: the symbolic display of the intensity of the wind, according to the Beaufort scale. The wind direction is represented by the orientation of the symbol with an angle equal to the direction received, referring to the bathymetry.

If there are obstacles, other data are displayed.
On the right panel:

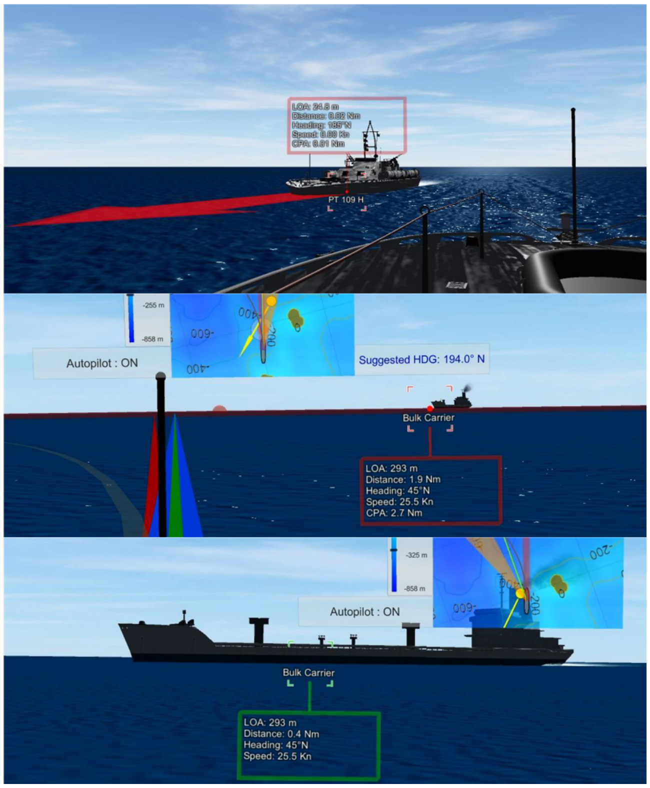
- recommended route (blue line).

On the central panel:

- position and speed of the obstacle in textual and iconic form (yellow circle positioned on the bathymetry with an arrow indicating the direction of movement, the length of which is proportional to the speed);
- Closest Point of Approach (CPA) indicated as a red circle and positioned on the bathymetry;
- superimposed, an indicator with the available data is displayed in the direction of the real obstacle (this is actually an element disconnected from the HUD, see next paragraph).

In the event of nearby obstacles, the bathymetry display automatically switches to a larger scale factor, highlighting the distance circles every 50m, from 100 to 250m. In Figure 6 the displayed information is:
1. information on the obstacle: some data relating to the obstacle (when present) are displayed in a box that also indicates its position; in the event of a possible collision, the outlines of the box are red, otherwise they are green;
2. obstacle speed: an arrow originating in the position of the obstacle indicates the direction and qualitatively the speed;
3. CPA (Closest Point of Approach): a small red sphere indicates its position, while a semi-transparent flashing area indicates the radius of the area of possible collision;
4. various routes are displayed as lines of various colors, starting from the ship and continuing on the sea; the colors reflect the display on the HUD:

- red: current route;
- green: course set on autopilot;
- blue: suggested route;
- transparent grey: curve corresponding to the tactical diameter.


LOA: 24.8 m
Distance: 0.02 Nm
Heading: 185°N
Speed: 0.00 Kn
CPA: 0.01 Nm

Autopilot : ON
Suggested HDG: 194.0° N

LOA: 293 m
Distance: 1.9 Nm
Heading: 45°N
Speed: 25.5 Kn
CPA: 2.7 Nm

Autopilot : ON

LOA: 293 m
Distance: 0.4 Nm
Heading: 45°N
Speed: 25.5 Kn

Fig. 6. Viewer functionalities.

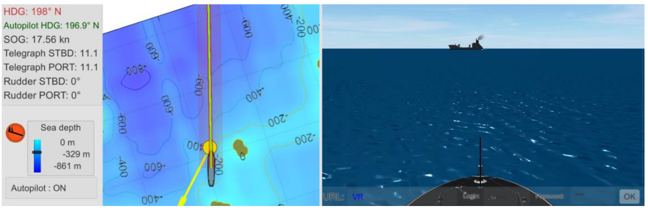

HDG: 198° N
Autopilot HDG: 196.9° N
SOG: 17.56 kn
Telegraph STBD: 11.1
Telegraph PORT: 11.1
Rudder STBD: 0°
Rudder PORT: 0°
Sea depth
0 m
-329 m
-861 m
Autopilot : ON

Fig. 7. Ashore Control Centre.

In Figure 7, the Ashore Control Center application shows the information processed from the data received from the ship (or from the simulator). The application presents almost the same information as the virtual Bridge Viewer except for some considered not helpful for ashore operators; furthermore, the augmented reality panel is not superimposed on the scene, and the view of this last is on the right of the screen. In addition to images, we can also use acoustic signals to alert the operator to events such as obstacles, the possibility of collision with other vessels, or the end of danger from previous events. The applications were made considering the type of ship they were intended to support with the maneuvering services. Augmented reality applications, therefore, offer support for these operations by facilitating operators on board and ashore thanks to a customized selection of the most helpful information according to the procedures.

IV. SYSTEM PERFORMANCE

The application scenario selected to test the system involves a ship that serves as a multi-mission patrol vessel. Such a selected vessel has many advantages, including a considerable versatility determined by its Search and Rescue (SAR) operating profile; it is a ship designed for anti-fire, anti-pollution, recovery, and rescue operations. The tool will have the function of increasing the performance of the bridge operators through additional digital information.

A. *Manouver Evaluation*

The test case is a sequence of challenging maneuvers such as acceleration, zig-zag, and turning circle, mimicking

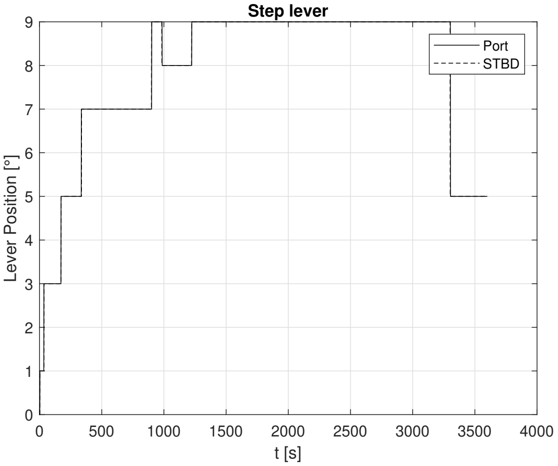


Fig. 8. Steplever position vs. time.

navigation in a crowded sea area. This test aims to test the responsiveness of the developed XR-based DSS in providing useful information for the bridge operator during challenging maneuvers. Furthermore, we used this test to evaluate the reliability of the ship simulator used to generate the parameters of the CPoSs composing a vessel.

The results are shown as time hystories hereinafter, the total duration of the simulation is one hour. Figure 8 reports the position of both starboard and port side steplevers, conventionally defined from -10 to 10, over time. The ship starting from rest is commanded to accelerate nearby the maximum speed, then the steplevers are kept constant, and then deceleration is required. Figure 9 hows the time history of the ordered rudder angle. To test the simulator capabilities, a 20/20 zig-zag maneuver is performed, followed by a sequence of hard turns to both port and starboard side, a turning circle, and at the end, a tight 35/35 zig-zag is performed. Figure 10 reports the ship's speed over time, in particular, it is worthy of notice the trend of the variable that evidences a drop (due to added resistance during turns) every time the rudder is actuated; this demonstrates the effectiveness of the modeling to catch the mutual interaction among the different dynamics. The heading is shown in Figure 11, where the 0 of the y-axis indicates the North. Eventually, the ship trajectory is shown in Figure 12, and it is reported in the non-dimensional form concerning the ship length (the waterline is, on purpose, out of scale for a better representation).

During the test, the operator is supported by an application running on the ship in carrying out the appropriate actions. The synthesis of the results can be displayed on a workstation remote control allowing monitoring of the operations from another point of view. As shown in Figure 13, a VR environment model has been developed.

B. Thgroughput Analysis

The simulations described in this section aim to test the reliability in terms of the transmission delay to prove that the developed infrastructure can provide real-time transmissions and resilience toward fault connections. To implement the

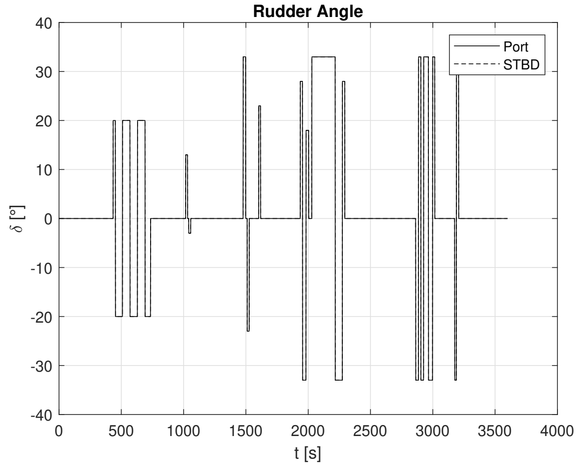


Fig. 9. Rudder angles vs. time.

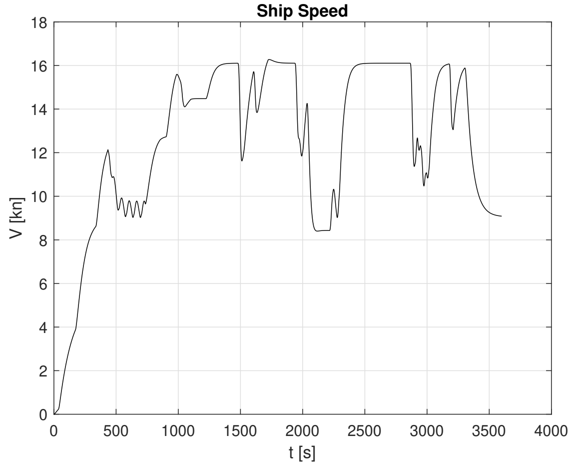


Fig. 10. Ship Speed vs. time.

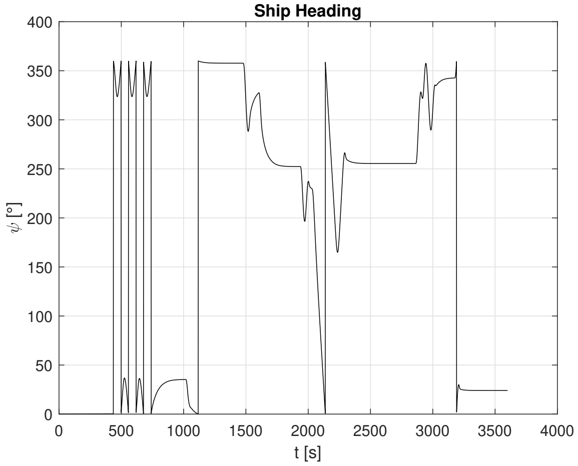


Fig. 11. Heading vs. time.

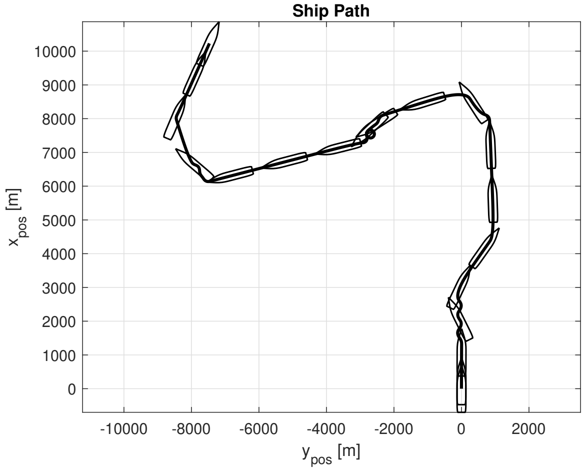


Fig. 12. Ship Trajectory vs. time.

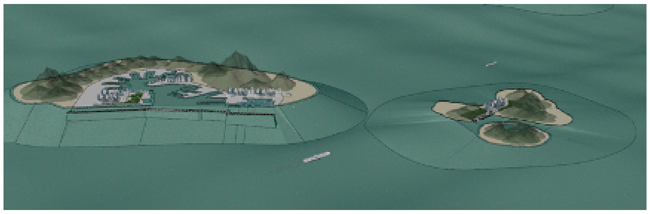


Fig. 13. Example of 3D scenario visualization.

communication infrastructure described in the previous section, we adopted as publishing/subscribing protocol MQTT, and precisely we used the Mosquitto libraries [40] [41]. Mosquitto also provides functionalities to support broker-broker communication, called MQTT bridge. In short, when one broker is configured as a bridge, it operates as a client by publishing and subscribing to topics on the second broker. At the same time, it will act as a broker for both the PPM and SPM modules. One of the advantages of bridging offered by Mosquitto is that system can be easily configured through a textual configuration file stored on the node operating as a bridge. Moreover, Mosquitto provides functionalities for the *Topic Remapping*. The broker acting as a bridge can change the topic prefix associated with received messages and the *Multi-broker bridging*. With this functionality, we can bridge to another broker in the event of disconnection by setting the $[address, port]$ pairs on the configuration file that identify the new brokers to be bridged.

The analysis conducted in the following aims to analyze the working point of the satellite system in terms of loss rate p, successful transmission probability p_{tx} and load G_n generated by the users using an ICN communication paradigm. This analysis is performed exploiting the formula of p_{tx} in equation 1 and the results shown in [18]–[20] about p as function of G_n, p_{tx}, $\Gamma^{-1}_{N_{rep}}$. Further, we referred to the results in [25], [42] for the analysis of the working point of a satellite system in a maritime scenario. The results, shown in Table II, represent the comparison between the transmission probability p_{tx} analytically evaluated and that obtained through Monte-Carlo simulations, by evaluating the working point of the system. The performance analysis is conducted by assuming an offered load G_n ranging from 50% to 100% of the total number of users $n = 64$ in the system. The results show that as G_n increases, p_{tx} (p) decreases (increases) due to the mutual interference generated by the users. The decrease of p_{tx} leads to a relative decrease in terms of global throughput. In the next section, we analyze also the performance of the satellite link in terms of the delay variance, which provides insights into the application-level performance.

TABLE II
WORKING POINT ANALYSIS

		G_n					
		50%	60%	70%	80%	90%	100%
Simulation	p_{tx}	0.856	0.764	0.672	0.604	0.539	0.494
	p	0.00114	0.00152	0.00194	0.00239	0.003	0.00358
Model	p_{tx}	0.899	0.81	0.688	0.608	0.546	0.497
	p	0.00165	0.00184	0.00221	0.00253	0.00289	0.00322

C. *On board Messaging*

This section is devoted to testing the message stream between the sensors deployed on the vessel, the MQTT broker, and the Virtual Bridge Visor. The 'Dynamic Propulsion and Manoeuvre Control Simulator' (SDCPM) generates the vessel control system data transmitted through the onboard broker. The simulator generates two types of quantities: time-driven and event-driven. The first category includes all quantities generated at regular intervals (1-4 sec). Conceptually, time-driven events are classified as the data associated with onboard telemetry, which is constantly updated and needs to be transmitted regularly to the entities, i.e., the onboard viewer and possibly the ground control center. Instead, the data generated following a temporally non-predictable 'event' belong to the event-driven category. Table III lists the quantities generated by the simulator and, for each of them, the type and the permissible range of values, all encoded with a float. The float data type corresponds to a 4-byte floating-point encoding according to the IEEE 754 standard [2]. On the other hand, the autopilot quantity is encoded by a single byte as an unsigned integer.

Regarding the time-driven category, it is assumed that all respective quantities are generated simultaneously and can therefore be encapsulated in a single message (more details below). In contrast, event-driven quantities may be generated and transmitted at different time instants. However, it is possible to group specific quantities and assume that those belonging to the same group are updated simultaneously.

The SDCPM was implemented using the Simulink platform and conceptually corresponds to the set of sensor modules allocated on board the ship. Once generated, the data must be transmitted to the Publisher Proxy module. For this purpose, a Simulink module was implemented to encapsulate and transmit the data. From the Simulink point of view, this transmission requires inter-process communication, which is realized using TCP/IP sockets. This way, interoperability between software modules is guaranteed, as no constraints

TABLE III
PARAMETERS FEATURES.

PARAMETER	TYPE	RANGE
Heading	time-driven	[0,360]Deg
GPS Speed	time-driven	[-13,18] kn
Autopilot	time-driven	[0,1]
Wind Speed	time-driven	[0.100] kn
Wind Dir	time-driven	[0,360]Deg
Rudder Angle	time-driven	[-35,35]Deg
Latitude	time-driven	[90S,90N]Deg
Longitude	time-driven	[180W,180E]Deg
Suggested Rudder Angle	event-driven	[-35,35]Deg
Obstacle Speed	event-driven	[0,30] kn
Obstacle Latitude	event-driven	[90S,90N]Deg
Obstacle Longitude	event-driven	[180W,180E]Deg
Recommended Position Telegraph	event-driven	[-10,+10] Lever

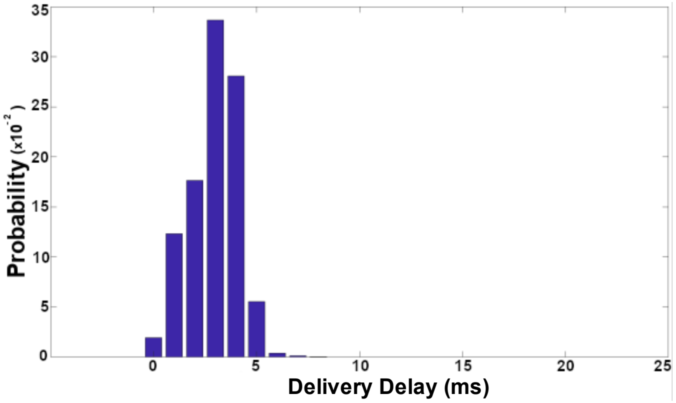


Fig. 14. Delay distribution of the messages exchanged between Publisher and Subscriber Proxies

are imposed on the operating system or programming language. As far as the transport protocol is concerned, there are essentially two possibilities: TCP and UDP. TCP guarantees reliability and orderliness in the delivery of messages. In a real scenario, it represents the preferred solution when it is impossible to assume the total absence of losses in the sensor-Publisher Proxy communication due to any damaged intermediate hardware components or anomalies linked to the physical transmission of information. Such losses may or may not be tolerated depending on the characteristics of the information transmitted. For example, we will likely send a value used for monitoring at periodic intervals. Therefore it is typically safe to assume that the loss of a single value is tolerable. The alternative is to use UDP, a connection-less protocol with limited overhead, even though it does not offer the same guarantees as TCP regarding reliability and sorting, for simulation purposes. UDP can represent an acceptable solution because the absence of logical connections makes the Publisher Proxy module even more ”transparent” and robust since it does not need to manage any disconnection problems typical of TCP. Moreover, it should be emphasized that the SDCPM-Publisher Proxy communication takes place locally (both modules are running on the same physical machine). Therefore, assuming a negligible probability of packet loss is reasonable, making TCP’s advanced functionalities unnecessary.

In this batch of tests, the main interest is to verify the resulting time distribution related to the implemented software modules. It is sufficient to consider the SDCPM as a data source for communication purposes.

1) Publisher Proxy: The main purpose of this module is to encapsulate in MQTT Publish messages the data received from the SDCPM/TrafficGen via TCP/IP sockets and forward them to the broker. This module is necessary if the sensor interfaces do not natively support the MQTT protocol. Simulink, the platform used to implement the SDCPM, does not provide any predefined MQTT module. The Publisher Proxy module is implemented in Java to minimize compatibility issues with the platform used for integration. The application’s configuration parameters are defined using a file encoded in XML. The control of the configuration file’s correctness is facilitated using the XSD scheme, which was specially defined to validate the XML document. In addition to the data required to establish the connection with the broker (IP addresses and port numbers), the file contains the ”SecureConnection” field, whose admissible values are ”false” and ”true”, which respectively indicate the use of TCP or TLS4 protocol to establish the connection with the broker.

2) Subscriber Proxy: This module receives the MQTT messages transmitted by the broker, extracts the navigation data, and forwards them to the Virtual Bridge module (and possibly other exciting modules). The main functionality is similar to that of the Publisher Proxy, i.e., offering an interface to modules that do not support MQTT. The module, implemented in Java, is configured using an XML file structured similarly to that described for the Publisher Proxy.

3) Numerical Results: The reference metric is the delivery delay, i.e. the time elapsed between the instant when the Publisher Proxy sends a single MQTT message and the instant when the Subscriber Proxy receives it. Conceptually, let us emulate a scenario where the publisher, subscriber, and broker nodes are physically allocated on board the vessel, and the subscriber corresponds to the Virtual Bridge Visor. As shown in Figure 14, most observations fall in the range [0-8] ms, relatively low values since the nodes involved in the communication are allocated within the same local network. We want to point out that we used timestamps with millisecond precision for the test. Hence, a delivery delay value of zero indicates a delay of less than one millisecond. Therefore, the presence of the MQTT broker does not introduce a considerable delay, thus confirming that the communication system can support real-time transmissions, such as the telemetry data flow destined for the Visor module.

D. Broker to Broker Messaging

This section tests the message stream between the MQTT broker installed on the vessel and the MQTT bridged broker on the shore. For convenience, in this testing phase, we developed the TrafficGen module, a simple traffic generator implemented in C++ which receives as input a text file containing a

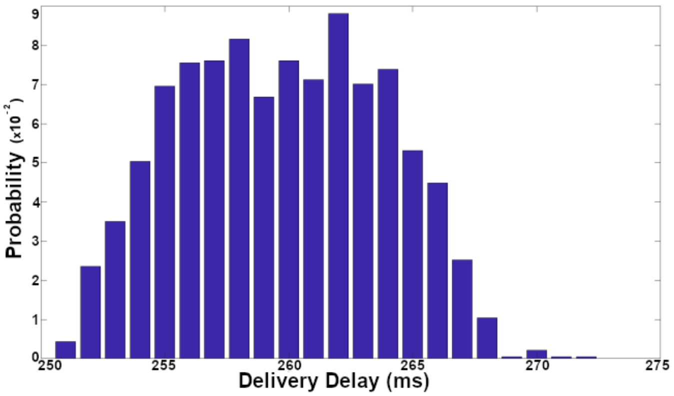


Fig. 15. Delay distribution of the messages exchanged between onboard broker and ashore broker with average traffic arrival of 1 message per second

sequence of inter-arrivals (measured in seconds) and which conceptually corresponds to the data packets generated by the onboard broker. This provides good flexibility in testing by varying the input series; this means we can analyze the behavior of the satellite communication subsystem with different traffic profiles.

1) Satellite channel emulator: Regarding the satellite channel emulation, we are interested in reproducing a high propagation delay and time-division access. These features are emulated through a *receiver buffer* by artificially delaying the processing of received packets for a time equal to the propagation delay of the satellite channel, corresponding to about 250ms. For time division access, we assume a setup of the satellite terminal as in [25] where a transmission can use a single time slot for each frame, apart from replicas for interference cancellation [25]. Considering a time frame of 13ms, a dedicated time division access implies that the throughput perceived by the receiver cannot be more than 1 packet/13ms. Thus, the implementation consists of forcing the receiver (the Subscriber Proxy) to read from the receive buffer at most one packet every 13ms.

2) Numerical Results: As mentioned earlier, we are interested in calculating the delivery delay to verify the correct emulation of the satellite channel and the transmission of messages from the onboard broker to the ashore one. As shown by the probability distributions depicted in Figure 15, most observations fall roughly within the 250-275ms range. This indicates that the delivery delay is never less than 250ms, as expected, since this value corresponds to the propagation delay set for the emulated satellite channel. The delivery delay depends not only on the propagation delay but also on the time-division access and the latency introduced by the broker to forward the MQTT message.

V. CONCLUSIONS

A multi-user decision support system responds to the fast-changing approach to navigation at sea. New systems are needed to face the autonomous navigation revolution, and AR and VR techniques are among the most promising. The system is fully scalable regarding the number of users and the level of autonomy of the ships in which it will be deployed. The study performed on the communication infrastructure proves the feasibility of exchanging data with a delay that doesn't affect the operations in terms of safety; this, of course, considers the ships' relatively slow dynamics. The recommendations for the near future are that the maritime community must agree on new standardized protocols and messages to be exchanged since what is currently in use is not suitable for autonomous navigation that needs much more information compared with manned navigation. Eventually, it is expected that the international legislative bodies release a clear regulatory framework, and the proposed architecture can be used as a test benchmark. Eventually, in future work, the testing in a controlled, but real scenario of the presented system will be an added value to verify the system in terms of external interference, lack of communications, and variable weather conditions. Moreover, is a matter of fact that the cyber-security aspects become crucial for a safe and resilient system, and this study will be part of future work.

ACKNOWLEDGMENTS

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (PNRR) Mission 4 Issue 2 Investment 1.4 ”Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S” CN00000023 ”Sustainable Mobility Center (CNMS)”

REFERENCES

[1] R. K. Barnhart, D. M. Marshall, and E. Shappee, *Introduction to unmanned aircraft systems*. Crc Press, 2021.

[2] V. Bertram, “Unmanned surface vehicles-a survey,” *Skibsteknisk Selskab, Copenhagen, Denmark*, vol. 1, pp. 1–14, 2008.

[3] P. Papadakis, “Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013.

[4] R. Goralski, C. Ray, and C. Gold, “Applications and benefits for the development of cartographic 3d visualization systems in support of maritime safety,” *TransNav-International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 5, no. 4, pp. 423–431, 2011.

[5] U. v. Lukas, M. Vahl, and B. Mesing, “Maritime applications of augmented reality–experiences and challenges,” in *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 2014, pp. 465–475.

[6] T. Butkiewicz, “Designing augmented reality marine navigation aids using virtual reality,” in *OCEANS 2017-Anchorage*. IEEE, 2017, pp. 1–9.

[7] M. Grabowski, “Research on wearable, immersive augmented reality (wiar) adoption in maritime navigation,” *The Journal of Navigation*, vol. 68, no. 3, pp. 453–464, 2015.

[8] M. Grabowski, A. Rowen, and J.-P. Rancy, “Evaluation of wearable immersive augmented reality technology in safety-critical systems,” *Safety science*, vol. 103, pp. 23–32, 2018.

[9] P. Ding, D. Liu, Y. Shen, H. Duan, and Q. Zheng, “Edge-to-cloud intelligent vehicle-infrastructure based on 5g time-sensitive network integration,” in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2022, pp. 1–5.

[10] A. Soua and S. Tohme, “Multi-level sdn with vehicles as fog computing infrastructures: A new integrated architecture for 5g-vanets,” in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2018, pp. 1–8.

[11] D. Aguiari, A. Ferlini, J. Cao, S. Guo, and G. Pau, “Poster abstract: C-continuum: Edge-to-cloud computing for distributed ai,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1053–1054.

[12] D. Bacciu, S. Akarmazyan, E. Armengaud, M. Bacco, G. Bravos, C. Calandra, E. Carlini, A. Carta, P. Cassarà, M. Coppola *et al.*, “Teaching-trustworthy autonomous cyber-physical applications through human-centred intelligence,” in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*. IEEE, 2021, pp. 1–6.

[13] V. De Caro, S. Bano, A. Machumilane, A. Gotta, P. Cassarà, A. Carta, R. Semola, C. Sardianos, C. Chronis, I. Varlamis *et al.*, "Ai-as-a-service toolkit for human-centered intelligence in autonomous driving," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2022, pp. 91–93.

[14] W. Hammedi, B. Brik, and S. M. Senouci, "Toward optimal mec-based collision avoidance system for cooperative inland vessels: A federated deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2022.

[15] M. Martelli, P. Cassarà, A. Virdis, and N. Tonellotto, "The internet of ships," *ERCIM News*, vol. 2020, no. 123, 2020.

[16] M. Martelli, A. Virdis, A. Gotta, P. Cassarà, and M. Di Summa, "An outlook on the future marine traffic management system for autonomous ships," *IEEE Access*, vol. 9, pp. 157 316–157 328, 2021.

[17] Y. Song, H. Shin, S. Koo, S. Baek, J. Seo, H. Kang, and Y. Kim, "Internet of maritime things platform for remote marine water quality monitoring," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[18] M. Bacco, P. Cassarà, E. Ferro, and A. Gotta, "Generalized encoding crdsa: Maximizing throughput in enhanced random access schemes for satellite," in *Personal Satellite Services*, R. Dhaou, A.-L. Beylot, M.-J. Montpetit, D. Lucani, and L. Mucchi, Eds. Cham: Springer International Publishing, 2013, pp. 115–122.

[19] M. Bacco, M. Colucci, and A. Gotta, "Application protocols enabling internet of remote things via random access satellite channels," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[20] M. Bacco, P. Cassarà, M. Colucci, and A. Gotta, "Modeling reliable m2m/iot traffic over random access satellite links in non-saturated conditions," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1042–1051, 2018.

[21] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "Networking with multi-service geo satellites: cross-layer approaches for bandwidth allocation," *International Journal of Satellite Communications and Networking*, vol. 24, no. 5, pp. 387–403, 2006.

[22] M. Bacco, A. Gotta, C. Roseti, and F. Zampognaro, "A study on tcp error recovery interaction with random access satellite schemes," in *7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*. IEEE, 2014, pp. 405–410.

[23] M. Bacco, T. De Cola, G. Giambene, and A. Gotta, "Tcp-based m2m traffic via random-access satellite links: Throughput estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 846–863, 2018.

[24] N. Celandroni, E. Ferro, and A. Gotta, "Ra and da satellite access schemes: a survey and some research results and challenges," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2670–2690, 2014.

[25] P. Cassará, A. Gotta, and T. de Cola, "A statistical framework for performance analysis of diversity framed slotted aloha with interference cancellation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4327–4337, 2020.

[26] M. Luglio, C. Roseti, and F. Zampognaro, "Vdes performance evaluation for future e-navigation services." in *ICETE (1)*, 2018, pp. 233–241.

[27] A. Baroni, A. Michel, M. Pannozzo, D. Zamberlan, M. Andrenacci, and F. Silvestri, "Performance assessment of the smart maritime satellite terminal for multimedia services and contents (saturn) system," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*. IEEE, 2014, pp. 136–143.

[28] S. Kraus, D. Kanbach, P. Krysta, M. Steinhoff, and N. Tomini, "Facebook and the creation of the metaverse: Radical business model innovation or incremental transformation?" *International Journal of Entrepreneurial Behaviour & Research*, vol. 28, 01 2022.

[29] E. Markopoulos, M. Luimula, C. Calbureanu-Popescu, P. Markopoulos, P. Ranttila, S. Laukkanen, N. Laivuori, W. Ravyse, J. Saarinen, and T. Nghia, "Neural network driven eye tracking metrics and data visualization in virtual reality maritime safety training," 09 2021.

[30] M. Martelli, "Marine propulsion simulation," in *Marine Propulsion Simulation*. De Gruyter Open Poland, 2015.

[31] M. Altosole, M. Figari, and M. Martelli, "Time domain simulation for marine propulsion applications," in *Proceedings of the 44th Summer Computer Simulation Conference*, 2012, pp. 8–11.

[32] M. Martelli, M. Viviani, M. Altosole, M. Figari, and S. Vignolo, "Numerical modelling of propulsion, control and ship motions in 6 degrees of freedom," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 228, no. 4, pp. 373–397, 2014.

[33] M. Martelli and M. Figari, "A design framework for combined marine propulsion control systems: From conceptualisation to sea trials validation," *Ocean Engineering*, vol. 254, p. 111282, 2022.

[34] C. A. Medina, M. R. Pérez, and L. C. Trujillo, "Iot paradigm into the smart city vision: A survey," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2017, pp. 695–704.

[35] L. Fiege, F. C. Gartner, O. Kasten, and A. Zeidler, "Supporting mobility in content-based publish/subscribe middleware," in *Middleware 2003*, M. Endler and D. Schmidt, Eds. Springer Berlin Heidelberg, 2003, pp. 103–122.

[36] E. R. Marques, G. M. Gonçalves, and J. B. Sousa, "Seaware: A publish/subscribe communications middleware for networked vehicle systems," in *7th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC'06)*, vol. 2006, 2006.

[37] M. Bacco, P. Cassará, A. Gotta, and T. de Cola, "Diversity framed slotted aloha with interference cancellation for maritime satellite communications," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[38] M. Bacco, M. Colucci, and A. Gotta, "Application protocols enabling internet of remote things via random access satellite channels," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[39] Unity Technologies. Unity 3d platform for real-time content creation. [Online]. Available: https://unity.com

[40] Eclipse Foundation. Mosquitto: An open source mqtt message broker. [Online]. Available: https://mosquitto.org

[41] ——. Paho:open source library implementations of mqtt and mqtt-sn for different programming languages. [Online]. Available: https://www.eclipse.org/paho/

[42] M. Bacco, P. Cassara, A. Gotta, and T. de Cola, "Diversity framed slotted aloha with interference cancellation for maritime satellite communications," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.