CargoAssist: Hybrid Communication Approach for Cargo Monitoring and Anomaly Detection

Mohamad Omran, Tim Brockmann, Frank Golatowski, Christian Haubelt
Institute of Applied Microelectronics and Computer Engineering, *University of Rostock*, Germany
Email: firstname.lastname@uni-rostock.de

***Abstract*—Cargo monitoring is essential for ensuring the safety and integrity of transported goods in maritime logistics. Traditional monitoring systems rely on wired communication, which is inflexible and costly. The *CargoAssist* project introduces an advanced wireless communication system utilizing *LoRa* for low-power, long-range data transmission and WiFi for high-rate local data exchange to enable real-time cargo condition monitoring onboard ships. Additionally, WiFi Mesh technology is employed to interconnect gateways, ensuring robust and self-healing network coverage across the ship. By combining these technologies, CargoAssist enhances data reliability, latency performance, and scalability, addressing the unique challenges posed by maritime environments.**

The project is currently in the development and evaluation phase, focusing on optimizing communication performance and system scalability. This paper outlines the ongoing work, and next steps toward a fully operational solution.

***Index Terms*—Maritime Logistics, Cargo Monitoring, Hybrid Wireless Communication, LoRaWAN, Wi-Fi Mesh (IEEE 802.11s), Anomaly Detection, Maritime Digitalization, Onboard Sensor Networks.**

I. Introduction

The maritime industry is experiencing a transformative shift toward autonomous and highly automated shipping, driven by the need for increased operational efficiency, reduced costs, and improved safety. In recent years, numerous research initiatives and pilot projects have demonstrated the feasibility and advantages of automation in maritime navigation. However, while significant progress has been made in autonomous navigation and remote monitoring, the automation of onboard operational processes remains an area requiring further development. A critical challenge in this context is the hazardous phenomenon of cargo liquefaction[1], where solid bulk materials such as nickel ore, iron ore fines, or wet coal transition into a liquid-like state due to moisture content and external forces such as ship movement and vibrations. Liquefaction can lead to sudden cargo shifts, destabilizing the vessel, increasing the risk of capsizing, and posing significant threats to crew safety, environmental protection, and cargo integrity. Addressing this issue requires advanced sensor networks and automated monitoring solutions capable of detecting early signs of cargo instability and issuing real-time alerts to prevent catastrophic failures.

A crucial aspect of this transition is the integration of intelligent digital platforms capable of monitoring shipboard systems, analyzing operational data, and generating automated responses. Existing solutions, such as digital logbooks, cloud-based MRV systems, and automated compliance tools, have laid the groundwork for further advancements. One such example is the Inventory of Hazardous Materials (IHM) maintenance tool, developed by NautilusLog[2], which ensures regulatory compliance by automating data processing and real-time reporting in accordance with EU regulations and the Hong Kong Convention of the International Maritime Organization (IMO).

Building on these advancements, the next phase of development envisions self-learning ships that actively support crew operations and, in certain cases, assume control over key processes. The CargoAssist project aims to explore this concept by deploying a network of industrial-grade sensors across various ship areas, including the engine room, cargo holds, ballast water systems, and bridge. These sensors will provide continuous monitoring of critical parameters such as temperature, humidity, accelerations, and structural forces, feeding data into a centralized decision-making platform.

The ability of such platforms to generate actionable insights, automated interventions, and emergency alerts represents a significant step toward fully autonomous ship operations. Moreover, a modular and scalable architecture ensures adaptability across different vessel types, operational environments, and propulsion technologies, extending applicability beyond maritime shipping to inland waterways.

This paper explores the technological framework necessary for autonomous onboard process automation, examining the challenges, opportunities, and implications of integrating intelligent sensor networks and decision-support systems in the maritime industry. The findings contribute to the broader discourse on maritime digitalization and serve as a foundation for future research on highly automated and autonomous ship operations.

Summarized, the key contributions of CargoAssist are:

- Real-time monitoring: Tracks cargo conditions via sensors.
- ML-based anomaly detection: Identifies hazards early.
- Automated logging: Records sensor data for analysis.
- Hybrid communication: Uses LoRaWAN and WLAN mesh.
- Energy efficiency: Extends sensor battery life.
- Seamless integration: Ensures smooth data exchange.
- Scalability: Adapts to different vessels and cargo.

[1]https://www.theshipyardblog.com/liquefaction-of-bulk-cargo-explained/

[2]https://www.nautiluslog.com/

II. Technical Background

A. LoRa, LoRaWAN and ChirpStack

LoRa (Long Range) is a wireless communication technology designed for long-range, low-power data transmission. It operates in the sub-GHz ISM (Industrial, Scientific, and Medical) bands, which are license-free frequency ranges available for public use. These bands vary by region, with 868 MHz used in Europe and 915 MHz in North America. *LoRa* uses Chirp Spread Spectrum (CSS) modulation, which changes signal frequency over time, making it resistant to interference and enabling long-range communication. Its signals can be detected even below background noise, and low-frequency bands improve obstacle penetration. These characteristics make LoRa ideal for low-power IoT applications like smart cities, agriculture, and industrial monitoring. Data is encoded in frequency sweeps (chirps), with the Spreading Factor (SF) determining transmission robustness and speed. Higher SF (e.g., SF12) improves range and interference resistance but lowers data rates, while lower SF (e.g., SF7) allows faster transmission at the cost of reduced range [1].

LoRaWAN (LoRa Wide Area Network) is a network protocol that works on top of *LoRa* technology. It defines how *LoRa* devices communicate with gateways and network servers. *LoRaWAN* supports bi-directional communication, adaptive data rates, and device authentication, making it ideal for large-scale IoT deployments. The architecture consists of end devices (LoRa Nodes), gateways, a network server, and an application server to manage communication efficiently [2].

ChirpStack[3] is an open-source *LoRaWAN* network server that provides a complete software stack for connecting *LoRaWAN* nodes and gateways. Figure 1 illustrates a typical *LoRaWAN* architecture using *ChirpStack*. *LoRa* nodes transmit data to gateways over *LoRa*, and these gateways use a Packet Forwarder to relay the data to the *ChirpStack* Gateway Bridge. From there, the gateway bridge forwards the *LoRaWAN* packets via `MQTT`[4] (Message Queuing Telemetry Transport) to the *ChirpStack* network server through a broker.

For fast, low-latency communication between the network server and the application server, *ChirpStack* leverages `gRPC`[5] (Google Remote Procedure Call). While `MQTT` follows a publish-subscribe model that works well for distributed messaging, `gRPC` provides direct, bidirectional communication over `HTTP/2`, reducing overhead and improving real-time responsiveness in specific scenarios.

Finally, the *ChirpStack* application server handles device management, data decryption, and integration services, enabling seamless connections to external platforms via `HTTP`, `MQTT` and other protocols. It also provides a web-based interface for administrators to monitor network activity, manage devices, and configure system settings. This architecture ensures a secure, scalable, and efficient LoRaWAN solution, making ChirpStack a robust choice for long-range, low-power IoT deployments.

[3]https://www.chirpstack.io/
[4]https://mqtt.org/
[5]https://grpc.io/

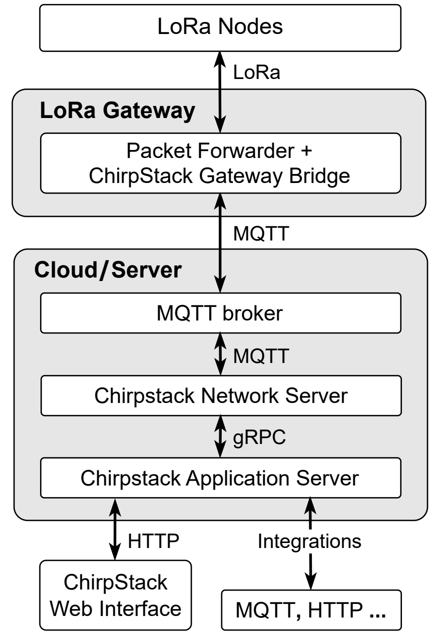

Fig. 1: LoRaWAN Architecture based on ChirpStack.

B. WLAN Mesh Networks and IEEE 802.11s

WLAN Mesh Networks (Wireless Local Area Network Mesh) enable multiple Wi-Fi access points (nodes) to collaborate, forming a self-organizing and self-healing wireless network. Unlike traditional WLANs, which rely on a central access point, WLAN mesh networks dynamically optimize data paths, enhancing coverage and reliability even in the event of node failures. *WLAN Mesh* is standardized under *IEEE 802.11s*, an extension of the *IEEE 802.11* standard that defines how wireless devices form a mesh topology. Operating primarily in the 2.4 GHz and 5 GHz frequency bands, *WLAN Mesh networks* use proactive routing protocols such as *Hybrid Wireless Mesh Protocol* (HWMP) to dynamically route data through the most efficient paths. Due to its resilient, scalable, and decentralized architecture, *WLAN Mesh* is widely deployed in smart cities, industrial automation, and remote monitoring systems.

III. Related Work

Existing research has explored sensor technologies for harsh environments, maritime communication, and onboard data processing. However, challenges remain in detecting cargo liquefaction, ensuring reliable wireless communication on ships, and automating cargo monitoring. This section reviews related work in these areas, highlighting key advancements and limitations.

A. Sensor Technology

Although industrial sensors for harsh environments exist (e.g., [3]), current systems cannot detect cargo liquefaction.

The required combination of physical effects, such as humidity, vibration, and ship movements, remains unclear. Previous research, like "LiquefAction" [4], focused on structural and operational guidelines but lacked automated detection methods. This study aims to explore cost-effective sensors and data-driven strategies, leveraging edge computing and cloud platforms.

B. Communication Technology

Wired communication on ships is complex and inflexible because it relies on dedicated cabling and connection points that must be integrated during the ship's construction. For example, installing or modifying cables later typically requires specialized labor, significant ship downtime, and expensive retrofitting. While wireless communication addresses some of these issues by offering greater flexibility, it also introduces new challenges.

Most research in maritime wireless communication focuses on ship-to-ship and ship-to-shore communication. These solutions primarily rely on existing systems such as the *Automatic Identification System* (AIS), radar, sonar, marine radio bands, satellite communications, and mobile networks, as discussed in [5]. To improve coverage, some approaches propose deploying wireless mesh networks using buoys [6], floating relay stations, or unmanned surface vessels [6]. These elements act as relay points to extend coverage. However, their effectiveness depends on line-of-sight connectivity, which can be severely affected by weather conditions and sea dynamics.

The CargoAssist project focuses on wireless communication onboard these ships to reliably transfer cargo monitoring data, such as sensor readings, video data, and alerts. However, challenges like steel walls and obstacles cause signal and interference problems [7, 8]. To solve these issues, it may be helpful to use a combination of communication technologies, such as LoRa, Bluetooth, WLAN, and others.

Past research often used *Wireless Sensor Networks* (WSN) technologies (e.g., *ZigBee*, *Bluetooth Low Energy* (BLE)), which require many nodes due to their low bandwidth and short range. For example, previous studies on marine environment monitoring have shown that *WSN* technologies face challenges in range and bandwidth limitations, making them less suitable for high-data applications [9]. On ships, *WLAN* is mainly used in public areas with centralized access points.

Proprietary solutions like *ScanReach* [10] use sub-GHz mesh nodes, but their closed hardware and protocols make it hard to adapt or integrate them with other systems. Sub-GHz networks do offer a longer range, yet they have lower data throughput, which may not meet the needs of data-heavy tasks like video or LiDAR. By contrast, *WLAN mesh networks* based on open standards like *IEEE 802.11s* deliver higher bandwidth and broader coverage using commercial off-the-shelf (COTS) hardware. This makes them more cost-effective and easier to fit into existing ship setups.

A hybrid communication approach works best for reliable and efficient data sharing on ships. *WLAN mesh* is perfect for high-bandwidth needs and can be placed in key areas for fast data transfer. *LoRa*, on the other hand, is better for status and event-based data because it covers a long range but supports lower data rates. By combining *WLAN mesh* for large data and *LoRa* for smaller, long-range communication, you can address different data requirements effectively.

The *University of Rostock* has deep expertise in IoT and *WLAN mesh* optimization—especially in improving *IEEE 802.11s* networks on Linux [11–14].

C. Data Processing & On-Board Assistance Systems

The maritime industry still relies heavily on paper-based workflows, even though documentation and regulations are strict. Challenges such as varying crew expertise, language barriers, and a shortage of skilled workers make things more difficult. Missing or incomplete documentation often makes it unclear whether damages are caused by crew actions, logistics partners, or technical issues. Additionally, incompatible processes, technical limits, and conflicts of interest between stakeholders, like cargo owners and insurers, add to the complexity. For example, opening a cargo hold at sea is not allowed due to safety risks for the ship, crew, and cargo.

Past projects like *B-ZERO* [15] focused on reducing crew workload by providing automated alerts, but these were limited to navigation issues and did not cover cargo monitoring.

Similarly, the *Horizon-2020* project *SmartShip* [16] targeted energy management but did not address cargo monitoring or predicting hazards too. Currently, cargo monitoring is done manually with rigid documentation, and there are no systems that can adapt, learn, or automate processes. This limits efficient data sharing between the ship, shore, and end-users.

Developed as part of the *ITEA* research project *I2PANEMA* [17], the *ISO 4891*[7] standard provides a framework for the interoperability of smart applications on ships, enabling seamless communication and data exchange to enhance efficiency, safety, and sustainability. *CargoAssist* aims to build on these advancements in the maritime sector and transfer their benefits to cargo management, enhancing efficiency, safety, and sustainability.

Although modern ships have the necessary hardware for computing and communication, it is mostly used to display data rather than analyze it or generate actionable insights. To overcome these challenges, a smart system is needed that combines sensor-based cargo monitoring with automated, self-learning data analysis to improve safety and efficiency.

IV. Realization

To address the need for reliable, flexible, and secure onboard cargo monitoring, the system is structured into four main layers: Sensor, Processing, Application, and Business. Each layer has specific tasks for efficient data collection, processing, and decision-making. This design improves real-time responsiveness, eases maintenance, and supports future scalability.

- **Sensor Layer:** Physical sensors, such as those measuring humidity, vibration, and motion, utilize low-bandwidth

[6]https://broadcast-solutions.de/

[7]https://www.dinmedia.de/de/norm/iso-4891/386108605

protocols (e.g., *ZigBee*, *LoRa*) for simple readings or switch to *WiFi* for higher data rates (e.g., video). By separating low- and high-bandwidth needs, the system avoids congestion and efficiently handles diverse data. Sensor readings then move to the Processing Layer through local hubs.
- **Processing Layer:** Mesh-enabled hubs receive sensor data. A server running on the hubs manages enrollment and control of low-bandwidth sensors. These hubs connect to each other through a wireless mesh, providing robust local coverage. They then forward collected data securely to the Application Layer over WLAN.
- **Application Layer:** A *System Interaction Unit*(SIU) on a portable device processes raw sensor data, configures and removes sensors, and integrates new hubs. However, a single centralized unit is responsible for storing all collected data in a database. This central unit also ensures secure *VPN*-based ship-to-shore communication, enabling the transmission of critical information for further analysis and distribution.
- **Business Layer:** Analyzes collected data using machine learning to detect anomalies and predict possible issues. It provides essential updates to the shipowners, and logistics partners, improving risk detection (e.g., cargo liquefaction) and sustaining continuous Ship2Shore communication.

The following implementation, as illustrated in Figure 2, builds on this four-layer concept to ensure a comprehensive and reliable view of cargo conditions. Each layer is mapped to specific hardware and software components, maintaining seamless data flow from acquisition to final analytics.

To enable sensor-based monitoring, the system utilizes the `Heltec LoRa 32`[8] (Version 3.1), an ESP32-based microcontroller unit (MCU) that integrates *LoRa*, *Wi-Fi*, and *Bluetooth* connectivity. It supports common bus interfaces such as I^2C, *SPI* and more, allowing connection to a wide range of sensor types. A key feature is its deep sleep mode, which consumes only about 10 µA of current, making it ideal for long-term, low-power applications. Each MCU-sensor combination is referred to as a *Smart Sensor*, which we classify into four main categories based on power consumption and sensor type:

- **Class 1 / Class 2: Low-Rate Physical Sensors**
  These sensors operate at low data rates and utilize LoRa for efficient long-range communication with minimal power consumption. **Class 1** sensors transmit raw measurements, such as temperature or air pressure, at regular intervals (e.g., every five minutes). In contrast, **Class 2** sensors perform local data processing (Edge Computing) to analyze measurements and detect critical conditions. Instead of continuously transmitting raw data, they send only event-based notifications when relevant changes occur, optimizing network efficiency and power usage.

[8]https://heltec.org/project/wifi-lora-32-v3/

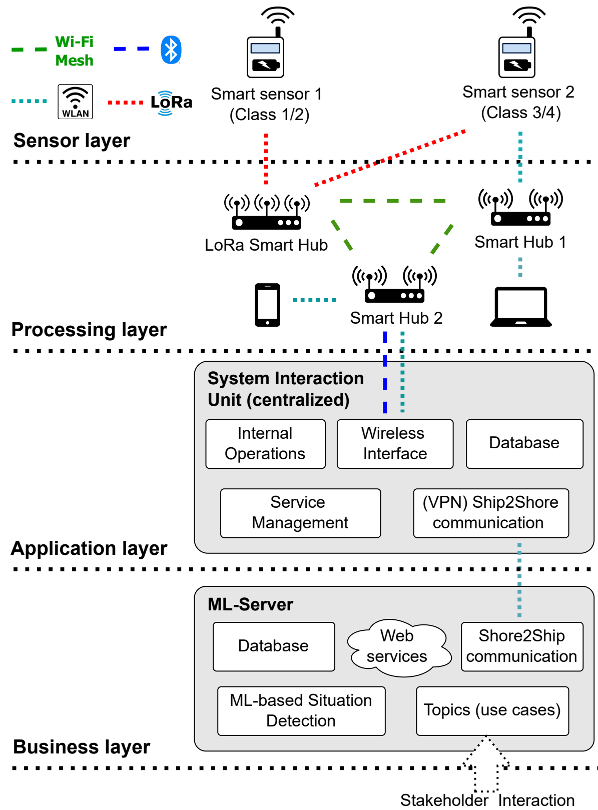

Fig. 2: CargoAssist System Architecture.

- **Class 3 / Class 4: High-Bandwidth Sensors**
  These sensors require higher throughput and rely on *WLAN*. Examples include video, LiDAR, and thermal imaging (FLIR). **Class 3** sends discrete images at set intervals, while **Class 4** provides continuous live streaming on a "best effort" basis. Both consume more power compared to Classes 1 and 2.

In the *Processing Layer*, the system consists of multiple *Smart Hubs*, each implemented on a single-board computer `PC Engines APU2D2`[9] equipped with a *WLAN* card supporting mesh networking and *Bluetooth*. These hubs form a scalable *wireless mesh network*, ensuring seamless intra-ship communication. Additionally, each *Smart Hub* is capable of connecting via traditional *Wi-Fi*, for both establishing a ship-wide *Wi-Fi* network for daily use and facilitating data transfer to the *Application Layer*.

However, only one hub is equipped with *LoRa*, as a single unit covers the entire ship. This *LoRa Smart Hub* integrates a `MikroTik R11e-LR8`[10] module, *ChirpStack* components, an `MQTT` broker, and a *Packet Forwarder*. The *ChirpStack Network Server* subscribes to `Mosquitto`[11] `MQTT` topics to receive *LoRaWAN* data, validates frames, manages network parameters, and handles device activation. The *Application*

[9]https://www.pcengines.ch/apu2d2.htm
[10]https://help.mikrotik.com/docs/spaces/UM/pages/14222503/R11e-LR8
[11]https://mosquitto.org/

Server then translates network data into application-level information.

To add new *Smart Sensor* nodes, we developed a process in the *LoRa Smart Hub* that relies on a pre-assigned `devEUI` (Device EUI) and `AppKey` (Application Key) exclusively for the initial connection. The `devEUI` is a 64-bit globally unique identifier that distinguishes each device on the network, while the `AppKey` is a private key used solely during the initial join process to authenticate the node and generate secure session keys.

Once a sensor successfully connects, the *System Interaction Unit* (SIU), a mobile device or tablet used by the crew, displays a message asking whether to add a new node. Upon user approval, the hub assigns the sensor a new, unique `devEUI` and sends it over *LoRa*. Simultaneously, the *Smart Hub* registers the newly added device in *ChirpStack*. Although *ChirpStack* primarily operates using `gRPC`, the *Smart Hub* triggers the registration with an initial `REST API` request. *ChirpStack*'s *API gateway* intercepts this request and converts it into a `gRPC` call, acting as an intermediary to manage protocol conversion, authentication, and request routing. From that point onward, both the hub and the sensor use this new `devEUI` for authentication and secure communication, adhering to *LoRaWAN* security guidelines. This method streamlines setup, allowing nodes to be seamlessly added or removed.

When a new *Smart Hub* is introduced to the network, it initially pairs with the *SIU* via *Bluetooth Low Energy* (BLE). During setup, the *SIU* provides essential network parameters, such as the *Service Set Identifier* (SSID), *WLAN* password, and server IP. After configuration, the new *Smart Hub* integrates seamlessly into the *mesh network*.

The *SIU* establishes a secure *VPN*-based ship-to-shore communication link, granting access to *Web Services* on the *ML-Server* for data processing. In the *Business Layer*, the *ML-Server* plays a pivotal role in analyzing sensor data, employing machine learning techniques for real-time pattern recognition and anomaly detection. The resulting insights are then delivered through *Web Services*, enabling stakeholders to monitor the system and make informed decisions.

Figure 3 illustrates how these components are distributed across the cargo ship. Because *LoRa* covers the entire vessel, the *LoRa*-enabled *Smart Hub* is typically placed on the bridge, offering a star topology to all *LoRa* sensors while minimizing interference. Additional *Smart Hubs*, without *LoRa* interfaces, are located across the ship to handle *Wi-Fi* and *Bluetooth* coverage. These hubs form a unified *WLAN mesh*, supporting local data exchange and routing collected sensor information to the *Application Layer*.

This integrated system ensures robust maritime cargo monitoring by combining low-power sensing, mesh networking, centralized data management, and advanced analytics for efficient, data-driven decision-making in ship operations.

V. Evaluation

To evaluate the system, we relied on several metrics such as the *Received Signal Strength Indicator* (RSSI) and the *Signal-to-Noise Ratio* (SNR). Additionally, we implemented other metrics such as packet loss and latency to gain deeper insights into the system's performance.

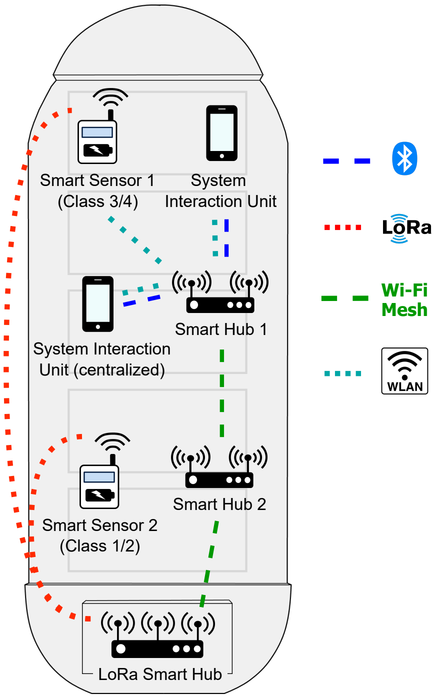

Fig. 3: On-Board Communication Diagram.

A. Packet Loss

Packet loss occurs when transmitted packets do not successfully reach their destination. To measure packet loss, we implemented a simple counter in the *Smart Node*. This counter starts at 0 and increases by 1 every time a transmission is done. When a response is successfully received, the counter decreases by 1. The *Smart Node* includes the current value of this counter in each *LoRa* packet sent to the *Smart Hub*. If the counter value grows over time, it indicates that multiple packets have been lost.

B. Latency Measurement

Latency refers to the time required for data to travel across the physical channel in a system, including propagation delay and transmission time. We developed a method to measure the latency of a communication cycle, which includes data transmission and response reception. This method involves sending a timestamp from the *Smart Hub* to the *Smart Sensor*. The *Smart Sensor* updates its *Software Timer* using this timestamp and then sends back a new timestamp along with its response. When the response reaches the *Smart Hub*, we can accurately calculate the latency since we have both the transmission and reception times. By subtracting the processing time at the *Smart Sensor*, we determine the actual communication latency.

In our experiment (Figure 4), the communication cycle begins with a *join request* from the *Smart Sensor* to the *Smart*

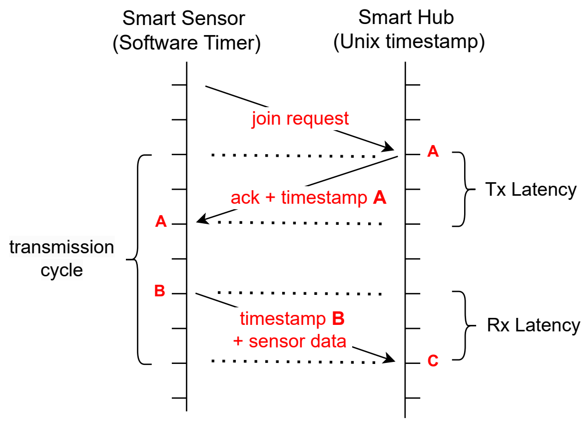

Fig. 4: Latency Calculation Diagram.

Hub, which responds with an acknowledgment and timestamp. The sensor updates its timer, processes data, and sends a response with a new timestamp and sensor data. Finally, the Smart Hub calculates the latency using the following Equation 1.:

$$Latency = (C - A) - (B - A) \quad (1)$$

We conducted an experiment to assess how distance affects RSSI, latency, packet loss (PL), and SNR, with the goal of identifying the reliable operational range of our LoRa system. The experiment was conducted using SF12 in a semi-open area, with the Smart Hub positioned approximately 18 meters above the ground. The results are presented in Table I.

TABLE I: Effect of Distance on RSSI, Latency, SNR, and Packet Loss (PL) using SF12

Distance (m)	Latency (s)	RSSI (dBm)	SNR (dB)	PL
300	4	-75	11	0
500	4	-76	8.2	0
1100	5	-107	4.8	0
1500	5	-110	2.5	0
1800	5	-112	-4	0
2300	26	-115	-12.8	2

Up to $1.8\,\text{km}$, *RSSI* gradually decreased, while latency remained stable at $5\,\text{s}$. *SNR* dropped from $11\,\text{dB}$ at $300\,\text{m}$ to $-4\,\text{dB}$ at $1800\,\text{m}$. Beyond this, latency increased due to weaker signals. Packet loss was zero until $2.3\,\text{km}$, where two packets were lost before a successful transmission, indicating reduced link reliability. Beyond $2.3\,\text{km}$, no packets were received, suggesting complete signal loss.

VI. Conclusion

The *CargoAssist* project introduces a hybrid wireless communication system for cargo monitoring in maritime environments. By integrating *LoRa* for long-range, low-power data transmission with *Wi-Fi Mesh* for high-bandwidth local communication, the system ensures efficient and reliable data exchange onboard ships while enhancing real-time monitoring capabilities. This enables early detection of potential cargo-related anomalies, improving overall operational safety.

Experimental results confirm the feasibility and effectiveness of the system in transmitting sensor data over long distances. Additionally, the proposed approach offers a promising solution for future anomaly detection. Future work will focus on optimizing *WLAN* performance to further enhance network reliability and coverage across the ship.

Acknowledgment

This work is currently being carried out within the research project *CargoAssist*, which is funded by the *Maritime Research Program*, an initiative of the *German Federal Ministry of Economics and Climate Protection* (BMWK) under the funding code `03SX601B`.

References

[1] M. Gilbert, "LoRaWAN® Gateways: Radio Coexistence Issues and Solutions,"

[2] H. U. Rahman, M. Ahmad, H. Ahmad, and M. A. Habib, "LoRaWAN: State of the Art, Challenges, Protocols and Research Issues," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Nov. 2020.

[3] *SmartGAS Mikrosensorik — smartgas.eu*, https://www.smartgas.eu, [Accessed 19-02-2025].

[4] *FDS: LiquefAction - Cargo Liquefaction in Ship Design and Operation — tuhh.de*, https://www.tuhh.de/fds/research/completed-projects/liquefaction-cargo-liquefaction-in-ship-design-and-operation, [Accessed 19-02-2025].

[5] K.-L. A. Yau, A. R. Syed, W. Hashim, J. Qadir, C. Wu, and N. Hassan, "Maritime networking: Bringing internet to the sea," *IEEE Access*, vol. 7, 2019.

[6] H. Chen, F. Yin, W. Huang, M. Liu, and D. Li, "Ocean Surface Drifting Buoy System Based on UAV-Enabled Wireless Powered Relay Network," *Sensors*, vol. 20, no. 9, Jan. 2020.

[7] B. De Beelde, E. Tanghe, M. Yusuf, D. Plets, and W. Joseph, "Radio Channel Modeling in a Ship Hull: Path Loss at 868 MHz and 2.4, 5.25, and 60 GHz," *IEEE Antennas and Wireless Propagation Letters*, vol. 20, no. 4, Apr. 2021.

[8] F. Dong and Y. H. Lee, "Non-line-of-sight communication links over sea surface at 5.5GHz," in *Asia-Pacific Microwave Conference 2011*, Dec. 2011.

[9] G. Xu, W. Shen, and X. Wang, "Applications of wireless sensor networks in marine environment monitoring: A survey," *Sensors*, vol. 14, no. 9, 2014.

[10] *ScanReach — scanreach.com*, https://www.scanreach.com/, [Accessed 19-02-2025].

[11] M. Rethfeldt, B. Beichler, P. Danielis, T. Brockmann, C. Haubelt, and D. Timmermann, "Chacha: Clustering heuristic and channel assignment for ieee 802.11s mesh networks," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018.

[12] M. Rethfeldt, B. Beichler, P. Danielis, F. Uster, C. Haubelt, and D. Timmermann, "Mentor: A wireless-mesh-network-aware data dissemination overlay based on bittorrent," *Ad Hoc Networks*, vol. 79, 2018.

[13] I. F. Priyanta, F. Golatowski, T. Schulz, and D. Timmermann, "Evaluation of lora technology for vehicle and asset tracking in smart harbors," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, 2019.

[14] T. Brockmann, M. Rethfeldt, B. Beichler, F. Golatowski, and C. Haubelt, "Mac-filter based topology control for wlan mesh networks," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2024.

[15] *B ZERO - Fraunhofer CML — cml.fraunhofer.de*, https://www.cml.fraunhofer.de/de/forschungsprojekte1/B-ZERO.html, [Accessed 19-02-2025].

[16] *Home — SmartShip — smartship2020.eu*, https://www.smartship2020.eu/, [Accessed 19-02-2025].

[17] J. van der Heide, *ITEA 4; Project; 17001 I2PANEMA — itea4.org*, https://itea4.org/project/I2PANEMA.html, [Accessed 20-02-2025].