Università degli Studi di Padova

Università degli Studi di Padova

Department of Information Engineering

Master Thesis in ICT for Internet and Multimedia

A Distributed Ranging Scheme for Underwater Acoustic Networks

Supervisor
Dr. Filippo Campagnaro
Università degli Studi di Padova

Master Candidate
Vincenzo Cimino

Academic Year 2023/2024
10/08/2024

Abstract

Swarms of Autonomous Underwater Vehicles (AUVs) are often employed in a wide range of missions, from civilian to military applications. However, the challenges imposed by the underwater acoustic channel make it particularly difficult to track the positions of the vehicles over time. Common underwater ranging systems rely on Two Way Travel-Time (TWTT) measurements, which require to measure the Round Trip-Time (RTT) of the signal sent. Combined with long propagation times and high ambient noise, this significantly limits the throughput of the underwater network. Although One Way Travel-Time (OWTT) reduces latency by half, by having synchronized clocks between all the nodes in the network, it necessitates the use of highly precise oscillators, such as Chip Scale Atomic Clock (CSAC) or Oven-Controlled Crystal Oscillator (OCXO), in each modem processing unit, which is expensive and energy-intensive. Nowadays ranging systems are shifting towards the use of distributed, low-cost and heterogeneous components for positioning and localization.

This thesis extends on the work conducted in [1], where a distributed ranging protocol is presented and simulated in the DESERT Underwater framework [2]. Specifically, we tested the protocol in real-world environments using actual modems, exploring the integration between the DESERT simulator and real devices, while addressing the challenges that arise when replacing a simulated physical layer with a real one. A key feature of this ranging protocol is its MAC-agnostic nature, allowing the underlying MAC protocol to be interchanged based on specific requirements.

Specifically, we evaluated the protocol using two distinct MAC protocols: a contention-based protocol and a contention-free protocol. We tested the protocol using two different modems: Evologics S2C R 18/34 WiSE [3] modems and low-cost modems, called Subsea acoustic Modem (SuM) [4] developed by SubSeaPulse SRL, a spinoff component of the Signet Lab at the University of Padova. Finally, we conduct an experiment using the Evologics modems on the Piovego River. The results, in terms of Root Mean Square Error (RMSE) of the measured distances and the jitter in the computed Time of Flights (ToFs), demonstrate that the DESERT Underwater Framework enables all AUVs in the swarm to accurately determine their distance from every other node in the network. This capability reduces the risk of vehicle collisions and improves mission coordination.

Contents

Abstract v

List of figures xi

List of tables xiii

List of acronyms xv

1 Introduction 1

2 State of the Art 5
- 2.1 Challenges in underwater acoustic communication . . . 5
- 2.2 Acoustic positioning techniques . . . 7
- 2.3 Baseline positioning systems . . . 8
- 2.4 Distributed Long BaseLine positioning systems . . . 9

3 System Description 13
- 3.1 Protocol stack and implementation . . . 13
  - 3.1.1 Packet structure . . . 15
  - 3.1.2 TAP layer . . . 16
- 3.2 Physical layer . . . 16
  - 3.2.1 Schedulers and Packers . . . 17
  - 3.2.2 Device drivers . . . 18
- 3.3 Modems description . . . 18
  - 3.3.1 S2C R 18/34 WiSE . . . 18
  - 3.3.2 SuM . . . 22

4 Test Settings 25
- 4.1 DESERT parameters . . . 26
  - 4.1.1 Packer settings . . . 27
- 4.2 Preliminary tests . . . 28
  - 4.2.1 S2C setup . . . 29
  - 4.2.2 SuM setup . . . 30
- 4.3 Piovego river test setup . . . 31

5 Results 33

5.1 Ranging efficiency metrics 34
5.2 S2C test results 35
5.3 SuM test results 38
5.4 Piovego river test results 41

6 Conclusions and Future Work **47**

References **49**

Acknowledgments **53**

Listing of figures

2.1 DLBL timing diagram for a network with 3 nodes. 10

3.1 Protocol stack of each node in the network. 14
3.2 Packet structure of a ranging message. 15
3.3 Structure of a `holdover_list` entry. 16
3.4 Interconnection between DESERT protocol stack and actual devices. . . . 19
3.5 Plots of the transmission duration and processing delay of S2C modems. . . 21
3.6 Plots of the transmission duration and processing delay of SuM modems. . 24

4.1 Packer protocol stack of each node in the network. 27
4.2 Setup with S2C modems, with each modem connected to a switch via ethernet. 29
4.3 Setup using SuM modems, with each modem connected to a set of audio speakers for output, a headset for input and to a switch via Ethernet. 30
4.4 Real-world experiment on the Piovego river's dock using S2C modems. . . 32

5.1 Box plots of the RMSE measured by each node in the network using S2C modems. 36
5.2 Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using S2C modems. The shaded area represents the 95% confidence interval for the mean. 37
5.3 Boxplots of the RMSE measured by each node in the network using SuM modems. 39
5.4 Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using SuM modems. The shaded area represents the 95% confidence interval for the mean. 40
5.5 Boxplots of the RMSE measured by each node in the network using S2C modems in the Piovego river. 43
5.6 Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using S2C modems in the Piovego river. The shaded area represents the 95% confidence interval for the mean. 44

Listing of tables

4.1 DESERT parameters common to all tests 26
4.2 Number of bits assigned to each layer of the protocol stack 28

Acronyms

AL Adaption Layer. 17, 27

AUV Autonomous Underwater Vehicle. vii, 1, 7

CSAC Chip Scale Atomic Clock. vii, 7

DLBL Distributed Long-baseline. 2, 9

DVL Doppler Velocity Log. 1

GIB GPS intelligent buoys. 8

GNSS Global Navigation Satellite System. 1, 7

IP Internet Protocol. 14

LBL Long-baseline. 1, 8

MAC Medium Access Control. 2, 9

MCM Mine Countermeasure Missions. 1, 7

NS2 Network Simulator 2. 17, 27

OCXO Oven-Controlled Crystal Oscillator. vii, 7

OWTT One Way Travel-Time. vii, 2, 7

RMSE Root Mean Square Error. vii, 29, 34

ROV Remotely Operated Vehicle. 1

RTT Round Trip-Time. vii, 2, 7

S2C Sweep Spread Carrier. 19

SBL Short-baseline. 1, 8

SuM Subsea acoustic Modem. vii, 2, 22, 33, 41

TDOA Time Difference Of Arrival. 2, 9

ToF Time of Flight. vii, 2, 7, 34

TWTT Two Way Travel-Time. vii, 2, 7

UDP User Datagram Protocol. 13

USBL Ultra-short-baseline. 1, 8

1

Introduction

The deployment of low-cost underwater assets such as Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) [5] is encouraging their use in swarms. In fact, these swarms can undertake a wide range of missions, including civilian applications like environmental monitoring [6], coastal surveillance [7], and assisted navigation [8], as well as military operations such as Mine Countermeasure Missions (MCM) [9] and military surveillance [10]. In AUVs formations, accurate navigation relies heavily on precise localization [11], which cannot be achieved using Global Navigation Satellite System (GNSS) due to the lack of signal underwater.

Common approaches often involve expensive specialised hardware, such as high precision inertial measurement units and gyroscopes paired with Doppler Velocity Log (DVL) systems. Moreover, traditional ranging and communication systems like Long-baseline (LBL), Short-baseline (SBL) and Ultra-short-baseline (USBL) [12] require multiple anchors and acoustic modems installed in the network nodes. However, current trends are shifting to-

wards the use of distributed, low-cost and heterogeneous components for positioning, such as Distributed Long-baseline (DLBL) [13] systems. Acoustic positioning primarily involves precisely measuring of the signal's Time of Flight (ToF) and its propagation speed (sound velocity) [14]. The precision in estimating ToF largely depends on the characteristics of the physical and data-link layers used. ToF can be determined using either One Way Travel-Time (OWTT) or Two Way Travel-Time (TWTT) methods. In the two-way approach, the Round Trip-Time (RTT) of the signal is measured, accounting for the ToF from the source to the target and back. In contrast, the one-way method requires synchronized clocks [15] between the source and target, allowing the transmission time to be measured during a single signal transfer.

A DLBL system has been implemented and tested within a simulated environment [1], using a module of DESERT Underwater [16], an open source [2] underwater network simulator developed and maintained by the SIGNET group at the University of Padova. This module, known as `uwRangingTDOA`, builds upon the research presented in [14], works on the principle that each node generates packets with a certain generation rate, which are broadcasted according to the Medium Access Control (MAC) policy. These packets contain Time Difference Of Arrival (TDOA) information for each node in the network. The receiving node uses this information to measure the ToF and estimate distances from all other nodes in the network, based on both direct and indirect measurements.

The goal of this thesis is to build upon previous work by testing the `uwRangingTDOA` module in a real-world experiment using actual acoustic modems. Specifically, we compare a low-cost modem, the Subsea acoustic Modem (SuM) [17], designed and built by SubSea-Pulse SRL, a spinoff company of the University of Padova, with a commercially available modem, the S2C R 18/34 WiSe [3], developed by Evologics GmbH. For both modems, we evaluated the system's performance using two different MAC protocols: a contention-free protocol known as `TDMA` and a contention-based protocol referred to as `CSMA-ALOHA`, with their implementations available in [2].

The rest of the thesis is organized as follows: Chapter 2 provides an overview of the current state of the art in underwater acoustic positioning systems, including an introduction to underwater acoustic communications. Chapter 3 details the protocol stack, its implementation in DESERT and includes a description of the modems used, as well as their integration into the simulation environment. Chapter 4 outlines the tests setup, while Chapter 5 presents and discusses the results obtained from the experiments. Finally, Chapter 6 offers conclusions and suggestions for future work.

2

State of the Art

This chapter introduces the key aspects related to underwater acoustic transmissions, with a particular emphasis on acoustic positioning and localization. It provides an overview of the principal ranging techniques, making a distinction between one-way and two-way measurements. These techniques are applied in commercial positioning systems, which are categorized based on their structural characteristics. Finally, the chapter presents a detailed description of a particular positioning system that forms the foundation of the research presented in this thesis.

2.1 Challenges in underwater acoustic communication

A thorough understanding of underwater acoustic communication systems [18], along with an awareness of their limitations and challenges, is essential for this work. In fact, acoustic communication and positioning systems often share the same electro-acoustic circuitry for transmitting and receiving signals [14]. Additionally, the same signal can simultaneously

carry both digital information and source position data.

The underwater acoustic channel is known to impose significant challenges to communication between underwater assets [19]. In the design of underwater acoustic networks, the main challenges include (i) heavy path loss and multi-path which severely impair the channel, (ii) long propagation delay which is five order of magnitude higher than in radio frequency terrestrial channel, (iii) low bitrate which is of the order of kbps (iv) and high dependence on weather conditions. Still, acoustic communication remains the most used and mature technology in underwater communication, due to the long distances (up to tens of kilometers) it can travel with a relative limited power and resilience to water turbidity. Most common solutions include expensive hardware which also contributes to making underwater acoustic communications more challenging. Nevertheless, a great effort is being put in the development of low-cost underwater vehicles and affordable underwater acoustic modems [20].

Other technologies such as optical communications [21] are rather limited by their short range (up to 100m), even though they allow for much higher datarate and bandwidth. Nevertheless, an increasing research effort is being put also on the realization of optical devices that can transmit data within short distances at a bit rate of the order of one or more Mbps. Unlike acoustic communications, optical communications are not affected by multipath and environmental conditions, but their performance mainly depends on water turbidity and sunlight noise [22].

Little research is being conducted on underwater radio waves [23], which, unlike acoustic and optical waves, can transition relatively smoothly through the air-water surface. However, a great disadvantage of radio waves is their high signal attenuation, allowing them to propagate only over short distances, typically up to tens of meters. This limitation significantly constrains their use in underwater communication systems, particularly in deep-sea environments where long-range communication is essential. Furthermore, the efficiency of radio waves underwater is highly dependent on factors like frequency and salinity, further complicating their practical application.

The use of low-cost AUV [5] swarms is a topic of increasing interest, since their missions can cover a large set of applications, ranging from civilian applications such as coastal monitoring and surveillance [7, 24] to military applications like Mine Countermeasure Missions (MCM) [9].

Precise navigation and localisation of underwater vehicles [11], is a key requisite in these contexts, which still remains a challenging issue nowadays. Due to the inavailability of radio signals underwater, localization cannot be obtained by means of GNSS. For this reason, a typical underwater acoustic positioning (or ranging) method relies on measuring the sound velocity and the signal ToF observing the RTT.

Generally speaking, the sound velocity in water is a complex function of temperature, salinity and depth. A sound speed profile, for instance, shows the sound velocity at different depths. The problem of measuring the sound speed goes beyond the scope of this work, thus, in the following we will assume that the sound waves propagate linearly and the sound velocity is known.

The TOF, on the other hand, can be measured either by using a OWTT or a TWTT ranging method.

- OWTT ranging is possible when all the node's clocks are synchronized. Such synchronization, requires the presence of high precision clocks such as CSAC [15], which are also highly expensive, or a less expensive and less accurate OCXO [25]. Since target and source node are synchronized, when the target receives a packet, it can directly compute the one-way travel time as the difference between the time the signal was received and the time it was transmitted.

- TWTT ranging, on the contrary, does not require nodes synchronization, but require more channel access and doubles the latency. The two-way travel time is computed as the time it takes for a signal to travel from a transmitter to a receiver and then back to the transmitter.

The distance is then computed by multiplying the ToF by the speed of sound.

In the second case, the result is then divided by a factor of two to account for the fact that the signal travels the distance twice.

2.3 Baseline positioning systems

The most common commercially available positioning systems are the so called baseline systems. These systems are characterized by the difference from some reference nodes (referred as baseline stations), called the baseline length, from which the name of such systems. Based on the distance between the baseline stations, these systems are categorized in three main types.

- Long-baseline (LBL) systems are comprised of baseline transponders deployed on the sea floor, placed 50 m to 200 m apart [19]. In some configurations, instead of being deployed on the seafloor, baseline nodes can even be placed on floating surface buoys with GPS positioning, known as GPS intelligent buoys (GIB) [26]. LBL systems are independent of water depth and have the highest possible accuracy (better than 1 m). Such systems, typically use a TWTT ranging computing the RTT from the target to baseline nodes. By analyzing the ToF data from multiple baseline nodes, the system calculates the position of the target nodes using trilateration. Essentially, it determines the targets location based on the distances from at least four baseline nodes: the distance from a baseline traces a sphere, the target node is positioned at the intersection of the four spheres.

- Short-baseline (SBL) are characterized by a rather short transponders distance, which ranges between 20 and 50 m. These systems are firmly mounted on floating platforms. The accuracy depends on the distance between the baseline transponders: the higher the distance the better the accuracy. The positioning method used is the same as LBL systems.

- Ultra-short-baseline (USBL) is the most convenient and popular of the three categories of underwater positioning systems. It is made of two elements: (i) a transceiver with an array of transducers, usually deployed less than 10 cm apart, (ii) and a transponder. Compared to the other two systems, it has the lowest positioning accuracy. The

distance from the target is obtained using a TWTT method. The transceiver also measures the angle at which the signal arrives. The angle of arrival measurements are used to triangulate the position of the target relative to the baseline.

2.4 Distributed Long BaseLine positioning systems

In addition, Distributed Long-baseline (DLBL) [14, 13] is a variant of a LBL system where there is no distinction between baseline and target nodes, but rather each node in the network has the same ranging roles and capabilities. Unlike previous systems, it does not require specific hardware nor the presence of anchors, drastically lowering the cost and complexity of the deployment. Moreover, DLBL is MAC agnostic, since it can be easily integrated into acoustic communication devices with different networking and MAC algorithms. DLBL uses a TWTT ranging method in order to provide localisation measurements to the nodes. Each node periodically receives broadcast messages from other nodes. Upon receiving a message, it stores the reception timestamp and, in subsequent transmissions, the replies include the holdover times: the time elapsed between the reception timestamps and the current transmission time.

In a network with N nodes, at the end of a cycle, each node will have collected $N - 1$ TDOA. Using these TDOA values and the sound speed, each node can then calculate the distances to every other node in the network as well as the distances between all pairs of nodes. Figure 2.1 shows the basic concept of DLBL in a network with three nodes. Each node n_i has its own system time t^i, which may differ from different nodes. The time instant at which node n_i broadcasts a message is denoted by $t^i_{b_i}$, while $t^i_{r_j}$ denotes the time instant at which node n_i receives a broadcast message from node n_j.

The example in figure, can be summarized as follows:

1. at $t^1_{b_1}$, n_1 sends a broadcast message to the other two nodes in the network;

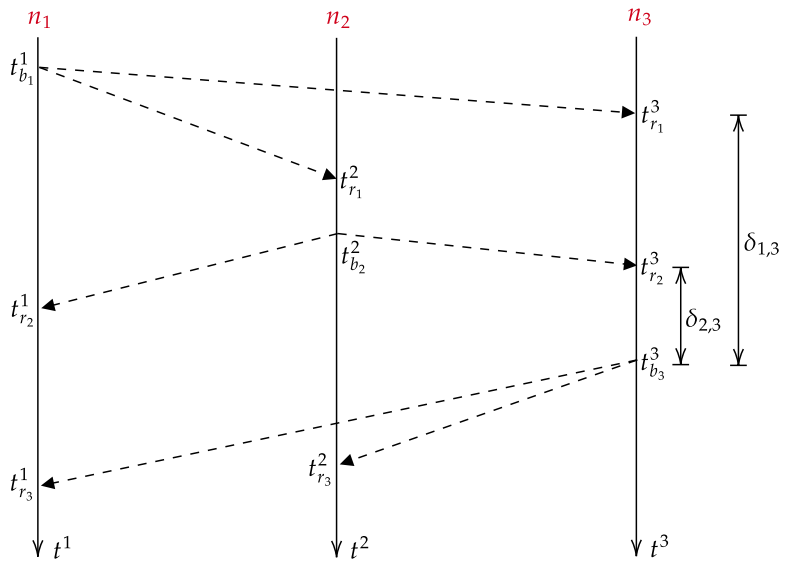


Figure 2.1: DLBL timing diagram for a network with 3 nodes.

2. at $t^2_{r_1}$ and $t^3_{r_1}$, n_2 and n_3 respectively, receive this message and store the reception times;

3. at $t^2_{b_2}$, also n_2 sends a broadcast message which is received at $t^1_{r_2}$ by n_1 and at $t^3_{r_2}$ by n_3;

4. at $t^3_{b_3}$, then n_3 transmits a broadcast message with a payload containing (i) the holdover time from the reception of the last packet from n_1, i.e., $\delta_{1,3} = t^3_{b_3} - t^3_{r_1}$ (ii) and similarly for n_2, $\delta_{2,3} = t^3_{b_3} - t^3_{r_2}$;

5. at $t^1_{r_3}$, n_1 finally computes the ToF of the message exchange with node n_3 as follows:

$$t^1_{1\leftrightarrow 3} = \frac{t^1_{r_3} - t^1_{b_1} - \delta_{1,3}}{2}. \tag{2.1}$$

This direct measurement is referred to as spherical positioning in [14]. Moreover, it computes the ToF of the message exchange between n_2 and n_3 using an indirect mea-

surement, referred to as hyperbolic positioning in [14].

$$t^1_{2\leftrightarrow 3} = t^1_{r_3} - t^1_{r_2} - t^1_{1\leftrightarrow 3} + t^1_{1\leftrightarrow 2} - \delta_{2,3}. \tag{2.2}$$

where $t^1_{1\leftrightarrow 2} = \frac{t^1_{r_2} - t^1_{b_1} - \delta_{1,2}}{2}$ is the spherical measurement between n_1 and n_2 and $\delta_{1,2} = t^2_{b_2} - t^2_{r_1}$ is the holdover time received at $t^1_{r_2}$ from n_2.

Equation (2.2) can be rearranged, by expanding the ToFs, that is

$$t^1_{2\leftrightarrow 3} = \frac{t^1_{r_3} - t^1_{r_2} + \delta_{1,3} - \delta_{1,2} - 2\delta_{2,3}}{2}. \tag{2.3}$$

Eventually, by multiplying these values by the speed of sound, the relative distances are obtained. A similar approach can be applied to the other two nodes in the network. It is important to note that direct measurements require two successful message receptions, whereas indirect measurements require five successful receptions.

3

System Description

This chapter is organized into three sections presenting the proposed ranging system implementation. Section 3.1 and Section 3.2 describes the DLBL protocol implementation within the DESERT Framework [2], including the protocol stack used for the experiments. Section 3.2 particularly emphasizes the physical layer and the interface between DESERT and real modems. Lastly, Section 3.3 provides a brief overview of the modems used and their associated driver modules.

3.1 Protocol stack and implementation

The DLBL ranging protocol described in Chapter 2, has been implemented in the DESERT underwater framework as a module called `uwRangingTDOA` [1]. Figure 3.1 shows the protocol stack used for this work. `uwApplication` is an application layer module, which allows a bidirectional communication between two given hosts. `uwUDP` and `uwIP` are respectively, (i) a simple implementation of a User Datagram Protocol (UDP), responsible of forwarding

packets to the right application and (ii) a simplified version of the Internet Protocol (IP) that minimizes the protocol overhead and that is responsible of forwarding packets to the right host. The `uwRangingTDOA` module, is placed below these layer so that the ranging packets are not loaded with other protocol headers. This ranging protocol is independent of the underlying `MAC` protocol. Therefore, for the time being, we do not specify any particular protocol in the figure. Precise transmission and reception timestamps are still needded. To perform this task, there is a module called `uwTAP`, interposed between `MAC` and `PHY` layers, which notifies the ranging protocol when the `MAC` schedules a new transmission. The `PHY` layer plays an important role in this thesis and will be discussed in Section 3.2.

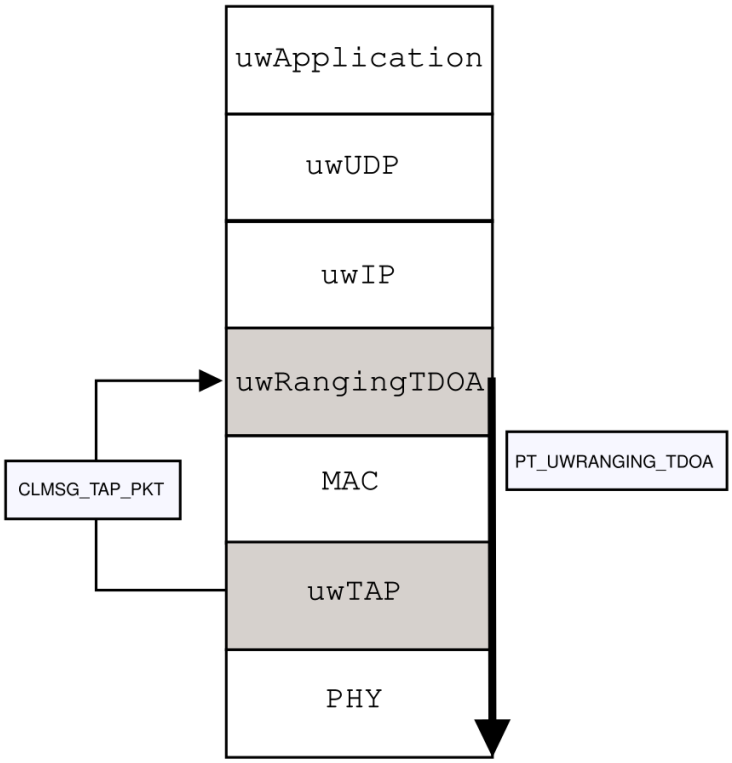


Figure 3.1: Protocol stack of each node in the network.

3.1.1 Packet structure

The uwRangingTDOA protocol, periodically sends broadcast messages containing the ranging information as PT_UWRANGING_TDOA packets. The structure of such packets is shown in Figure 3.2.

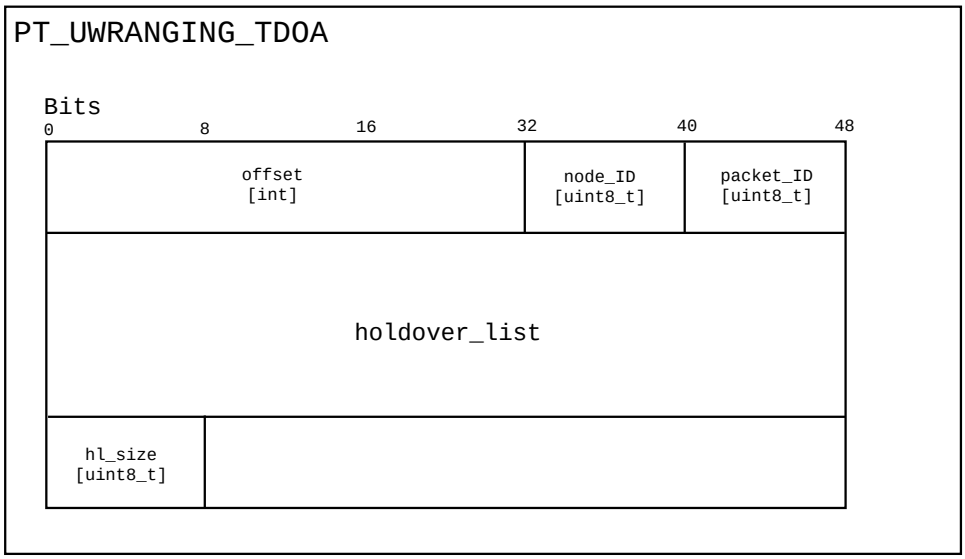


Figure 3.2: Packet structure of a ranging message.

Field offset is needed by ns-Miracle's PacketHeaderManager class to access the header. Fields node_ID and packet_ID are respectively the identifier of a node and a packet in the network. The second identifier avoids the possibility that two ranging packets from the same node get aliased, which is a possibility when a packet is received by some node but lost by others.

Finally, we have the holdover_list field, whose size is variable and stored inside the hl_size field. As explained in [1], the size can be optimally computed, but for the purpose of this work, it is always set to be equal to the number of nodes minus one. Each entry of the holdover_list field has the structure depicted in Figure 3.3. Each entry stores a holdover_time, which is the time between the reception of a packet, identified by the

packet_ID field, from a node with identifier Node_ID, until the transmission of the current packet.

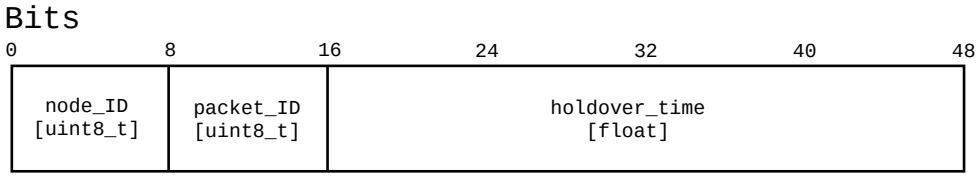


Figure 3.3: Structure of a holdover_list entry.

3.1.2 TAP layer

The generation policy for the uwRangingTDOA module, is fixed with a period $T = 1/\lambda$, where the parameter λ can be set accordingly to the system needs. When a new packet is generated for transmission, each holdover_time is computed as the current time minus the time at which the last packet from a given node was received. The generated packets are then sent down the stack to the MAC protocol, which schedules their transmissions according to its policy. As soon as the uwTAP module receives a packet from the upper layer, it notifies the uwRangingTDOA module via a cross-layer synchronous message CLMSG_TAP_PKT, containing the transmission duration computed by the physical layer. On reception of the cross-layer message, uwRangingTDOA adds to the holdover_time, the time elapsed since the generation of the packet and the transmission duration. To compute the elapsed time from generation and actual transmission, each packet generation is associated with a timestamp.

3.2 Physical layer

The main objective of this thesis is to extend the work conducted in [1] to real-world field experiments. To achieve this, we need to substitute the simulated physical layer with a real one. This involves three distinct processes: (i) replacing the simulated scheduler with a real-time

scheduler; (ii) translating virtual packets into real packets, and vice versa; and (iii) connecting DESERT with actual modems.

3.2.1 Schedulers and Packers

DESERT inherits its inner simulator structure from Network Simulator 2 (NS2), which is based on a linked list of events. Each event is comprised of:

- a time of happening;
- a link to the previous event and a link to the next one;
- the handler reference.

The handler is a C++ class responsible for executing the event's designated actions. The event manager, known as the scheduler, is in charge of setting the event's timing, i.e., when the handler should be called and the event consumed. In a simulated environment, the scheduler has its own time reference. Conversely, a real-time scheduler's clock is continuously synchronized with the host's current time. This may introduce some delay, as explained in Chapter 4.

In NS2, a packet is an object of the class `Packet`, which inherits from the class `Event`. These packets are not actual packets but rather events that are generated and passed down the stack to be scheduled for transmission, i.e., to be consumed. To translate virtual packets into real ones, we use DESERT modules called `Packers`. These modules map a NS2 packet into a binary string that a modem can understand. The `Packer` class is defined inside the Adaption Layer (AL) module and provides methods for packing and unpacking virtual packets. For this work, we specifically developed a module called `packer_uwRangingTDOA`, which inherits the methods for packing and unpacking and redefines them for the packets depicted in Figure 3.2.

Note that to have a proper translation, we need to correctly configure and add the packers for all layers within the protocol stack.

3.2.2 Device drivers

Once we have properly converted packets into binary strings, we need a so called device driver: a module which understands the language of a specific device and works as interface between it and DESERT. A driver is responsible for interpreting the status of the modem (and its meaning), parsing its responses, and controlling its operations.

Every driver module inherits basic methods from the `UwModem` class, which includes declarations for transmission and reception queues, a handler for executing modem events, and methods for managing transmission and reception. Additionally, drivers make use of a connector to send and retrieve bits and an interpreter to understand their meaning. The class `UwConnector` implements the methods required for the actual connection between drivers and modems, which can happen via the creation of a TCP/UDP socket or a serial interface.

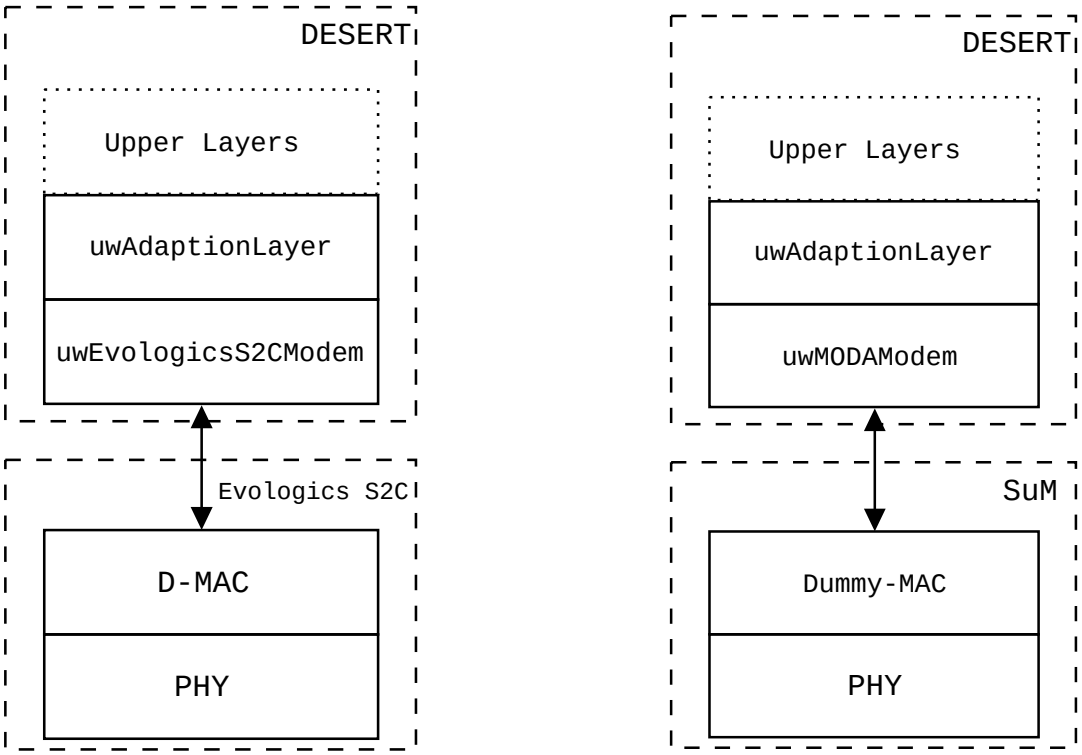
Figure 3.4 shows the actual physical layers used in this thesis, which include the AL and modem drivers, along with their connection to the actual modems. The `uwMODAModem` and `uwEvologicsS2CModem` modules, which are built into the DESERT framework, are respectively, the drivers for the SuM modems and Evologics S2C modems. These are the modems used for the tests.

3.3 Modems description

The next two sections provide a brief overview of the modems used in this thesis, along with their characteristics and the modifications required for their drivers to meet our needs.

3.3.1 S2C R 18/34 WiSE

The S2C R 18/34 WiSE [3] (referred to as S2C, in the following) is an underwater acoustic modem with an horizontally omnidirectional beam pattern, optimized for short to medium-range transmissions. It operates within a range of up to 3500 m and uses a frequency band of 18-34 kHz.



(a) Actual physical layer and its connection with Evologics S2C modems.

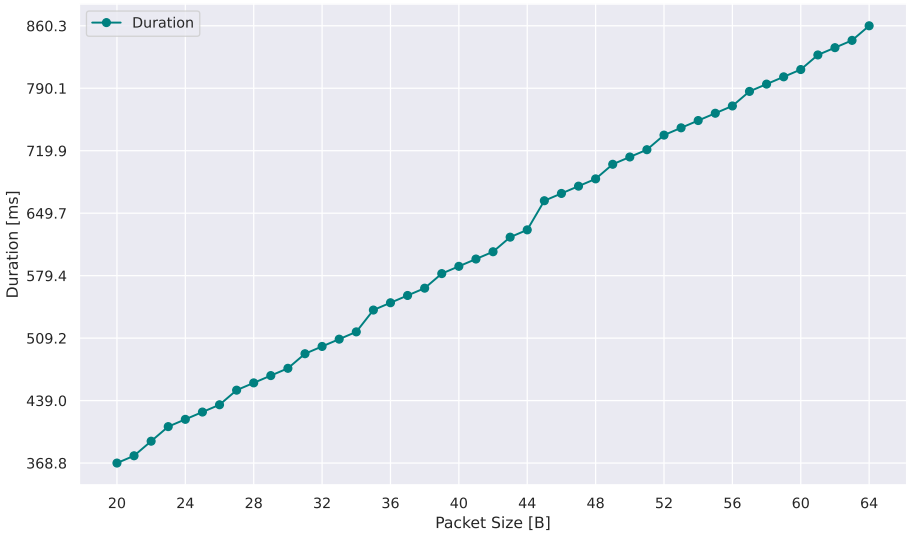
(b) Actual physical layer and its connection with SuM modems.

Figure 3.4: Interconnection between DESERT protocol stack and actual devices.

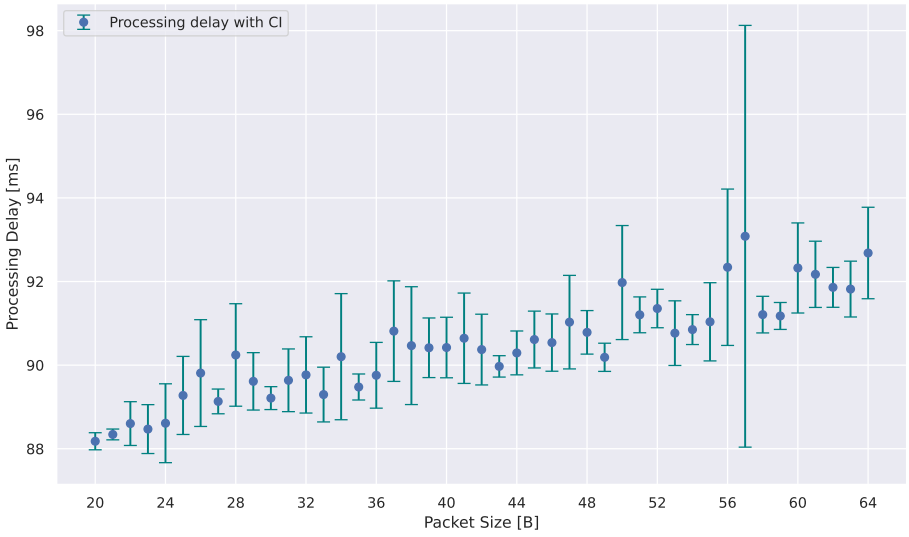
The modem has a power consumption of less than 1.3 W on reception, and between 2.8 W and 80 W during transmission, depending on the distance. The physical layer implements the Sweep Spread Carrier (S2C) spread spectrum signal modulation technique. At the data link layer, it implements the D-MAC protocol, which employs different media access algorithms for transmitting small or large volumes of data: referred to as ″Instant Messages" and ″Burst data", respectively. For this work, we focus on the first type of transmission. This protocol enables instant bidirectional exchange of short messages up to 64 B, which can be sent immediately or scheduled to start at a predefined time. S2C devices are controlled by a specific set of commands called ″AT commands", short text strings that instructs the device to perform an action. In the DESERT protocol stack, the `uwEvologicsS2CModem` module

is responsible for sending and receiving these commands via TCP port 9200.

In a simulated environment, the transmission duration is easily computed by the physical module, given the bitrate. However, in a real-world experiment, the driver module must be properly configured. Upon reception of a instant message, S2C modems, output a string including also the transmission duration in microseconds, calculated by dividing the packet size by the bitrate. We made a table containing all these values for a packet size ranging from 20 to 64 B, as shown in Figure 3.5a. As we can see, the relationship between the size of a packet and the duration is not properly linear, possibly due to some form of block coding. Since we could not establish a direct mapping between the two, the simplest solution was to create a lookup table containing these values. To ensure the driver modules have information about transmission duration, we implemented the possibility to load a lookup table mapping packet sizes to duration. Moreover, we need to account for processing delays. As the modems do not provide specific information on this delay, we conducted the following study using two S2C modems. For each packet size, we send an instant message (without DESERT interposed) containing the transmission timestamp. On the receiving side, we subtract the received timestamp and the known duration from the reception timestamp to calculate the processing delay. By averaging over one hundred transmission for each packet size, we obtained the results shown in Figure 3.5b. These processing delay values are also included in the lookup table and are added to the transmission duration obtained earlier. It is important to note that these measurements were carried out by connecting the two nodes and the machine sending the ″AT commands" to the modems via Ethernet, rather than WiFi, to minimize latency and jitter.



(a) Transmission duration in milliseconds against packet size in bytes for S2C modems.



(b) Measured processing delay in milliseconds (with 95% confidence interval for the mean) against packet size in bytes for S2C modems.

Figure 3.5: Plots of the transmission duration and processing delay of S2C modems.

3.3.2 SuM

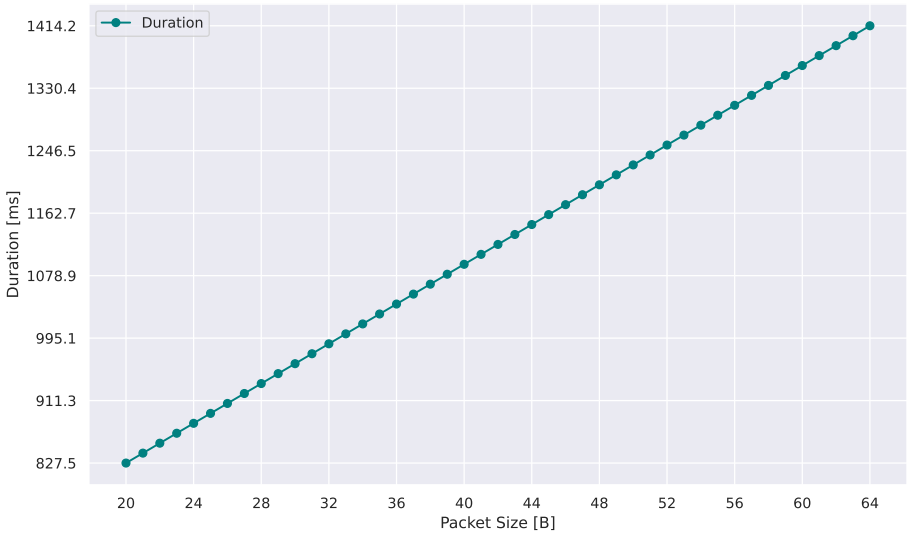
The SuM [17] is a low-cost acoustic modem, whose initial design was developed in the context of the Italian PNRM MODA project [27, 28]. It is composed of (i) a Raspberry Pi 4B processing board; (ii) a HiFiBerry DAC+ADC Pro 192 kHz analog/digital converter and (iii) the SuM analog frontend. The analog frontend, developed by SubSeaPulse SRL, includes the circuitry required for the switching between transmission and reception of acoustic signals. It pre-amplifies the received signal and amplifies the output signal to be transmitted. It supports a transmission frequency of up to 70 kHz, depending on the used transducer. The SuM has been tested to communicate over several hundreds of meters with a low-cost high-frequency transducer or up to a few kilometers with a medium-frequency transducer [29].

The SuM has been developed with a modular approach, consisting mainly of a physical and a data link layer. A very simple MAC module takes care of forming the data frames with source and destination addresses. The physical layer is responsible for encoding and modulating the data symbols, then sending the sampled symbols to the DAC during transmission. On reception, it receives samples from the ADC, demodulates the symbols, and decodes the packet. All the modem settings such as carrier frequency, Forward Error Correction (FEC), symbol duration and modulation, can be configured through a script available within the modems. Additionally, the modem software allows the user to connect to a TCP socket for sending and receiving data. A second TCP socket is used by the modem to exchange control information, facilitating its integration with the DESERT framework.

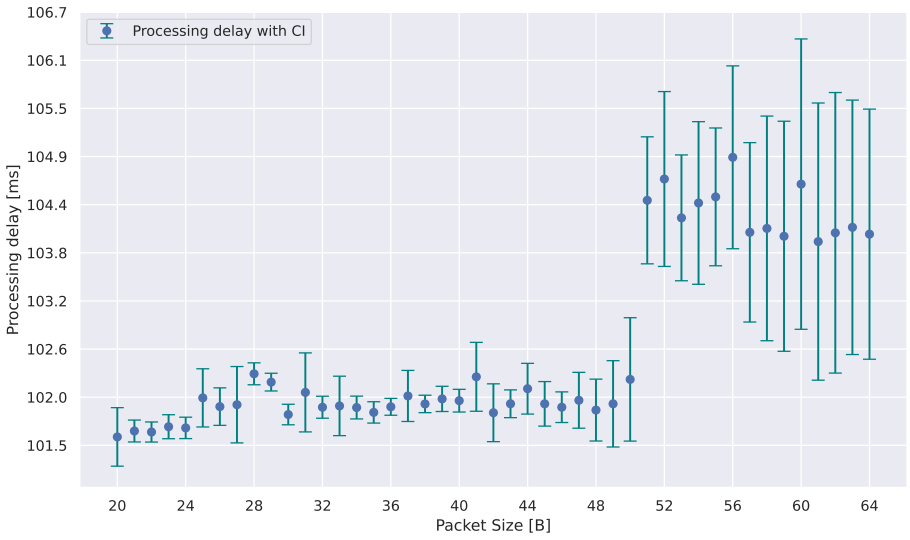
During transmission, the modems log the number of audio samples written by the selected physical protocol for a given packet. By dividing this value by the sampling frequency (default is 192 kHz), we can determine the transmission duration for that packet. Finally, by interpolating between the transmission duration for packets of 8 bytes and 64 bytes, we derived a linear formula to calculate the transmission duration based on packet size

$$t_{tx} = 0.56083 + 0.013 \cdot PacketSize. \tag{3.1}$$

We implemented this formula in the `uwMODAModem` module to compute the transmission duration of a packet. Figure 3.6a shows the values of transmission duration for packet sizes ranging from 20 to 64 B. Regarding the processing delay, the ″premodulation" option must be enabled on the modems. This option introduces a fixed delay of approximately 100 ms by scheduling the transmission with sufficient advance to ensure that the modulation process is complete and the signal samples are ready for the DAC. To show that this delay remains nearly constant, we averaged the delay over one hundred transmissions for packet size ranging from 20 B to 64 B. Each measurement is obtained by subtracting to the timestamp at which one modem begin receiving the timestamp when the other modem started transmitting the packet (which is part of the payload). As shown in Figure 3.5b, for a packet size between 20 B and 50 B, the processing delay remains consistently around 102 ms, on average. For a packet between 51 B to 64 B there is an average increase of approximately 2 ms, suggesting that a slightly longer ″premodulation" delay is required for larger packet sizes.



(a) Transmission duration in milliseconds against packet size in bytes for SuM modems.



(b) Measured processing delay in milliseconds (with 95% confidence interval for the mean) against packet size in bytes for SuM modems.

Figure 3.6: Plots of the transmission duration and processing delay of SuM modems.

4

Test Settings

To evaluate the performance of our system, we conducted three distinct experiments using DESERT. The first two experiments have the same network topology but utilize different modems, aiming to determine which one is better suited for our system's measurements.

Section 4.1 presents all the parameters used to configure the DESERT instances, which are the same for all experiments. It also provides an overview of the bits allocated to each protocol's packer for computing the total packet size. Section 4.2 outlines the topologies and configurations of the two preliminary experiments using the S2C and SuM modems, respectively, as well as the the calculation of additional delays introduced by the DESERT drivers. Lastly, Section 4.3 presents similar findings for a real-world experiment conducted on the Piovego river.

4.1 DESERT parameters

Unlike in simulation, for emulation or real-field testing, we need an instance of DESERT running for each node in the network. Each instance is marked with a unique identifier, which ranges from 1 to the total number of nodes. Each instance requires the IP address of the modem it connects to, as well as a TCP port number to establish the socket connection.

Even though our protocol does not require the nodes' system clocks to be synchronized, it is essential that all the DESERT instances start simultaneously and that the packet transmit ranging period is the same across all nodes, to avoid potential collisions. Also for this reason, the `uwRangingTDOA` module on each node has a starting delay, calculated by multiplying the node's `uwRangingTDOA` module ID by the starting delay value. For this work, we set the ranging period and the starting delay respectively to 15 s and 5 s.

Table 4.1 provides a summary of the DESERT parameters common to all the tests conducted for this work.

Table 4.1: DESERT parameters common to all tests

Parameter	**Value**
Number of nodes	3
Packet size	33 B
TCP port (S2C)	9200
TCP port (SuM)	55555
Ranging period	15 s
Starting delay	5 s

4.1.1 Packer settings

As already mentioned in Section 3.2, each layer of the protocol stack has a corresponding packer which maps its NS2 packets to binary strings. Figure 4.1 depicts the complete protocol stack with the associated packer modules, the AL and the driver modules. Additionally, Table 4.2 shows the number of bits needed for each packer module and the adaption layer.

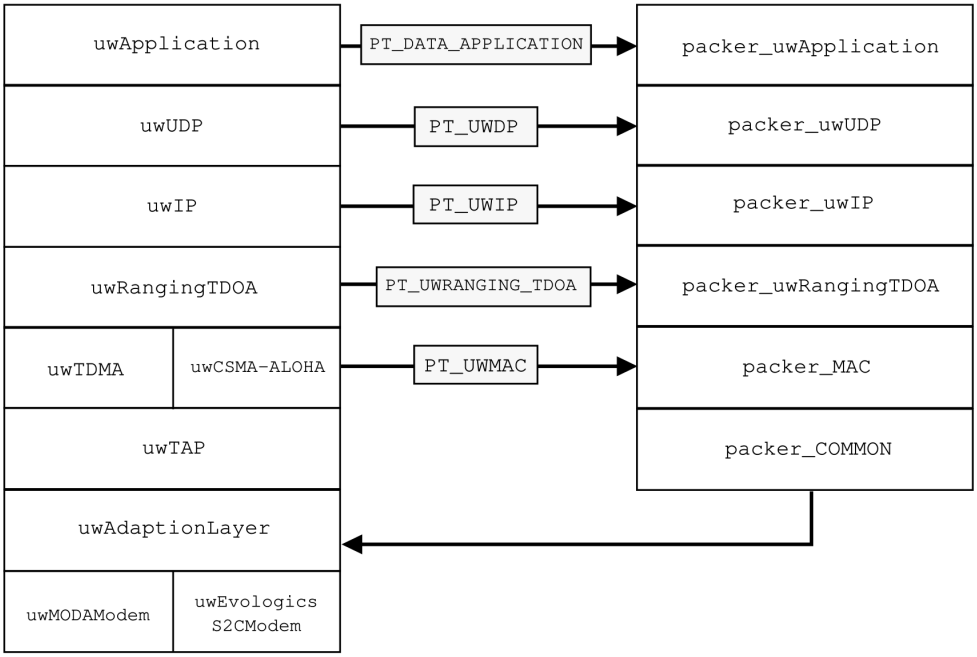


Figure 4.1: Packer protocol stack of each node in the network.

Note that, there is no packer for the `uwTAP` protocol being it transparent to the other layers in the stack. The `packer_COMMON` module is a `Packer` class responsible of mapping NS2 common header into a bit stream and vice-versa.

In addition to the bits listed in Table 4.2, we have to account for the variable size of the `holdover_list` field in the `PT_UWRANGING_TDOA` packets. Each element in this field is 48 bit long (Figure 3.3). Given that all tests in this work involve three nodes, which means two elements per `holdover_list` field, the total number of bits required is 261 bit.

This corresponds to a packet size of $\left\lceil \frac{261\,\text{bit}}{8} \right\rceil = 33\,\text{B}$.

Table 4.2: Number of bits assigned to each layer of the protocol stack

Module	**Bits**
packer_uwApplication	31
packer_uwUDP	4
packer_uwIP	16
packer_uwRangingTDOA	14
packer_uwMAC	16
packer_uwCommon	52
uwAdaptionLayer	32
Total	165

4.2 Preliminary tests

The first couple of tests is intended to demonstrate the functionality of the system we developed. For this purpose, we consider the simplest scenario: three nodes communicating over an acoustic channel in the air, at a distance of less than 1 m. To reduce latency and jitter typically introduced by wireless communications, all nodes are connected via Ethernet.

4.2.1 S2C setup

The setup for this experiment is shown in Figure 4.2. The test is carried out using the S2C modems, with each modem positioned so that the signal transmitted by one can be reliably received from the others. The modems are configured with a source level of 3 (the minimum source level) to ensure the lowest sound pressure level during transmission, along with a low input amplifier gain, which is recommended for short-distance testing.

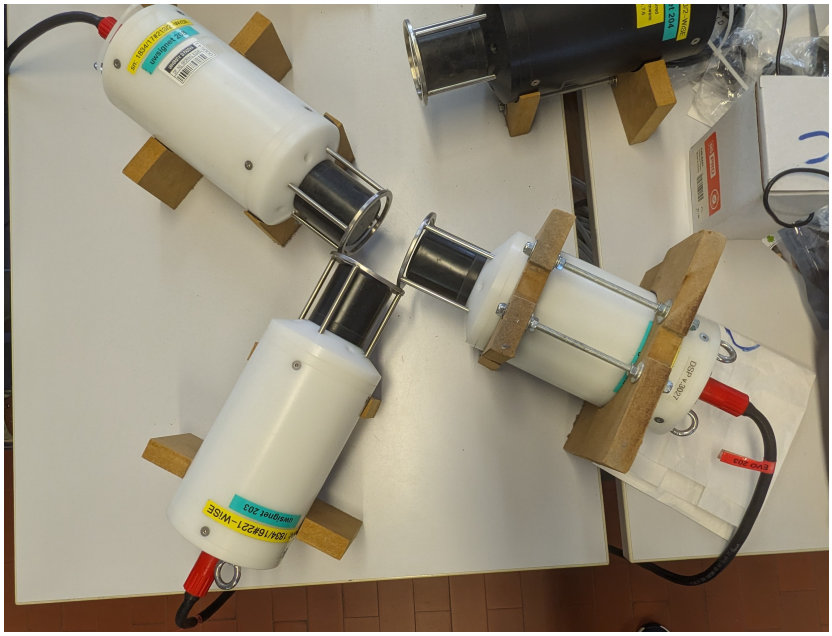


Figure 4.2: Setup with S2C modems, with each modem connected to a switch via ethernet.

In addition to the transmission duration and processing delay (discussed in Section 3.3), we also have to consider an additional delay introduced by the DESERT driver for the Evologics modems. Specifically, we observed a consistent error in the ToF calculations, resulting in an error of a few meters in the ranging measurements. By computing the Root Mean Square Error (RMSE) between these ranging measurements and the actual distance, which is approximately 0 m, and then dividing the value by the sound speed (assumed to be 343 m/s in air) and multiplying by 2, we determined that this delay, for the specific packet size used, corresponds to 15.9 ms.

4.2.2 SuM setup

Figure 4.3 shows the hardware setup for this experiment. The SuM analog frontend is removed and the HiFiBerry audio output is directly connected to a pair of speakers, while the input microphone is connected to a headset. We use a center frequency of 5 kHz to ensure that the low-cost microphones can reliably receive the transmitted signals. The system uses a BPSK modulation combined with a convolutional code with a code rate of 1/2 and constraint length 9, followed by an outer Reed-Solomon code with block length of 255, a message length 223 and an alphabet size of 256.

Figure 4.3: Setup using SuM modems, with each modem connected to a set of audio speakers for output, a headset for input and to a switch via Ethernet.

Also for this test, we have to account for the delay introduced by the DESERT driver module. By repeating the same process conducted in Section 4.2.1, we determined that this delay is equals to 33.8 ms.

4.3 Piovego river test setup

Figure 4.4 shows the area of Piovego river where the field experiment took place. During the test, we submerged three S2C modems in the water to a depth of 1 m, from the dock shown in the upper part of the picture. The modems are positioned in a straight line, spaced 2 m apart, resulting in a total distance of 4 m between the two modems at the ends. Testing at longer distances, with S2C modems, was not possible due to the unfavorable conditions of the Piovego river at the time when the test was conducted, which make it difficult for the modems to reliably receive signals. Consequently, the source level is increased from the minimum level of 3 to 1.

As in the preliminary tests, each modem and the computer running the DESERT instances are connected via Ethernet to a switch to minimize latency and jitter as much as possible. Finally, we computed the delay introduced by the DESERT driver, which differs slightly from the preliminary tests that used the sound speed in air. Given that the speed of sound underwater is approximately 1481 m/s, the resulting computed delay is 29.95 ms.

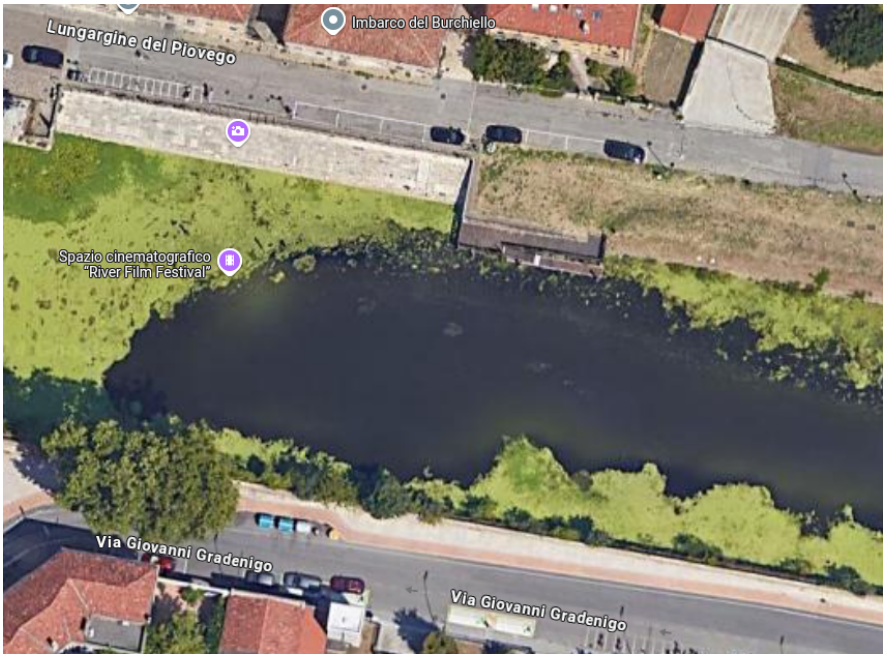


Figure 4.4: Real-world experiment on the Piovego river's dock using S2C modems.

5

Results

In this chapter, the system is evaluated through three distinct tests with similar topologies and two different MAC protocols.

Section 5.1 outlines the key metrics used to asses the system performance and provides a brief summary of the MAC protocols used. Section 5.2 and Section 5.3 present the results of two preliminary tests, using S2C and SuM modems respectively, to compare the performance of the two modems and determine the most suitable option for these measurements. Finally, Section 5.4 reports the findings from a real-word experiment conducted with SuM modems in the Piovego river.

In order to measure the performances of our system, for each node we compute the time series of the RMSE.

$$RMSE_n(t) = \sqrt{\frac{1}{D}\sum_{i=0}^{D-1}\left(d_{n,i}(t) - \widehat{d_{n,i}(t)}\right)^2}. \tag{5.1}$$

Here, $\widehat{d_{n,i}(t)}$ is the i-th distance estimated by node n at time t, while $d_{n,i}(t)$ is the actual distance. Each node calculates $D = N(N-1)/2$ distances, where N is the number of nodes in the network. Additionally, we are interested in how the ToFs vary over time, so for each node, we compute the mean jitter over the entire test period.

$$J_n = \frac{1}{T_n}\sum_{i=0}^{T_n-1}\left|ToF_{n,i}(t) - ToF_{n,i}(t-1)\right|. \tag{5.2}$$

Where T_n is the total number of ToFs recorded by node n and $ToF_{n,i}(t)$ is the i-th ToF measured by node n at time t. The ToFs are sampled at intervals of $t = 15$ s and converted to distances by multiplying them by the speed of sound, which are then compared to the actual distances. The following experiments span a duration of 600 seconds (10 minutes).

Although the `uwRangingTDOA` protocol is MAC agnostic, it is still influenced by the underlying MAC protocol. To investigate this, we compared two different MAC protocols under the same network configuration and channel conditions, per each test:

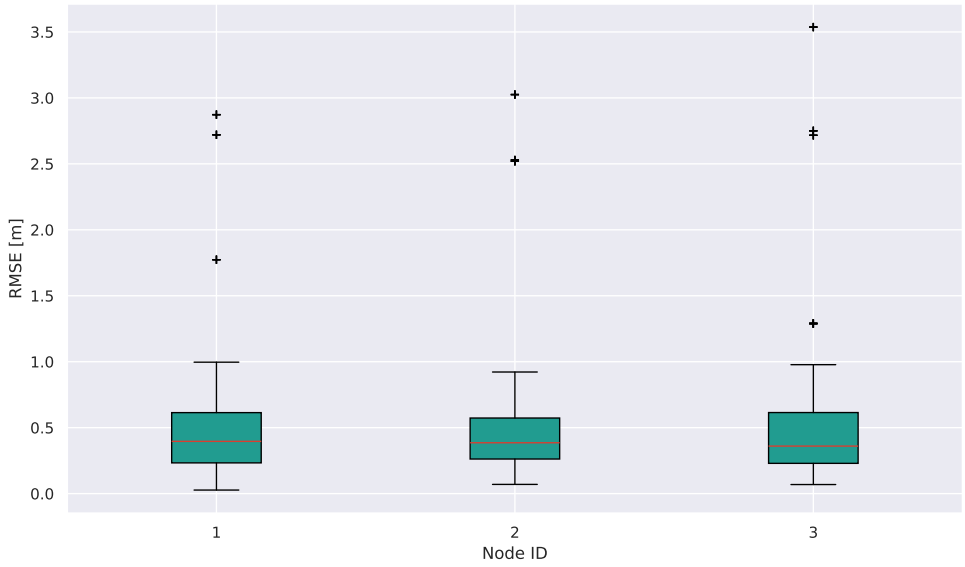
1. `TDMA` is a contention-free MAC protocol where the time is divided in time frames, each consisting of as many time slots as there are nodes, allowing only one node to transmit per slot. In particular, we set a guard time of 1 s to ensure to minimize packet collisions and configured the frame duration to match to the ranging period of 15 s;
2. `CSMA-ALOHA` is a carrier-sense contention-based MAC. Before transmiting a packet, it senses the channel: if it is busy, the node waits for a random time before retrying. The channel listening time is uniformly distributed between 0 and 0.5 s plus an additional 0.1 s constant. The maximum number of transmission attempts is set to 5.

Figure 5.1 shows the box plots of the RMSE measured in a network with three nodes using S2C modems placed less than 1 m apart. The sound speed used in these calculations is assumed to be constant at 343 m/s. Figure 5.1a and Figure 5.1b illustrate the results for the `TDMA` and `CSMA-ALOHA` MAC protocols, respectively. With `TDMA`, the average RMSE for all nodes is less than 1 m, though some outliers appear, possibly due to collisions or fluctuations in the processing delay. In contrast, the contention-based MAC protocol, `CSMA-ALOHA`, has a higher probability of collisions, leading to more outliers, with some exceeding 5 m. The average RMSE values measured are slightly higher than the ones obtained with `TDMA`, but are still generally below 1 m.

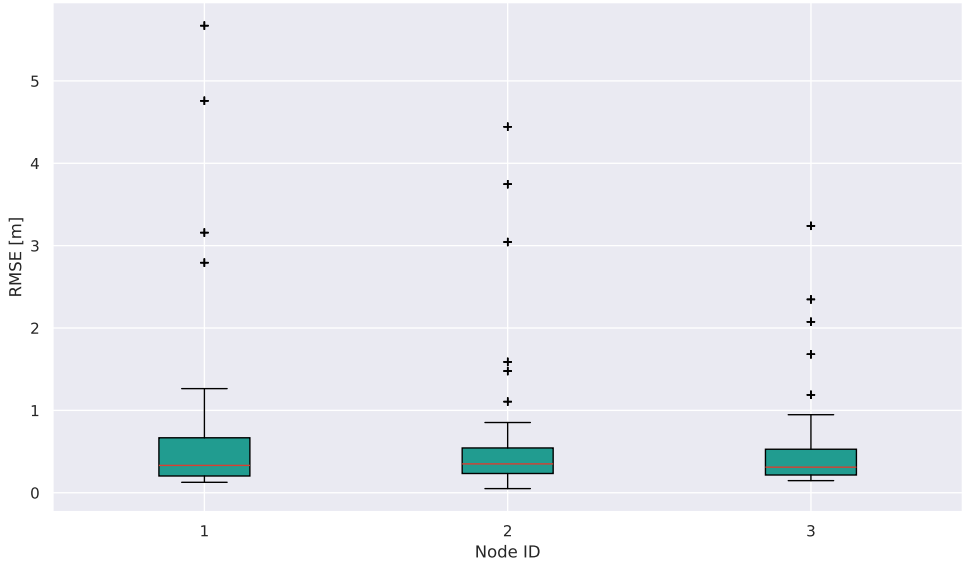
Figure 5.2 displays the time evolution of the ToFs for each node in the network. With both MAC protocols, on average the ToFs remain below 2 ms. It is important to note that at each time step, each node measures $D = 3$ ToFs. The impact of collisions is more evident in these plots, particularly with the `CSMA-ALOHA` protocol, where some ToFs exceed 20 ms. The mean jitter is computed as $J = \frac{1}{N}\sum_{n=1}^{N} J_n$ with $N = 3$. The shaded region around the mean jitter represents the 95% confidence interval for the mean, determined by averaging the confidence intervals of each J_n, that is $CI_n = 1.96 \cdot \sigma_{J_n}/\sqrt{T_n}$. Where σ_{J_n} is the standard deviation of J_n.

Using `TDMA`, the mean jitter is 1.57 ms, as shown in Figure 5.2a, with a standard deviation of 2.17 ms. In comparison, with `CSMA-ALOHA`, the mean jitter slightly increases to 1.95 ms, as illustrated in Figure 5.2b, and the standard deviation rises to 3.52 ms.

However, more precise results could be achieved with these devices. As noted in Section 3.3, "Instant Messages" can be scheduled to start at specific times. This capability would enable very accurate processing delay measurements and, consequently, more consistent ToFs measurements. Implementing this would involve additional work on integrating DESERT with the S2C modems, which is beyond the scope of this thesis.

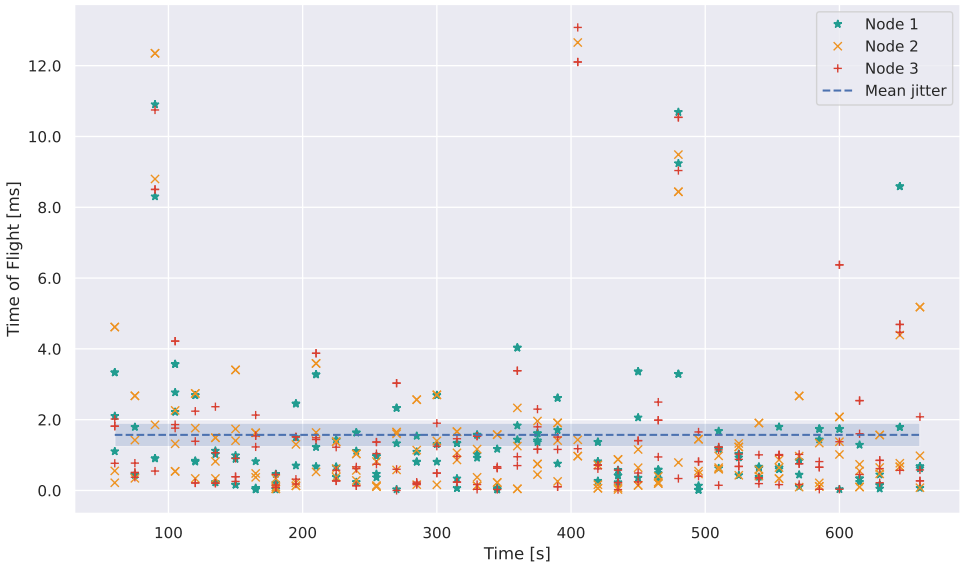


(a) Box plots of the RMSE measured with TDMA.

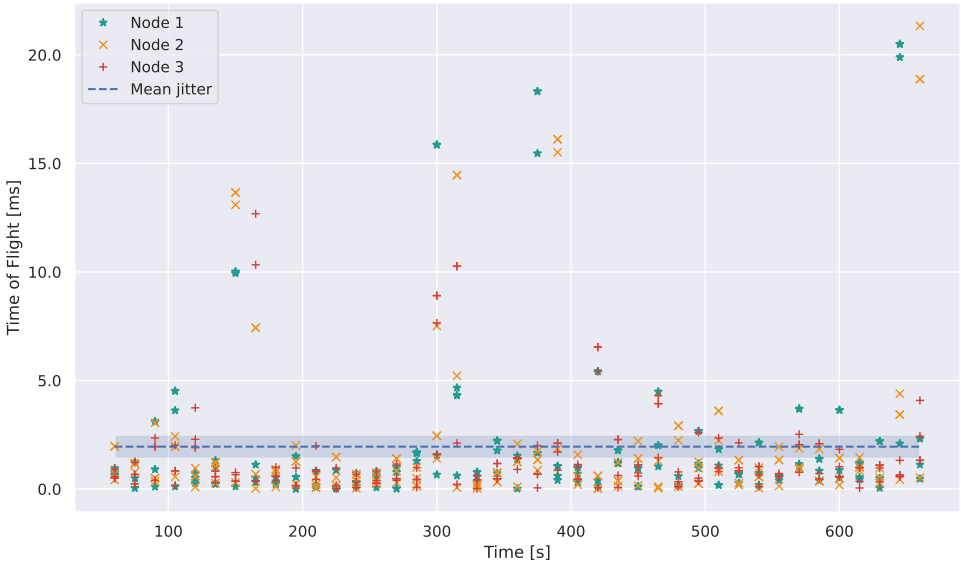


(b) Box plots of the RMSE measured with CSMA−ALOHA.

Figure 5.1: Box plots of the RMSE measured by each node in the network using S2C modems.



(a) ToFs against test duration time and mean jitter with TDMA.



(b) ToFs against test duration time and mean jitter with CSMA−ALOHA.

Figure 5.2: Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using S2C modems. The shaded area represents the 95% confidence interval for the mean.

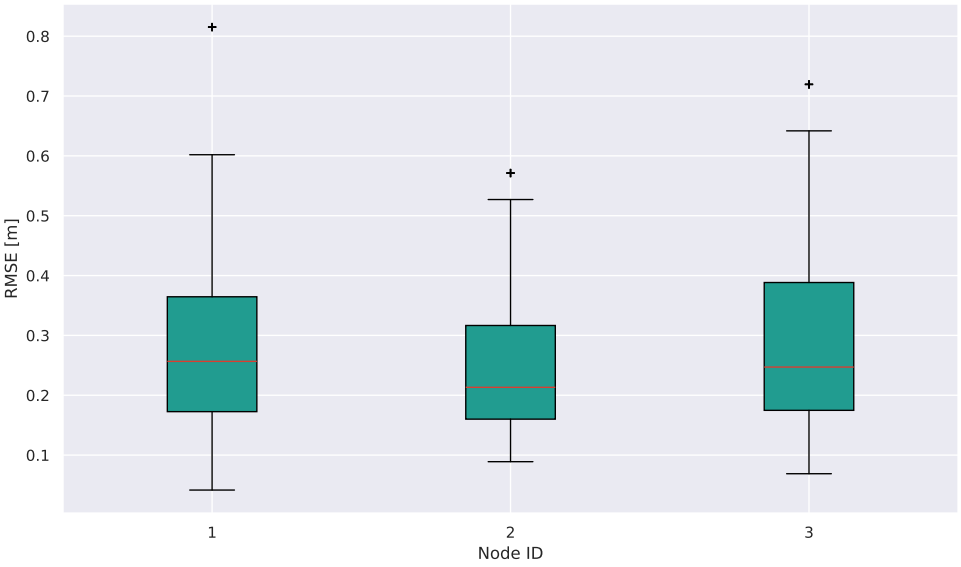
5.3 SuM test results

Figure 5.3 presents box plots of the RMSE measured in a network with three nodes using SuM modems placed less than 1 m apart. The sound speed used for these calculations is assumed to be constant at 343 m/s. Figure 5.3a and Figure 5.3b display the results for the `TDMA` and `CSMA-ALOHA` MAC protocols, respectively.

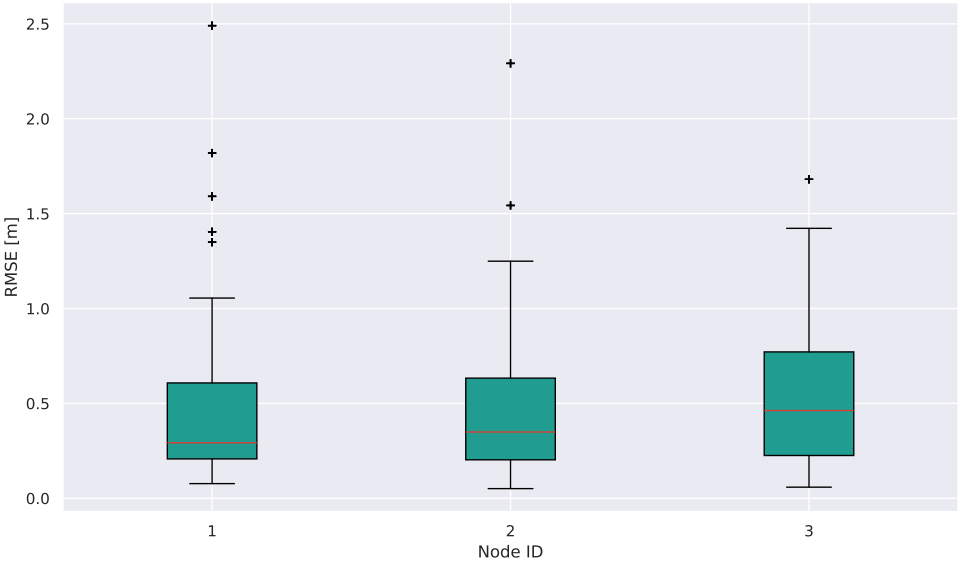
Remarkably, with `TDMA`, the average RMSE is below 0.7 for all nodes expect for a single outlier in Node 1 and Node 2. The results using `CSMA-ALOHA` are similar, although there are more outliers, highlighting the impact of collisions on the `uwRangingTDOA` module. Despite this, the performance with SuM modems is better than with S2C modems using `TDMA`, likely due to the pre-modulation feature of the SuM modems, which provides a nearly constant processing delay.

The advantages of using pre-modulation are particularly evident in the the time evolution of the ToFs for each node in the network, as shown in Figure 5.4. In this experiment, the measured ToFs, on average, remain below 2 ms, with collisions having a lesser impact compared to S2C modems. Even with `CSMA-ALOHA`, the ToFs do not exceed 10 ms, and they stay below 3.5 ms with `TDMA`. This suggests that a contention-free MAC protocol may be likely the most suitable for our protocol.

The mean jitter further supports this obeservation, with a value of 0.63 ms and a standard deviation of 0.53 ms, as shown in Figure 5.4a. Figure 5.4b, shows that, even with `CSMA-ALOHA`, the jitter remains lower than that observed with S2C modems, with a mean jitter of 1.39 ms with a standard deviation of 1.53 ms.

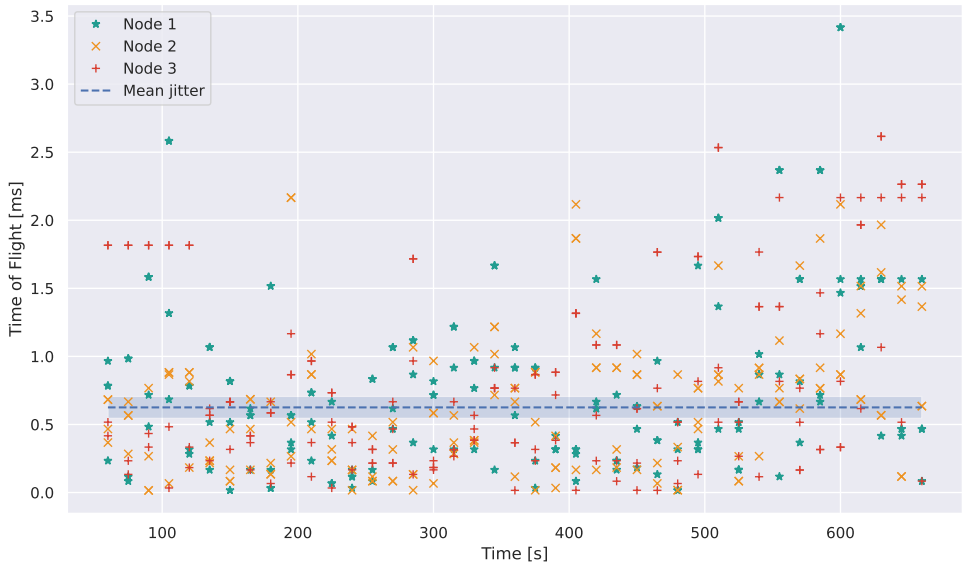


(a) Boxplots of the RMSE measured with TDMA.

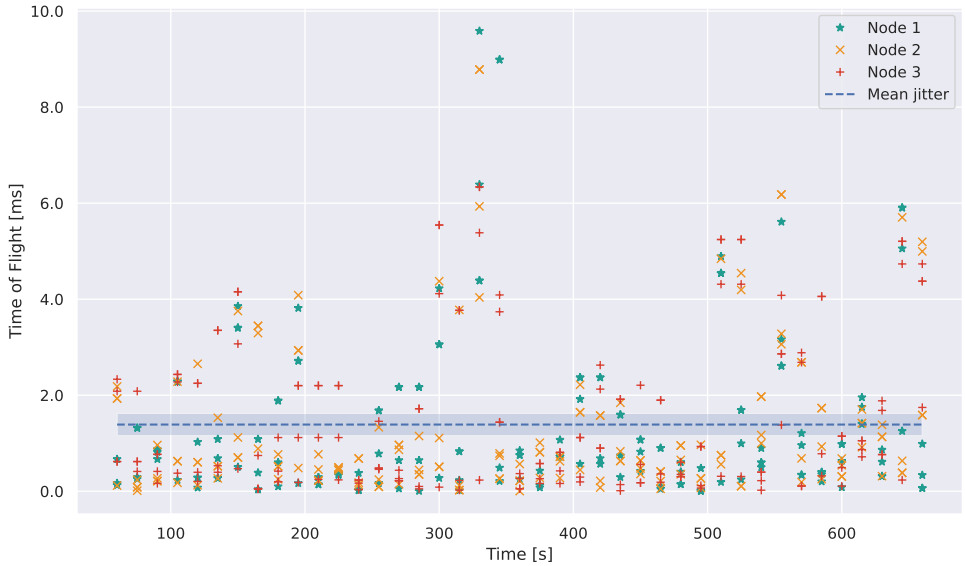


(b) Boxplots of the RMSE measured with CSMA−ALOHA.

Figure 5.3: Boxplots of the RMSE measured by each node in the network using SuM modems.



(a) ToFs against test duration time and mean jitter with TDMA.



(b) ToFs against test duration time and mean jitter with CSMA−ALOHA.

Figure 5.4: Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using SuM modems. The shaded area represents the 95% confidence interval for the mean.

Although the preliminary tests yielded better results both in terms of RMSE and jitter, using SuM modems in a real-world setting did not produce satisfactory outcomes. Specifically, with three nodes placed in a straight line, spaced 7 m apart, the system inaccurately measured the distance between all nodes as 7 m. This discrepancy is not inherently due to the modems themselves, but rather to the settings used, which may require further fine-tuning and testing on more favorable environments. However, exploring this issue further is outside the scope of this work.

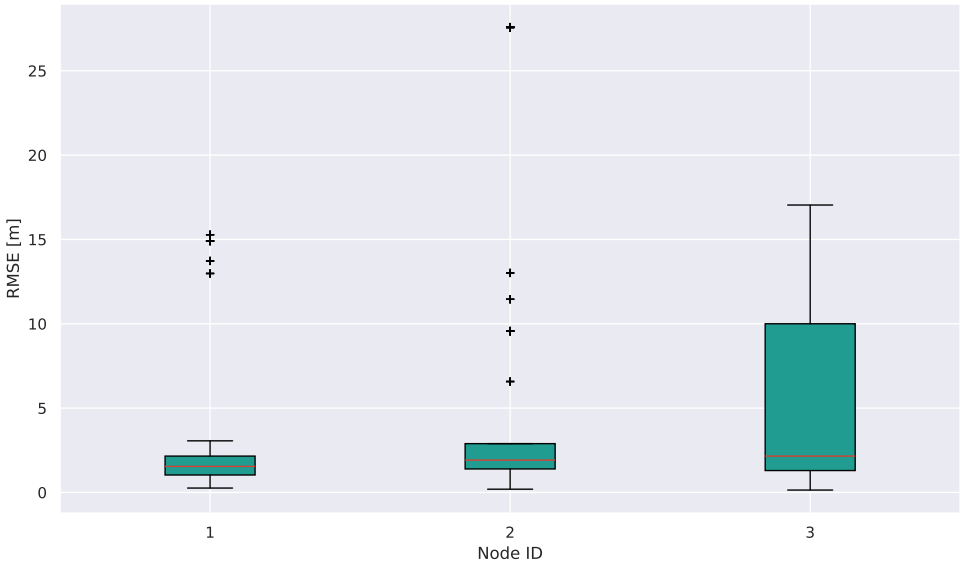
As described in Section 4.3, the experiment was carried out using three S2C modems. The results obtained are presented in Figure 5.5 and Figure 5.6. It is important to note that the left-most modem (Node 2) is positioned approximately 2 m from the center modem (Node 1), which is also about 2 m from the right-most modem (Node 3). For these measurements, the speed of sound is assumed to be constant at 1481 m/s.

Figure 5.5a presents the box plots of the RMSE obtained using `TDMA`. The average error remains below 2 m, although the measurements from Node 3 extend to an error of 15 m. This highlights the asymmetry of the channel, as the distances measured between the first two nodes and Node 3 changes significantly. It is very likely that the placement of Node 3 was not optimal, causing it to experience greater multi-path fading compared to the other two nodes. On the other hand, with `CSMA-ALOHA`, the RMSE shows a slight increase for Node 2 and Node 1, as depicted in Figure 5.5b. However, there is a minor improvement for Node 3.

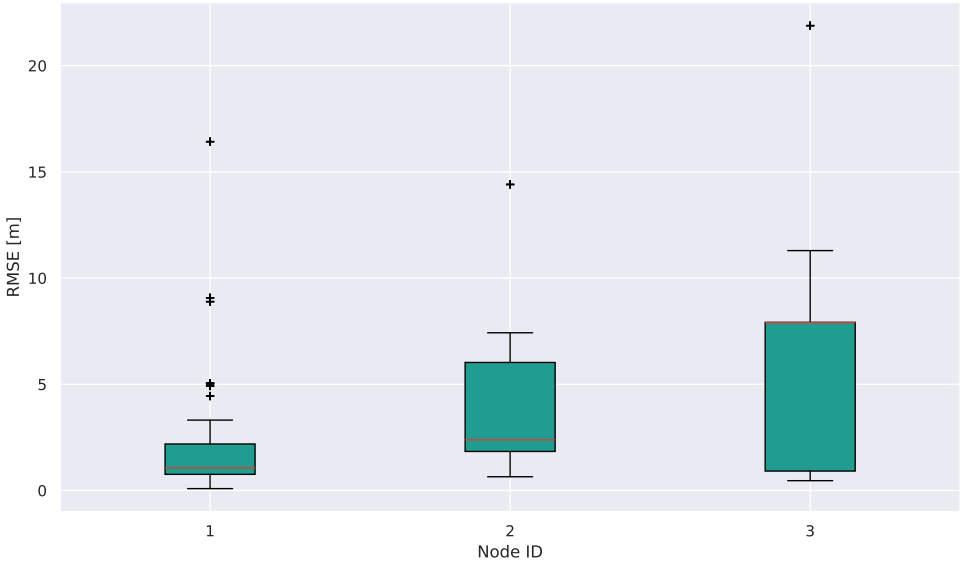
Figure 5.6 presents the result in terms of the time evolution of the ToFs and their mean jitter. It is important to note that the mean jitter improved when using both MAC protocol compared to the preliminary tests, especially when using `CSMA-ALOHA`. This improvement may be attributed to the fact that S2C modems are designed specifically for underwater communications. The mean jitter obtained with `TDMA` (Figure 5.6a) is on average 1.49 ms with a

standard deviation of 3.69 ms. It is worth mentioning that, additional noise from two swans moving rapidly near the test area contributed to the already harsh conditions of the channel during this part of the test.

In contrast, the mean jitter obtained with `CSMA-ALOHA` (Figure 5.6b), is surprisingly smaller than that of `TDMA`, measuring an average of 0.48 ms with a standard deviation of 1.38 ms. Note that, this contrasts with what might be deduced from only examining the preliminary tests.

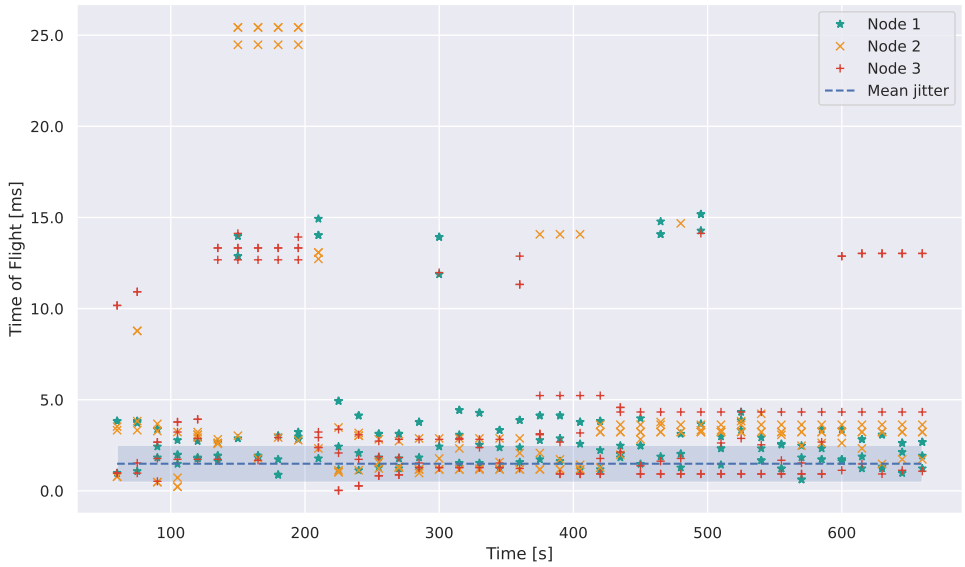


(a) Boxplots of the RMSE measured with TDMA.

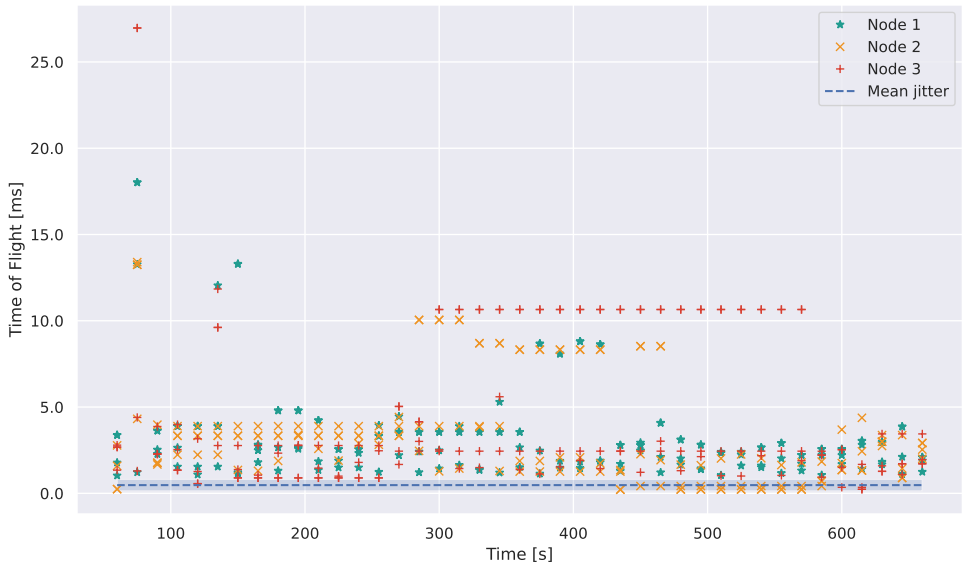


(b) Boxplots of the RMSE measured with CSMA−ALOHA.

Figure 5.5: Boxplots of the RMSE measured by each node in the network using S2C modems in the Piovego river.



(a) ToFs against test duration time and mean jitter with TDMA.



(b) ToFs against test duration time and mean jitter with CSMA−ALOHA.

Figure 5.6: Plots of the ToFs (milliseconds) for each node against the test duration time (seconds) and the mean jitter using S2C modems in the Piovego river. The shaded area represents the 95% confidence interval for the mean.

While the first two preliminary tests help demonstrate the usability of the `uwRangingTDOA` module on real modems, this experiment is particularly significant as it shows that in a real-world test environment with harsh conditions, the choice of MAC protocol plays an important role. This highlights the importance of having a module capable of working with different MAC protocols, as in certain scenarios, a contention-free approach may be preferred over a contention-based one, or vice versa.

Furthermore, it highlights how, despite optimal results in controlled test environments, real-world experiments may yield different results, emphasizing the challenges of working with underwater acoustic channels.

6

Conclusions and Future Work

In this paper, we presented a MAC-agnostic distributed ranging protocol, implemented as a module of the DESERT underwater framework. Building on the work presented in [1] using simulated scenarios, we extended the reserarch by testing the `uwRangingTDOA` module on actual modems in real-word experiments, basically removing the simulated physical layer and using a real one. Specifically, we first evaluated the module on Evologics S2C R 18/34 WiSe modems, followed by tests on a low-cost modem developed by the Signet lab at the University of Padova, known as the SuM. Notably, the drivers for both modems are already integrated into DESERT, facilitating our work, although minor modifications to the drivers were required to meet our specific needs.

We tested the protocol in three different experiments with similar topologies, with the first two preliminary aimed at demonstrating the feasibility of the system. The final experiments was a real-wolrd field test conducted on the Piovego river. This experiment, in particular, highlighted the need for further field testing with different configurations of the

SuM modems. The limitations observed are not inherent to the SuM modems themselves, which had initially shown more promising results than the Evologics modems during preliminary tests. The SuM modem provided more accurate results in terms of jitter in the measurements (due to the pre-modulation option), a critical factor for the performance of the `uwRangingTDOA` module.

Each test was conducted twice, using two different MAC protocols: a contention-based protocol (`CSMA-ALOHA`) and a contention-free protocol (`TDMA`). This allowed us to demonstrate the advantage of being able to switch between MAC protocols, as the choice between them can be crucial depending on the specific scenario, with one protocol being more suitable than the other.

Overall, this work represents just the starting point for extensive research in this area. Future works will focus on refining the SuM modems for use in field experiments to potentially achieve more accurate results, as well as testing over longer distances between nodes. Additional field experiments will be carried out to evaluate the module's performance in real-world experiments. Furthermore, we plan to investigate the possibility of reducing the packet payload size in the network. One promising approach explored in [1] involves organizing packet entries to minimize the age of information while considering channel conditions to maximize the goodput of the network. In addition to improving ranging precision, these improvements are expected to also benefit the energy efficiency, which is also a key concern in AUV swarm scenarios.

References

[1] A. Montanari, F. Campagnaro, and M. Zorzi, "An adaptive mac-agnostic ranging scheme for underwater acoustic mobile networks," *Comput. Netw.*, vol. 247, no. C, jul 2024. [Online]. Available: https://doi.org/10.1016/j.comnet.2024.110430

[2] "DESERT: Design, simulate, emulate and realize test-beds for underwater network protocols," Last time accessed: Apr. 2024. [Online]. Available: https://desert-underwater.dei.unipd.it/

[3] "EvoLogics S2C R 18/34D Underwater Acoustic Modem," Last time accessed: Apr. 2024. [Online]. Available: https://evologics.de/product/s2c-r-18-34d-27

[4] F. Campagnaro, R. Francescon, E. Coccolo, A. Montanari, and M. Zorzi, "A software-defined underwater acoustic modem for everyone: Design and evaluation," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 102–107, 2023.

[5] ecoSUB Robotics Ltd, "ecoSUBμ5 - 500 m rated Micro-AUV," 2023. [Online]. Available: https://www.ecosub.uk/ecosubu5---500-m-rated-micro-auv.html

[6] K. Casey, A. Lim, and G. Dozier, "A sensor network architecture for tsunami detection and response," *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, pp. 27–42, Jan. 2008.

[7] F. Mason *et al.*, "Automatic shark detection via underwater acoustic sensing," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 18–23, 2022.

[8] M. Waldmeyer, H.-P. Tan, and W. K. Seah, "Multi-stage AUV-aided localization for underwater wireless sensor networks," in *IEEE Workshops of International Conference on Advanced Information Networking and Applications*, Biopolis, Singapore, Mar. 2011, pp. 908–913.

[9] S. Hożyń, "A review of underwater mine detection and classification in sonar imagery," *Electronics*, vol. 10, no. 23, pp. 1–22, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/23/2943

[10] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater surveillance systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, Jul. 2006.

[11] F. Mason *et al.*, "Low-cost AUV swarm localization through multimodal underwater acoustic networks," in *Proc. MTS/IEEE Global OCEANS*. Virtual, Singapore – U.S. Gulf Coast: IEEE, Sep. 2020, pp. 1–7.

[12] S. A. H. Mohsan, Y. Li, M. Sadiq, J. Liang, and M. A. Khan, "Recent advances, future trends, applications and challenges of internet of underwater things (iout): A comprehensive review," *Journal of Marine Science and Engineering*, vol. 11, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/1/124

[13] T. C. Furfaro and J. Alves, "An application of distributed long baseline — node ranging in an underwater network," in *2014 Underwater Communications and Networking (UComms)*, 2014, pp. 1–5.

[14] O. Kebkal, K. Kebkal, and R. Bannasch, "Long-baseline hydro-acoustic positioning using d-mac communication protocol," in *2012 Oceans - Yeosu*, 2012, pp. 1–7.

[15] K. Kebkal, O. Kebkal, I. Glushko, V. Kebkal, L. Sebastiao, A. Pascoal, J. Gomes, J. Ribeiro, S. H., and M. Ribeiro, "Underwater acoustic modems with integrated atomic clocks for one-way travel-time underwater vehicle positioning," 09 2017.

[16] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. Ucomms*, Lerici, Italy, Sep. 2016.

[17] A. Montanari, V. Cimino, D. Spinosa, F. Donegà, F. Marin, F. Campagnaro, and M. Zorzi, "Psk modulation for underwater communication and one-way travel-time ranging with the low-cost subsea software-defined acoustic modem," pp. 1–8, 2024.

[18] A. Song, M. Stojanovic, and M. Chitre, "Editorial: Underwater acoustic communications: Where we stand and what is next?" *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, Jan. 2019.

[19] H.-P. Tan, R. Diamant, W. K. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14, pp. 1663–1676, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801811001624

[20] F. Campagnaro, F. Steinmetz, and B.-C. Renner, "Survey on low-cost underwater sensor networks: From niche applications to everyday use," *Journal of Marine Science and Engineering*, vol. 11, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/1/125

[21] G. Schirripa Spagnolo, L. Cozzella, and F. Leccese, "Underwater optical wireless communications: Overview," *Sensors*, vol. 20, no. 8, p. 2261, Apr. 2020.

[22] A. Signori, F. Campagnaro, and M. Zorzi, "Modeling the performance of optical modems in the desert underwater network simulator," in *2018 Fourth Underwater Communications and Networking Conference (UComms)*, 2018, pp. 1–5.

[23] S. M. Maher, Z. M. Ali, H. H. Mahmoud, S. O. Abdellatif, and M. M. Abdellatif, "Performance of RF underwater communications operating at 433 MHz and 2.4

GHz," in *International Conference on Innovative Trends in Computer Engineering (ITCE)*, Feb. 2019, pp. 334–339.

[24] Y.-S. Ryuh, G.-H. Yang, J. Liu, and H. Hu, "A school of robotic fish for mariculture monitoring in the sea coast," *Journal of Bionic Engineering*, vol. 12, no. 1, pp. 37–46, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1672652914600986

[25] G. Ferri, R. Petroccia, T. Fabbri, A. Faggiani, and A. Tesei, "A network navigation system with opportunistic use of one-way and two-way acoustic ranging: the dans20 experience," in *OCEANS 2021: San Diego – Porto*, 2021, pp. 1–8.

[26] H. Thomas, "Gib buoys: an interface between space and depths of the oceans," in *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)*, 1998, pp. 181–184.

[27] E. Coccolo, F. Campagnaro, D. Tronchin, A. Montanari, R. Francescon, L. Vangelista, and M. Zorzi, "Underwater acoustic modem for a morphing distributed autonomous underwater vehicle (moda)," *OCEANS 2022 - Chennai*, pp. 1–8, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248922167

[28] ——, "Underwater acoustic modem for a morphing distributed autonomous underwater vehicle (moda)," in *OCEANS 2022 - Chennai*, 2022, pp. 1–8.

[29] A. Montanari, F. Marin, V. Cimino, D. Spinosa, F. Donegà, D. Cosimo, L. Bazzarello, D. Natale, F. Campagnaro, and M. Zorzi, "Experimenting various janus frequency bands with the subsea software-defined acoustic modem," pp. 1–5, 2024.

Acknowledgments

I would like to express my gratitude to Filippo Campagnaro who always believed in me and was there to help. I also want to deeply thank Filippo Donegà, Antonio Montanari and Roberto Francescon for the help they provided with the field tests. My sincerest thanks should also go to my dear friends, Agnese and Francesco, for always listening to my daily complains and helping me through difficult times.