Software Defined Underwater Communications: an Experimental Platform for Research

Roberto Francescon[‡], Filippo Campagnaro[‡], Federico Favaro[‡], Federico Guerra[‡], Michele Zorzi[‡] *

[‡] Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy,

* Wireless and More srl, piazza Luigi da Porto 20, 35131 Padova, Italy

[‡]{frances1,campagn1,favarof,zorzi}@dei.unipd.it, federico@guerra-tlc.com

*michele.zorzi@wirelessandmore.it

***Abstract*—Underwater Acoustic Networks have become widely adopted in many marine and underwater contexts thanks to the increased availability of devices and technologies over the years. Even though the number of Commercial Off-the-Shelf devices has increased in the last decade, underwater technology still remains a very expensive asset to acquire. Moreover, the technology is almost always licensed and inaccessible to the user, to protect intellectual property and industrial patents. This last point is often a problem for the researchers that do not intend to make profit from algorithms or patented ideas but instead leverage on the existing technology to install and test their own solutions to advance the state of the art. Also, researchers very often resort to simulation and emulation to test and evaluate their algorithms: this approach not only guarantees freedom of implementation but also avoids the issues of organizing a field trial in the ocean or at sea, and is definitely cheaper. However, to prove the effectiveness of the proposed solutions in real environments, simulation may not be sufficient. In this paper we present a complete tool for testing and evaluating the full protocol stack, from application to physical layer, with both simulation and real field experimentation, that integrates the open source DESERT Underwater framework and a low cost software-defined acoustic modem.**

I. Introduction

Underwater acoustic modems have witnessed continuous and active development from their first commercial versions, such as the WHOI Micromodem [1], until today [2], [3]. Underwater Acoustic Networks (UANs) have widely proven their effectiveness in many fields: coastal monitoring, ocean rescue, underwater facilities maintenance, maritime defense, and many others. Nowadays, there is a wide availability of modems for underwater acoustic communications, covering a large span of applications and scenarios [4]–[6]. As a result, applications are emerging at a fast pace, putting new requirements on underwater communications, and leading to the development of optical and magneto-inductive underwater modems, that can reach higher data rates, and of Software Defined Modems (SDMs), that allow a higher degree of customization and adaptability. The quest for optimal performance in the various application-driven scenarios led to a vast body of research, prototypes and Commercial Off-the-Shelf (COTS) units: as examples, channel coding techniques were widely evaluated in [7], Orthogonal Frequency Division Multiplexing (OFDM) for underwater acoustics is explored in [8], and Multiple-Input Multiple-Output (MIMO) underwater communication as a channel access technique is investigated in [9].

R. Francescon, corresponding author (email: frances1@dei.unipd.it), is with the Department of Information Engineering, University of Padova, Italy

This work has been partially supported by European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE0000001 - program “RESTART”) and the FSE REACT EU, PON Research and Innovation 2014-2020 (DM 1062/2021).

Over the years, the need for re-programmable and reconfigurable platforms for underwater communications became more and more apparent: to this aim, SDMs started to be a preferred architecture. Dol *et al.* [10] picture the state-of-the-art of a few years ago, highlighting the features and the drawbacks of each platform. Specifically, the NILUS platform proved to be a versatile and effective programmable platform on which both the physical and network layers could be programmed but whose size made it difficult to deploy. Subnero, from the first prototypes, has developed advanced units and has been selling an SDM that can be easily configured through the UNET stack software [11], [12]. In [13] a complete framework for the joint optimization of device and network is presented. In [14] the NATO CMRE STO developed the concept of Software Defined Open Architecture Modem (SDOAM), generalizing the structure of an underwater communication architecture to a novel communication stack.

The concepts of SDMs and Software Defined Networks (SDNs) have recently gained increased interest in the underwater communications community thanks to its flexibility and configurability [13]: the major beneficiaries are Universities and research groups that are able to tailor prototypes and installations to the needs of the current experiment or research, but other players may also benefit from this paradigm shift. On the other hand, the fact that underwater modems are heavily dependent on company-owned knowledge and patents and are very application-oriented, along with the absence of well-established and shared protocols, contributed to an environment where device interoperability is virtually absent. In addition, even in the variety-rich environment of modern underwater modems, the costs for obtaining a few units remain an obstacle for many research groups.

In this paper we present a novel and inexpensive software and hardware architecture that allows great flexibility and reconfigurability of UANs. The proposed platform is composed by two main components: an SDM the University of

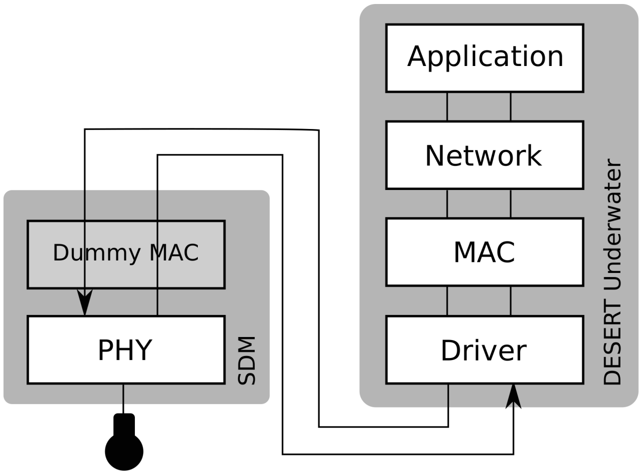

Fig. 1. SDM and DESERT protocol stacks.

Padova has recently developed, and a software protocol stack, implemented through the DESERT Underwater simulator. The resulting Software Defined Platform can be used to implement and test protocols in simulation, through DESERT and then, once ready, to test and evaluate these protocols in real field-trials, by connecting the SDM as the physical layer of DESERT. We also showcase for the first time this Software Defined Platform in a test involving a meta-layer able to optimize the transmission by changing the physical settings in the SDM, such as transmission gain and coding scheme. This optimization is based on physical parameters such as the Received Signal Strength Indicator (RSSI) and on statistical ones, such as the Packet Delivery Ratio (PDR). In detail, in Section II we illustrate the components of the SDM both hardware and software while in Section III we introduce DESERT Underwater, and the components that allow to connect it to the software-defined modem are explained. In Section IV we show how the connection of the two previous components results in the complete experimental platform while in Section V we give the details of the demonstration we performed to showcase its the capabilities. Finally, in Section VI we give some concluding remarks, highlighting the future works.

II. Software Defined Modem

Recently, the University of Padova has developed an SDM for high-frequency UANs in shallow water that operates in the medium range [15]. The modem is still a prototype and was developed in the context of the MODA project funded by the Italian Navy, but has been already tested in the shallow waters of a river nearby the University, proving capabilities that exceed the requirements of the project [16]. The MODA SDM is written in C++ and uses the *liquid-dsp* library [17] to perform encoding and modulation. Its structure relies on a 2-layer stack: the physical layer, that includes a Data AcQuisition (DAQ) object and a Digital Signal Processing (DSP) object to physically interact with the audio interface and modulate the waveform, and a Medium Access Control (MAC) layer, that includes a Logical Link Control (LLC)

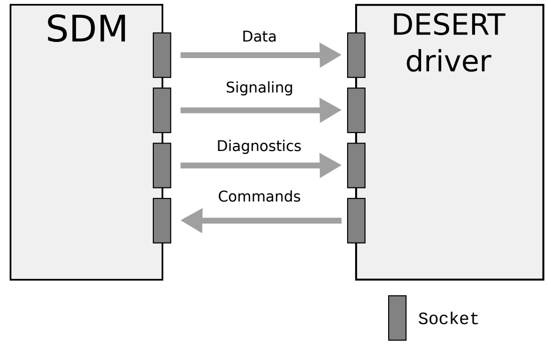

Fig. 2. Socket connections for Driver/SDM interaction.

sub-layer, responsible for segmentation and reassembly. The hardware is composed of a Raspberry Pi 4 single board computer: this powerful processing board is able to support the computational requirements of the encoding and decoding processes in real time. On top of it, the *HiFiBerry DAC+ ADC Pro* [18], a dedicated Digital to Analog Converter (DAC) card, is responsible for the signal conversion between the Raspberry Pi and the transducer at a sampling frequency of 192 kHz. For the transducer, we selected the Aquarian Scientific AS-1 [19], that costs about 400 USD, paired with a pre-amplifier in reception: the PA-4 Hydrophone Preamplifier [20], again from Aquarian Scientific. The SDM was previously tested with different physical configurations: two different modulation schemes, various channel encoding schemes for error correction and various samples-per-symbol settings. As shown in Figure 2, the SDM software opens a number of server sockets to communicate with the protocol section of the platform (i.e., the DESERT protocol stack) and the user, namely:

- a socket for transmitting diagnostics information: this includes the number of headers decoded, the number of packets decoded, RSSI values and others;
- a socket for receiving commands: this is used to send configuration changes to the SDM such as the transmission gain or the Forward Error Correction (FEC) scheme to use;
- a socket for transmitting event notifications to the driver: this messages include events such as *transmission ended* or *reception began* to allow connected drivers to correctly manage the sending and reception processes;
- a socket to send and receive data, or the DESERT Service Data Unit (SDU): the SDM will take care of generating the Protocol Data Unit (PDU) to be sent over the channel.

To allow the DESERT framework to take care of the whole protocol section, we set as MAC layer of the SDM the *Dummy MAC*, as illustrated in Figure 1: this implies that, once packets are demodulated by the physical layer, they are immediately forwarded to the interface connecting to DESERT and that, conversely, the packets that are coming from the DESERT interface are directly sent to the physical layer of the SDM, that prepares them for transmission. The user can thus switch among all the modulation and encoding schemes and settings available in the SDM with a transparent MAC layer. The DESERT framework, instead, is best suited to study, implement and evaluate protocols, from the MAC to the application layer: a vast array of these protocols are already implemented in the current version.

III. DESERT UNDERWATER

DESERT Underwater [21] is a collection of protocols and modules, written in C++, freely available as open source software, that can be used to simulate underwater networks either through a simple channel model based on Urick's formula, or through WOSS [22], a powerful channel simulator based on the Bellhop ray-tracing engine. Based on the NS-2 network simulator, the DESERT Underwater libraries are meant as a flexible and reliable framework to support the design and implementation of underwater network protocols. DESERT can switch the channel model with a device driver: then, by connecting real devices through the socket connections that DESERT sets up, it is possible to send and receive packets in real field-trials. Up to date, DESERT already includes the device drivers for the EvoLogics S2C [23] modems and the TUHH *ahoi* modems [24].

We implemented the device driver able to connect the protocol stack, developed and tested in DESERT, to the MODA SDM described in Section II, thus realizing a flexible, open and inexpensive architecture, presented in Section IV, able to configure both the protocol stack and the physical layer parameters. The device driver is based on a simple state machine that, based on the notifications coming from the *signaling* channel, manages the transmission and reception processes. The *diagnostics* channel, on the other hand, makes it possible to implement the optimization algorithm that researchers desire to implement and test, and whose changes are carried out through the *commands* channel.

IV. COMPLETE PLATFORM

The final Software Defined Platform, presented in Figure 1, is based on the communication of the DESERT physical layer, the driver, with the SDM through the various socket channels presented earlier and depicted in Figure 2. We highlight that both the device driver in DESERT and the SDM are still under active development: the demo presented in this work has the purpose of showcasing the capabilities and not to study or evaluate a particular protocol.

Because the architecture is made of two main components that communicate via sockets, these two can be run on different hardware. In particular, if the processing board chosen to run the SDM has limited processing or memory capabilities, the protocol stack implemented in DESERT can be run on a separate more powerful machine, thus lowering the burden on the SDM platform as long as the two entities are able to communicate in the same local IP network. This scheme would also ease the process of collecting diagnostic data and logs as well as the supervision of the novel protocol algorithm under evaluation. However, this is not mandatory and the two

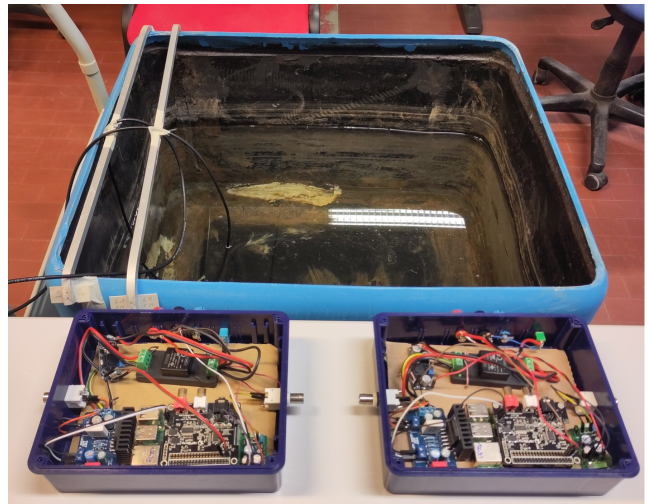

Fig. 3. Demo setup: the two SDM modems, on the table, in front of the water tank deployed during the test.

components can run on the same processing unit if the user so desires.

The SDM includes a MAC layer, which, in this configuration, is set to be transparent to allow DESERT Underwater to implement it. The connection servers reside on the SDM, meaning that it must be the first to be set up and running: the DESERT layers will then be activated and connected to these socket servers. While the *data* and *signaling* channels, that are devoted to correctly guide the driver's state machine, are automatically connected, the *diagnostics* and the *commands* channels are connected to assess the performance or to switch configurations, or can be accessed directly via, e.g., the `netcat` utility.

The resulting platform is thus composed of the MODA SDM and DESERT Underwater: the former can be used to configure different modulation and channel encoding schemes (constrained only by the deployed transducer), while the latter can be used to develop and configure protocols for the MAC, network and application layers making it a very versatile and flexible platform.

The SDM developed at the University of Padova is affordable and highly customizable: the cost of making one unit could be less than a few hundred EUR, depending on the desired features and, mostly, on the chosen transducer. The DESERT Underwater framework is an open source software freely available on GitHub [25]. Thus, the emulation platform we are proposing is extremely affordable, and provides great benefits to research institutions and to the community in general.

V. DEMO

To test this Software Defined Platform, we implemented a simple mechanism that, based on the readings performed on the diagnostics channel, decides to switch some parameters based on a selected threshold, in detail:

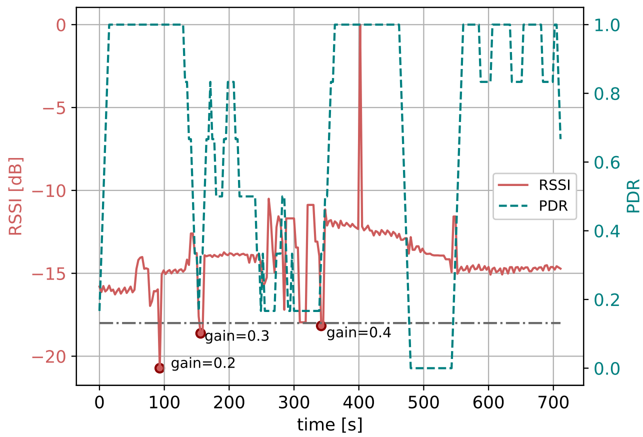

Fig. 4. PDR values obtained by switching the transmission gain based on an RSSI threshold.

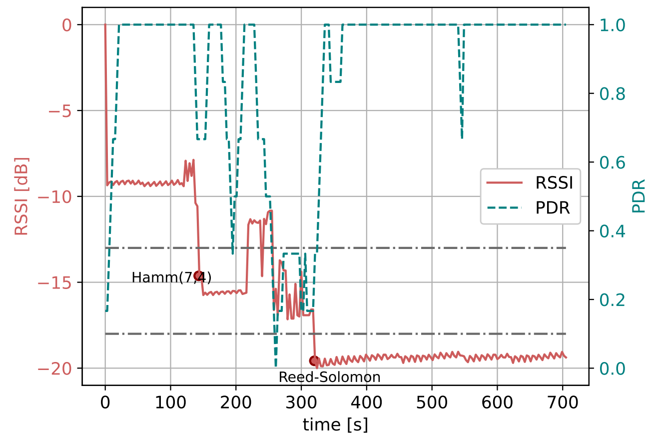

Fig. 5. PDR values obtained by switching the FEC scheme based on an RSSI threshold.

- we selected the Received Signal Strength Indicator (RSSI) and the PDR as metrics to trigger configuration changes;
- we tested the change of transmission gain and FEC;
- we observed the effects on the PDR.

We want to emphasize that the purpose of this demonstration is not to prove the advantage of some proposed adaptation algorithm to counter the adverse channel effects: rather, our intent is to demonstrate the capabilities and effectiveness of such an architecture.

In our test we had the DESERT protocol stack of both nodes running on a desktop PC, while the SDM was running on two Raspberry Pi 4 processing boards. The other components are those described in Section II. We put the two transducers in the water tank, depicted in Figure 3, that is accurately described in [26]. In this test we assume a perfect feedback link implemented through radio communications, in order to focus our study on how the change of communication conditions and system parameters affect the system performance and to simplify our analysis that, in the presence of an imperfect feedback channel (such as the one experienced with underwater acoustic transmissions) would have become more difficult. We set up a node as a receiver, and a node that transmitted a fixed-size packet every 3 s, employing the Constant Bit Rate (CBR) module of DESERT. The size of the packets was 3 bytes at the application layer and 18 bytes as SDU for the modem. To trigger the configuration changes, that are based on performance thresholds, we slowly moved the receiving transducer, linearly, along the tank length.

A. An RSSI-based algorithm

The first test consisted of analyzing the RSSI values provided by the SDM and changing first the transmission gain, and subsequently the FEC scheme whenever the RSSI value became lower than a selected threshold. In the first case, each time the threshold of -18 dB was reached, the protocol stack increased the transmission gain by 10%, starting from an initial value of 10% of the total gain. The results, expressed in terms of the PDR, can be seen in Figure 4: we can see that the RSSI curve, being strongly dependent on the transmission gain, is clearly increased after each change occurs but the same is not obvious for the PDR (dashed curve). Its value can be seen recovering after each gain transition, but soon decreases anyway and, until the threshold is not reached again triggering a new gain transition, the PDR continues to decrease as can be seen in the section between values 0.3 and 0.4 of the transmission gain. In an evaluation test, one could infer that such a simple algorithm needs better threshold definition. In the second case, we decided to evaluate a possible algorithm still based on RSSI readings, but this time to trigger a switch of the FEC scheme. In particular, starting from a transmission with no encoding at all, we selected a first transition to a Hamming (7,4) encoding scheme for the inner code, on a -13 dB threshold and a second transition to a Reed-Solomon scheme for the outer code of the FEC scheme on a -18 dB threshold. In this case, from Figure 5, we can see that the recovery of the PDR is very weak after the first transition and its effect seems to disappear after a few packets. On the other hand, the second transition, that introduces a very strong scheme based on Hamming(7,4) for the inner code and Reed-Solomon for the outer one, is able to re-establish the maximum PDR, even when the RSSI values reach their minimum. If this were a protocol evaluation test, one could deduct that the second switch should be done at the first threshold and maybe that another configuration change would not be helpful.

B. A PDR-based algorithm

In a second test, we emulated the evaluation of a switching protocol based on the measurement of the PDR. In particular, we implemented a simple algorithm that collected the correctly received packets for an interval of 9 s, at the end of which it was able to calculate a sort of instantaneous PDR by dividing

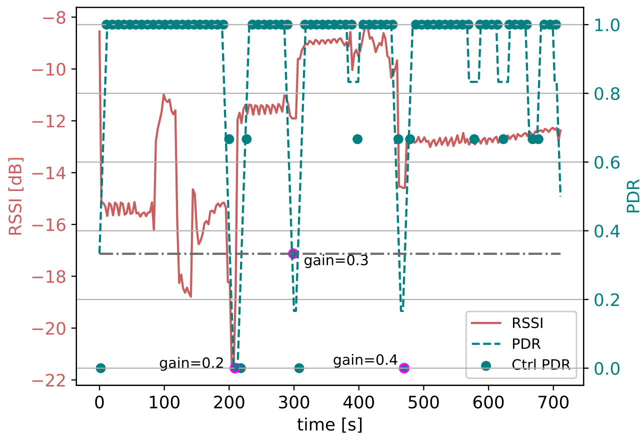

Fig. 6. PDR values obtained by switching the transmission gain based on a PDR threshold.

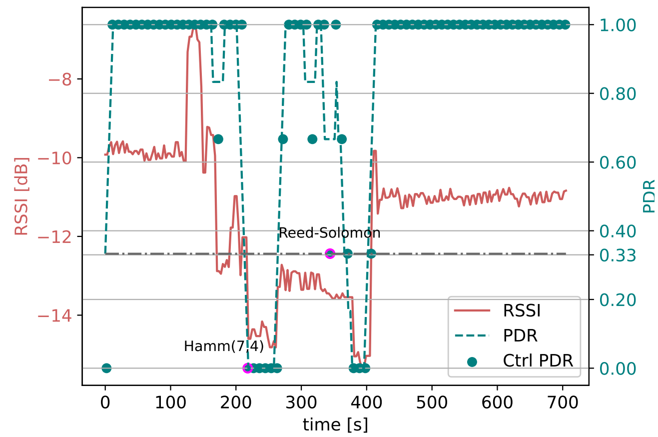

Fig. 7. PDR values obtained by switching the FEC scheme based on a PDR threshold.

this value by the maximum number of transmitted packets that, for the given interval, is known to be 3. In a real evaluation scenario, where the rate of transmitted packets may not be known, the receiver would still be able to infer the failed packets by being able to decode the header or detect the reception. In any case, it would be possible for a receiving node, on a long test run, to estimate the transmission pattern and make assumptions on the transmission rate as many widely deployed traffic analyzers used in terrestrial networks, e.g., LoRaWAN.

Then, similarly to the previous test, the protocol stack increased, in the first case, the transmission gain by 10% starting from an initial value of 10% of the maximum gain on a selected threshold of 0.33 of the PDR. From Figure 6 it is possible to see that the PDR is able to recover after each triggered transmission gain change, apparently better than the RSSI-based case: in particular, each interval between gain changes shows a very high PDR and that this drops very quickly at specific time instants. The dots-based plot, named *Ctrl PDR* is the *instantaneous PDR* used for controlling the configuration switch, computed on a 9 s interval, as explained above while the dashed line represents the PDR retrieved from the diagnostics as done in the previous test.

In the second case, we added a coding scheme whenever the PDR value reached, from higher values, the threshold of 0.33. It can be seen from Figure 7 that both the first change in FEC scheme, that adds a Hamming(7,4) as the inner code, and the second one that adds a Reed-Solomon as outer code, are able to completely recover the packets reception rate. Summarizing, in a real evaluation scenario, it could be inferred that a PDR-based algorithm would perform better than an RSSI-based one.

VI. Conclusions and Future Works

In this work, we presented an open, customizable and inexpensive architecture for building underwater nodes. This architecture can be easily deployed by researchers and users to test and evaluate novel protocols and communication schemes. It is made of two main components: DESERT Underwater is an open source underwater network simulator freely available [25], while the MODA SDM can be requested from the authors [27] or built in-house and is composed of the actual SDM software and the hardware supporting it. We believe that this platform is a very flexible and useful tool for the study of underwater acoustic networks, that can be afforded by most of the users and research teams, even in batch quantities.

In our future research and development, we plan to advance the number of available metrics that the SDM is capable of sending to the protocol stack as well as the number of configurations that DESERT is able to change on the modem. We also plan to exploit this tool to carry out research on adaptation protocols and algorithms for underwater networks, especially in the context of swarm drones operations and localization.

References

[1] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, “The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms,” in *Proceedings of OCEANS 2005 MTS/IEEE*, vol. 2, Washington D.C., US, Sep. 2005.

[2] 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.

[3] M. Y. I. Zia, J. Poncela, and P. Otero, “State-of-the-art underwater acoustic communication modems: Classifications, analyses and design challenges,” *Wireless personal communications*, vol. 116, pp. 1325–1360, Jan. 2021.

[4] I. Lončar, A. Babić, B. Arbanas, G. Vasiljević, T. Petrović, S. Bogdan, and N. Mišković, “A Heterogeneous Robotic Swarm for Long-Term Monitoring of Marine Environments,” *Applied Sciences*, vol. 9, no. 7, pp. 2076–3417, Apr. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/7/1388

[5] N. Morozs, P. D. Mitchell, and R. Diamant, “Scalable Adaptive Networking for the Internet of Underwater Things,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 023–10 037, Apr. 2020.

[6] F. Steinmetz and B.-C. Renner, “From the Long-Range Channel in the Ocean to the Short-Range and Very Shallow-Water Acoustic Channel in Ports and Harbors,” in *Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, Aug. 2021.

[7] M. Falk, G. Bauch, and I. Nissen, “On channel codes for short underwater messages,” *Information*, vol. 11, no. 2, Jan. 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/58
[8] L. Ma, H. Jia, S. Liu, and I. U. Khan, “Low-Complexity Doppler Compensation Algorithm for Underwater Acoustic OFDM Systems With Nonuniform Doppler Shifts,” *IEEE Communications Letters*, vol. 24, no. 9, pp. 2051–2054, May 2020.
[9] A. Pottier, P.-J. Bouvet, and P. Forjonel, “Performance evaluation of Multi-User MIMO Underwater Acoustic Communications,” in *Fifth Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, Aug. 2021.
[10] H. S. Dol, P. Casari, T. van der Zwan, and R. Otnes, “Software-Defined Underwater Acoustic Modems: Historical Review and the NILUS Approach,” *IEEE Journal of Oceanic Engineering*, vol. 42, no. 3, pp. 722–737, Sep. 2017.
[11] “Subnero–Subnero Products,” Last time accessed: Apr. 2023. [Online]. Available: https://subnero.com/products
[12] “UnetStack3–The Underwater Networks Project,” Last time accessed: Apr. 2023. [Online]. Available: https://unetstack.net
[13] I. V. Zhilin, O. M. Bushnaq, G. De Masi, E. Natalizio, and I. F. Akyildiz, “A Universal Multimode (Acoustic, Magnetic Induction, Optical, RF) Software Defined Radio Architecture for Underwater Communication,” in *Proc. International Conference on Underwater Networks & Systems (WUWNet)*, Shenzhen, China, Nov. 2021.
[14] J. Potter, J. Alves, T. Furfaro, A. Vermeij, N. Jourden, D. Merani, G. Zappa, and A. Berni, “Software Defined Open Architecture Modem development at CMRE,” in *Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, Sep. 2014.
[15] 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),” in *Proc. MTS/IEEE Oceans*, Chennai, India, Feb. 2022.
[16] E. Coccolo, R. Francescon, F. Campagnaro, and M. Zorzi, “Field Tests of the Software Defined Modem Prototype for the MODA Project,” in *Sixth Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, Aug. 2022.
[17] “liquid-dsp GitHub repository,” Last time accessed: Apr. 2023. [Online]. Available: https://github.com/jgaeddert/liquid-dsp
[18] “HiFiBerry DAC+ ADC Pro,” Last time accessed: Apr. 2023. [Online]. Available: https://www.hifiberry.com/shop/boards/hifiberry-dac-adc-pro
[19] “AS-1 Hydrophone,” Last time accessed: Apr. 2023. [Online]. Available: https://www.aquarianaudio.com/as-1-hydrophone.html
[20] “PA-4 Hydrophone Preamplifier,” Last time accessed: Apr. 2023. [Online]. Available: https://www.aquarianaudio.com/pa4.html
[21] “DESERT Underwater website,” last time accessed Apr. 2023. [Online]. Available: https://desert-underwater.dei.unipd.it/
[22] “WOSS website,” Last time accessed: Apr. 2023. [Online]. Available: http://telecom.dei.unipd.it/ns/woss/
[23] “Evologics Acoustic Modems,” Last time accessed: Apr. 2023. [Online]. Available: https://evologics.de/acoustic-modems
[24] C. Renner and A. J. Golkowski, “Acoustic Modem for Micro AUVs: Design and Practical Evaluation,” in *Proc. International Conference on Underwater Networks & Systems (WUWNet)*, Shanghai, China, Oct. 2016.
[25] “DESERT Underwater repository,” Last time accessed: Apr. 2023. [Online]. Available: https://github.com/signetlabdei/DESERT_Underwater
[26] F. Meneghello, F. Campagnaro, R. Diamant, P. Casari, and M. Zorzi, “Design and Evaluation of a Low-Cost Acoustic Chamber for Underwater Networking Experiments,” in *Proc. International Conference on Underwater Networks & Systems (WUWNet)*, Shanghai, China, Oct. 2016.
[27] “Wireless and More – MODA SDM,” Last time accessed: Apr. 2023. [Online]. Available: https://www.wirelessandmore.it/sdm.html