

The DESERT Underwater Framework v2: Improved Capabilities and Extension Tools

Filippo Campagnaro, Roberto Francescon, Federico Guerra, Federico Favaro,
Paolo Casari, Roe Diamant, Michele Zorzi

Abstract—The DESERT Underwater emulation system (<http://nautilus.dei.unipd.it/desert-underwater>), originally designed for testing underwater acoustic networks, has been recently extended. The new framework now includes multi-modal communication functionalities encompassing low rate and high rate acoustics as well as optics, the capability of testing wireless telemetry for underwater equipment, a connection to the most recent version of the World Ocean Simulation System (WOSS), a modification to the RECORDS system for sea trial remote control, and an interface between external tools, e.g., Matlab, and the EvoLogics modem. In addition, experimental activities are now supported by an accurate real-time event scheduler which has been shown to support, among others, long experiments involving time-division multiple-access (TDMA)-based MAC protocols. These additional protocol schemes from the MAC to the application layer (most of which have been tested in controlled environments and sea trials) now make DESERT Underwater a comprehensive tool for underwater network simulation and experimentation. In this paper, we present the new functionalities developed over the last two years.

Index Terms—Underwater networks, simulation, emulation, test-bed, NS-Miracle, WOSS, RECORDS, multi-modality, optical, Hermes, T2Mo, Matlab, EvoLogics.

I. INTRODUCTION

With the advance of underwater wireless communications, efficient network protocols have been recognized as the bottleneck for underwater network deployments [1], [2]. A means to objectively evaluate the performance of such protocols has therefore become an important requirement, and underwater network simulators play a crucial role in this respect. The panorama of simulators for underwater networks is quite broad. Starting from application-specific code (e.g. [3], [4]), several tools were evolved into (or as part of) larger frameworks [5]–[9]. Among these, [5] is based on the GloMoSim framework. Aqua-Net Mate [8] extends the AquaNet framework with the possibility to switch between simulations and real-life experiments in underwater acoustic networks. The SUNSET framework [7] is based on ns2 and MIRACLE, and provides support for simulation, emulation and experimentation. Recently, [9] was presented as a Java/groovy based development for seamless simulation and experimentation. The open-source DESERT Underwater [6] emulator, available

online [10], offers a complete set of underwater networking capabilities. The framework is based on the MIRACLE extensions [11] of the network simulator ns2 [12]. MIRACLE, developed at the University of Padova, enhances ns2 with the support of modular stacks and cross-layer communications. Additionally, it enables the co-existence of multiple modules within each layer of the protocol stack. DESERT was established to simulate and experiment underwater acoustic networks, and features an increasingly broader community of users. Among others, SINTEF (Norway) employed DESERT for an under-ice sensing project [13], the RACUN consortium adopted DESERT for code reuse between simulations and experimentation [14], [15], and SPAWAR (San Diego, CA) is using DESERT as a realistic simulation engine. DESERT is also receiving contributions through its dedicated mailing list with the majority of users located in Asia, Europe and the US.

In this work, we report the recent extensions of the DESERT framework. The development of different rate acoustic and non-acoustic communications via, e.g., optical technology, calls for a tool for simulating general, multi-modal networks. Addressing this need, the MIRACLE libraries have been updated to support multiple and heterogeneous protocol stacks in each node. This exploits the modularity of the system and its cross-layer message exchange capabilities. This feature enables the development of intermediate layers, that decide where to forward the traffic according to the status and the metrics collected from each sub-stack. The new version of DESERT provides several tools to simulate different underwater communications types. This includes the interface to the new version of the World Ocean Simulation System (WOSS) [6] and a new module to simulate a real acoustic modem behavior. Optical communications have been modeled in detail and made part of multi-modal communication systems. DESERT also provides application examples for multimodal network, such as a wireless telemetry add-on to control underwater vehicles remotely.

In the most recent version of DESERT (currently v2.1.2), we also improved its field experimentation components by implementing modular and stackable processes for the translation from the data structures of the simulator into actual bit streams and vice-versa. We have also added support for packet fragmentation and reconstruction (see, e.g., [6], [15]). We also combined DESERT with the RECORDS framework, in order to perform accurate and controlled experiments and measure the acoustic channel during field experiments. Moreover, we improved the support for real-time event scheduling, and introduced T2Mo, a new framework that exploits DESERT's

F. Campagnaro (Corresponding author, email: campagn1@dei.unipd.it), R. Francescon, F. Guerra, F. Favaro, R. Diamant and M. Zorzi are with the Department of Information Engineering, University of Padova, Italy. R. Diamant is also with the Department of Marine Technology, University of Haifa, Israel. P. Casari is with IMDEA Networks, Madrid, Spain.

This work has been supported in part by the US Office of Naval Research under Grant no. N62909-14-1-N127.

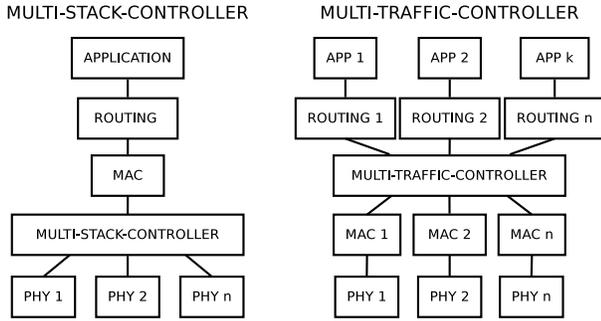


Fig. 1. Structure of the MULTI-STACK-CONTROLLER and the MULTI-TRAFFIC-CONTROLLER protocol stacks for underwater communications.

modem drivers to perform real sea tests using external processes, such as Matlab [16] scripts.

In the following sections, we provide more details on DESERT's new capabilities, complemented by a short recap on further tools for underwater network simulation and experimentation developed at the University of Padova. First, in Section II, the new DESERT functionalities are presented. This includes new modem support capabilities (II-A and II-C), multi-modal networks (II-B) and the new real-time scheduler (II-D). In Section III, we describe the connection to the WOSS system (III-A), the new functionality for connecting EvoLogics modems to external processes like Matlab (III-B) and the modification of the RECORDS tool (III-C). In Section IV we draw conclusions and illustrate ongoing work.

II. THE NEW DESERT UNDERWATER FUNCTIONALITIES

A. Simulation of optical communications

The new DESERT framework provides modules for simulating both optical propagation and the behavior of an optical PHY. The model implemented has been presented in [17], where the authors consider an optical transmitter based on a high-power Light Emitting Diode (LED), and a photodiode receiver [18]. We extended this model by including background ambient light noise into DESERT, in the form of a per-depth look-up table. The table is obtained either by employing radiative transfer model simulators, such as Hydrolight [19], or from real measurements. We implemented a reception error model based on a signal-to-noise ratio (SNR) threshold and a second one based on the on-off keying (OOK) error equations. More details about the optical modules in DESERT are presented in [20], where optical communications are employed to efficiently transfer bulk data.

B. Multi-modality

The support for multi-modal underwater networks simulations is one of the primary novelties of the new DESERT Underwater release. Two add-ons have been created to support multi-modality at two different levels: *i*) a MULTI-STACK-CONTROLLER layer that coordinates the switch between multiple physical layers (PHYs), either with or without signaling; and *ii*) a MULTI-TRAFFIC-CONTROLLER layer that decides to which layer incoming traffic should be forwarded.

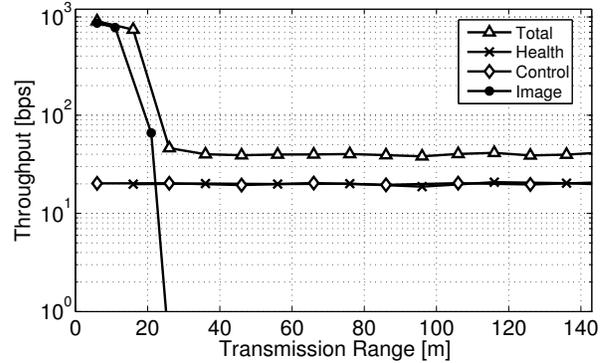


Fig. 2. Throughput vs. transmission range in a multi-modal system where different types of traffic are transmitted.

The MULTI-STACK-CONTROLLER layer is placed between the MAC and the PHYs (Fig. 1, left), and decides which is the best communication technology according to some quality metric, e.g., the average received power measured over a prescribed time period. This multi-modal system can be employed for different applications. For example, a switch policy in a master-slave fashion for data-muling applications is presented in [20]. In [21], the same layer has been extended to simulate a wireless remote control application for underwater vehicles. This new version employs a signaling mechanism to probe the channel and make the switch among different PHYs faster when no incoming packets are received over a sufficiently long time. This remote control module also provides different quality-of-service (QoS) depending on the distance between the transmitter and the receiver. This application has been included in the new version of DESERT as an add-on.

The MULTI-TRAFFIC-CONTROLLER layer is placed between the NETWORK and MAC layers (Fig. 1, right). This add-on can manage different traffic types and forward them to the best PHY according to predefined policies. In [22], a heterogeneous system is described where a set of divers exchange various types of traffic having different QoS requirements with a common sink. The multi-modal capabilities of DESERT are exploited to manage traffic and data forwarding by monitoring the availability of different communication technologies. A different policy was defined for each of three traffic types, in order to provide the required QoS. For instance, health and control packets were transmitted via long range acoustics, whereas larger data packets (such as images and chunks thereof) were conveyed through short-range, high-data rate transmission technologies such as optical modems, when the transmitter and receiver are sufficiently close. The new layers can support the simultaneous transmission of different traffic types via different PHYs while avoiding packet collisions. In Fig. 2, we show an example of the average throughput vs. transmission range for different types of traffic as obtained from a DESERT simulation. The tested system supports three communication types, namely, short periodic health packets, short sporadic control packets, and sizeable images, that are serviced through short- and long-range acoustic communications and via very short-range optical communications. Full

acoustic coverage leads to a constant throughput when control and health messages are serviced via long-range acoustics. Conversely, images are only sent via optical communications: hence, a high throughput is achieved when two peers are within the supported range. The result is an adaptive communication system, built to optimally employ the channel resources.

C. Generic modem performance module and HERMES model

A new PHY technology has been implemented in DESERT to simulate a real modem behavior. Given the bit rate, data rate, correction code employed, and the transmission range, the system sets the packet delivery ratio (PDR) according to a set of look-up tables. The details of the physical layer implementation have been presented in [23], where the authors have simulated the FAU Hermes acoustic modem [24] behavior. The FAU prototype provides high bit rate communications over short ranges: with a data rate of 87 kbps, it may have the ability to transmit low-quality videos and images over its range. In [25], the authors describe the modem in detail, and provide some sea-trial evaluations, where the packet delivery ratio is measured at different transmission ranges, from 25 to 180 m. The information from these sea trials was imported into DESERT to form the look-up tables.

D. Real-time scheduler

A key requirement for DESERT to dependably support sea experiments is accurate time reckoning via a real-time scheduler capable of correctly handling time synchronization between the *virtual* clock (C_V), kept by the software, and the *wall* clock (C_W), which is the operating system's absolute time reference. In the new version of DESERT, this capability has been added. The scheme in [26] served as the basis for the implementation of the real-time scheduler. Events are dispatched if C_V is greater than the next scheduled expiration time. In this case, the virtual clock is updated by setting $C_V = C_W$. The new version of DESERT compensates for the time difference between the instant the software realizes an event has to be dispatched and the actual scheduled time. This is performed by forcing C_V to be equal to the scheduled time. Time reckoning was further refined by introducing busy waiting for a short time interval before the expiration of each event. Specifically, even when the CPU is put to sleep until the event firing time, it still wakes up earlier and performs busy waiting before the scheduled epoch: the busy waiting interval has been set to 2 ms after [26], as an average among various platforms and to strike a balance between clock precision and energy consumption. Fig. 3 shows the offset correction performance of the DESERT real-time scheduler, compared to the version bundled in ns-2 and used in previous DESERT releases. To that end, at the execution of each event, C_W is compared with the true execution time for the case where the ns2 instance on the platform emulates 3 or 6 nodes.

We observe that the ns-2 software introduces an incremental time offset which may heavily affect real-time simulations. The new version of DESERT provides a substantial reduction of this offset and prevents the compounding of time reckoning errors. Performance reaches a bound when multiple events are

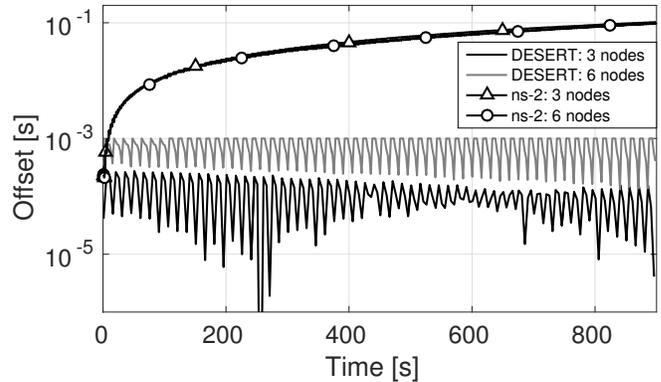


Fig. 3. Time offset correction capability of the improved real-time scheduler.

to be executed at the same time because, as the events overlap, each event has to wait for the execution of the previous one. This is believed to not strongly impact emulations as usually, in this setting, each platform runs a single instance of the emulation software and controls only one node.

III. TOOLS FOR EXTENDING THE DESERT CAPABILITIES

A. The World Ocean Simulation System (WOSS)

DESERT extends its simulation features by including the last version of the World Ocean Simulation System (WOSS) [27], an open-source framework aimed at improving underwater network simulations through realistic signal propagation modeling. The new version of WOSS can be integrated into the PHY of both ns3 [28] and ns2/MIRACLE, and is compatible with all MIRACLE-based frameworks, such as DESERT and SUNSET [7]. WOSS can generate time-varying environmental conditions by retrieving data from public databases. This includes the WOA database for sound speed profile, GEBCO for bathymetry, and Deck41 for the geoacoustic properties of bottom sediments. WOSS automatically interfaces itself with the Bellhop ray tracing software [29] to obtain simulated channel impulse responses. WOSS can also import performance figures derived from offline simulations of a full transmitter-receiver chain. Such figures are employed to characterize the packet error probability of a specific transceiver. Further details are given in [6], [30]. The last version of DESERT also exploits the large set of mobility models implemented in WOSS. This makes it possible to generate more realistic time-varying impulse responses.

B. Tool/Task to Modem (T2Mo)

The new T2Mo Framework interfaces external processes to the EvoLogics modem by employing the new version of the DESERT drivers. This capability can also be used to readily conduct real sea experiments not involving the ns2 framework. The big advantage of T2Mo is to quickly and easily develop code for sea trials, either by using code created specifically for the user's purposes, or by starting from the code previously used for other tasks such as Matlab simulations. This capability was first exploited in a sea experiment to run a handshake-based scheduling protocol implemented in Matlab

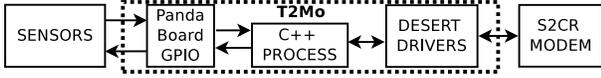


Fig. 4. Using the T2Mo framework in a PandaBoard ES.

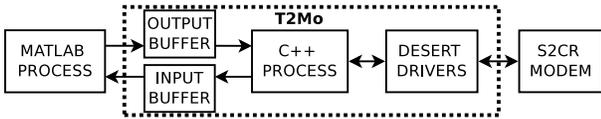


Fig. 5. Using the Tool to Modem framework (T2Mo) with Matlab.

on the EvoLogics modems [31]. Although the Matlab code is not optimized or designed to run in embedded systems, it can be extremely useful to interface the code with the modems. This allows a user to perform complex trials with no need to re-write the code for more complex or efficient systems. T2Mo is very light, and can be employed to run experiments and demos using single board computers (SBCs), such as the PandaBoard ES. An example of this usage is illustrated in Fig. 4, where this tool is used to control and retrieve data from sensors connected to the PandaBoard’s pins, by employing C++ GPIO libraries.

For an external Matlab process, the structure of the framework is presented in Fig. 5. A Matlab process generates packets according to the designed protocols, and saves each packet’s meta-data and transmission time in a file that implements an output buffer. This operation is executed before the packet transmission time, in order to avoid buffering and transmission delay. A C++ process periodically accesses the output buffer, to check whether a new packet needs to be transmitted. In that case, it exploits the DESERT drivers (adapted to run independently of ns2) to send the packet at the scheduled time, through the EvoLogics modem. The same process also receives incoming packets and stores them in a file used as input buffer. Finally, the Matlab process periodically checks the incoming buffer to handle the received packets. While the C++ process manages transmissions and receptions, the user can develop his own network protocols in Matlab. In Fig. 6, we report the processing delay from when the Matlab process schedules the transmission of a packet of duration 0.066 s until its successful reception, using two EvoLogics modems located 6 cm apart on a table. The processing delay consists of the T2Mo delays, drivers processing, TCP/IP connections between the PC and the modems, and modem processing. To measure the processing delay, the packet duration is subtracted from the end-to-end delay. We note that the propagation delay is negligible. Fig. 6 shows that the processing delays from Matlab to C++ and vice-versa are negligible compared to the total processing delay. We also observe that buffering packets from Matlab to C++ is faster than the converse, since C++ processes are more efficient than interpreted Matlab scripts.

C. RECORDS: a network remote control framework

RECORDS [32] is a remote control framework that supports underwater networking experiments in distributed contexts, where direct access to nodes is limited. RECORDS is available

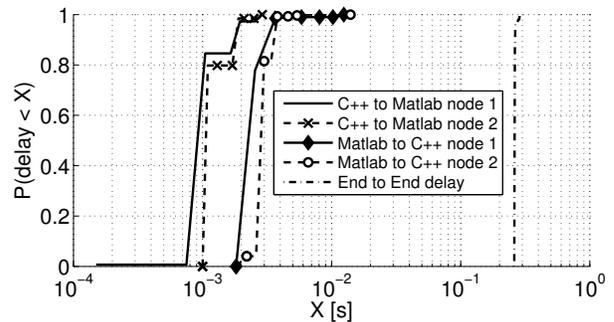


Fig. 6. Cumulative distribution function (CDF) of the processing delay. The processing delay from the Matlab to the C++ process and vice-versa is negligible compared to the total processing delay. The same behavior was obtained in both nodes of the network for a transmission of 250 packets.

at [33]. The main capability of RECORDS is to control and monitor each underwater node remotely. This includes starting and terminating DESERT runs, checking local as well as remote logs and failures, setting the transceiver parameters and logging multipath realizations. The former version of RECORDS was successfully implemented for the CMRE-organized COMMSNET’13 cruise. The new version of this tool has been made more versatile to fit the requirements of typical sea experiments: by exploiting all measurements the modem can provide, the tool now logs also Doppler measurements, delay, and noise power (besides multi-path realizations) for later post-processing. This new version has been employed during the CMRE ALOMEx’15 cruise, when measurements of acoustic channel realizations were recorded. RECORDS is optimized for the EvoLogics modem [34]; however, it can be adapted to any modem by programming a simple driver module.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented new modifications and functionalities of DESERT Underwater v2. The new version of DESERT supports a flexible multi-modal underwater network, provides extended capabilities to support a variety of underwater modems, connects the DESERT framework with the WOSS channel simulation system, and allows external processes like Matlab to connect to the EvoLogics modems. DESERT now also provides a more extensive set of tools for conducting sea experiments. This includes a real-time scheduler and a real-time reliable data logging system. In addition, the collection of supported network protocols was extended and the DESERT code was refactored for improved stability. The DESERT framework is under continuous development to make it a lightweight and reliable tool for the simulation, emulation, and experimentation of underwater networks for the underwater communication community at large. Specifically, new features to support additional underwater optical and acoustic modems will be implemented, and the interface between DESERT and the EvoLogics modems will allow the user to log and possibly retrieve the status of the channel (multipath, noise power, RSSI) in real time with tunable granularity.

REFERENCES

- [1] I. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Elsevier's Ad Hoc Networks*, vol. 3, no. 3, 2005.
- [2] P. Casari and M. Zorzi, "Protocol design issues in underwater acoustic networks," *Elsevier Computer Communications*, vol. 34, no. 17, pp. 2013–2025, Nov. 2011.
- [3] A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, 2008, code available at <http://www.isi.edu/ilense/software/>.
- [4] K. Ovaliadis and N. Savage, "Underwater sensor network simulation tool (USNet)," *Int'l J. of Computer Appl.*, vol. 71, no. 22, pp. 19–27, Jun. 2013.
- [5] S. K. Dhurandher, M. S. Obaidat, and M. Gupta, "An acoustic communication based AQUA-GLOMO simulator for underwater networks," *Springer Human-centric Computing and Information Sciences*, vol. 2, no. 3, 2012. [Online]. Available: <http://dx.doi.org/10.1186/2192-1962-2-3>
- [6] P. Casari *et al.*, "Open-source suites for the underwater networking community: WOSS and DESERT Underwater," *IEEE Network, special issue on "Open Source for Networking: Development and Experimentation"*, vol. 28, no. 5, pp. 38–46, Sep. 2014.
- [7] "The Sapienza University Networking framework for underwater Simulation, Emulation and real-life Testing - SUNSET," Last time accessed: June 2016. [Online]. Available: http://reti.dsi.uniroma1.it/UWSN_Group/index.php?page=sunset
- [8] Y. Zhu, S. Le, L. Pu, X. Lu, Z. Peng, J.-H. Cui, and M. Zuba, "Aqua-Net Mate: A real-time virtual channel/modem simulator for Aqua-Net," in *Proc. MTS/IEEE Oceans*, Bergen, Norway, Jun. 2013.
- [9] M. Chitre, R. Bhatnagar, and W. S. Soh, "UnetStack: An agent-based software stack and simulator for underwater networks," in *Proc. MTS/IEEE OCEANS*, St. John's, NL, Canada, Sep. 2014.
- [10] "DESERT Underwater github repository," Last time accessed: June 2016. [Online]. Available: https://github.com/uwsignet/DESERT_Underwater
- [11] N. Baldo, M. Miozzo, F. Guerra, M. Rossi, and M. Zorzi, "Miracle: the multi-interface cross-layer extension of ns2," *EURASIP J. of Wireless Commun. and Networking*, 2010. [Online]. Available: <http://www.hindawi.com/journals/wcn/aip.761792.html>
- [12] "The Network Simulator - ns-2," Last time accessed: June 2016. [Online]. Available: http://nsm.isi.edu/nsm/index.php/User_Information
- [13] T. A. Reinen, A. Lie, and F. T. Knudsen, "Sensis - underwater acoustic network for ice-monitoring," in *Proc. Scandinavian Symposium on Physical Acoustics*, Gello, Norway, Feb. 2016. [Online]. Available: <http://arxiv.org/pdf/1604.02253.pdf>
- [14] J. Kalwa, "The RACUN-project: Robust acoustic communications in underwater networks – an overview," in *Proc. IEEE/OES OCEANS*, Santander, Spain, Jun. 2011.
- [15] C. Tapparello, P. Casari, G. Toso, I. Calabrese, R. Otnes, P. van Walree, M. Goetz, I. Nissen, and M. Zorzi, "Performance evaluation of forwarding protocols for the RACUN network," in *Proc. ACM WUWNet*, Kaohsiung, Taiwan, Nov. 2013.
- [16] MathWorks, "Matlab," Last time accessed: June 2016. [Online]. Available: www.mathworks.com/products/matlab/
- [17] Davide Aguita, Davide Brizzolara, Qilng Hu, "Optical Wireless Underwater Communication for AUV: Preliminary Simulation and Experimental Results," in *IEEE Oceans 2011*, 6-9 June 2011, pp. 1–5.
- [18] "Si PIN photodiode S5971," accessed: March 2015. [Online]. Available: <http://www.hamamatsu.com/eu/en/product/category/3100/4001/4103/S5971/index.html>
- [19] Sequoia Scientific, "Hydrolight radiative transfer numerical model," Last time accessed: June 2016. [Online]. Available: <http://www.sequoiasci.com/product/hydrolight/>
- [20] F. Campagnaro, F. Favaro, F. Guerra, V. Sanjuan, M. Zorzi, and P. Casari, "Simulation of multimodal optical and acoustic communications in underwater networks," in *Proc. MTS/IEEE OCEANS*, Genova, Italy, May 2015.
- [21] F. Campagnaro, F. Guerra, F. Favaro, V. S. Calzado, P. Forero, M. Zorzi, and P. Casari, "Simulation of a multimodal wireless remote control system for underwater vehicles," in *Proc. ACM WUWNet*, Washington, DC, Oct. 2015.
- [22] F. Campagnaro, F. Guerra, R. Diamant, P. Casari, and M. Zorzi, "Implementation of a multimodal acoustic-optic underwater network protocol stack," in *Proc. MTS/IEEE OCEANS*, Shanghai, China, Apr. 2016.
- [23] F. Campagnaro, F. Favaro, P. Casari, and M. Zorzi, "On the feasibility of fully wireless remote control for underwater vehicles," in *Proc. Asilomar Conf. SS&C*, Pacific Grove, CA, Nov. 2014.
- [24] P.-P. Beaujean, J. Spruance, E. A. Carlson, and D. Kriel, "HERMES - A high-speed acoustic modem for real-time transmission of uncompressed image and status transmission in port environment and very shallow water," in *Proc. MTS/IEEE OCEANS*, Québec City, Canada, Sep. 2008.
- [25] P.-P. Beaujean and J. Spruance, "A very high bit rate broadband acoustic modem for short-to-medium range data transmission in ports and shallow water using spread spectrum modulation and decision feedback equalizing," Florida Atlantic University, Tech. Rep. ONR project N00014-06-C-0368, Sep. 2008, Accessed: Oct. 2014.
- [26] D. Mahrenholz and S. Ivanov, "Real-time network emulation with ns-2," in *Proc. IEEE DS-RT*, Washington, DC, Oct. 2004, pp. 29–36.
- [27] "The World Ocean Simulation System - WOSS," Last time accessed: June 2016. [Online]. Available: <http://telecom.dei.unipd.it/ns/woss/>
- [28] ns3 Network Simulator, <http://www.isi.edu/nsm/ns/>, Last time accessed: June 2016.
- [29] M. Porter *et al.*, "Bellhop code," Last time accessed: June 2016. [Online]. Available: <http://oalib.hlsresearch.com/Rays/index.html>
- [30] F. Guerra, P. Casari, and M. Zorzi, "World Ocean Simulation System (WOSS): a simulation tool for underwater networks with realistic propagation modeling," in *Proc. of ACM WUWNet 2009*, Berkeley, CA, Nov. 2009.
- [31] R. Diamant, P. Casari, F. Campagnaro, and M. Zorzi, "A handshake-based protocol exploiting the near-far effect in underwater acoustic networks," *IEEE Wireless Commun. Lett.*, vol. 5, no. 3, pp. 308–311, Jun. 2016.
- [32] G. Toso, I. Calabrese, P. Casari, and M. Zorzi, "RECORDS: a remote control framework for underwater networks," in *Proc. IEEE/IFIP Med-Hoc-Net*, Piran, Slovenia, Jun. 2014.
- [33] "RECORDS source code," Last time accessed: June 2016. [Online]. Available: <https://github.com/uwsignet/records>
- [34] "S2CR 18/34 Acoustic Modem," Last time accessed: June 2016. [Online]. Available: http://www.evologics.de/en/products/acoustics/s2cr_18_34.html/