Towards Experimental Evaluation of Software-Defined Underwater Networked Systems

Hovannes Kulhandjian, Li-Chung Kuo, Tommaso Melodia, Dimitris A. Pados
Department of Electrical Engineering
State University of New York at Buffalo
Buffalo, NY 14260 USA
E-mail: {`hkk2, lkuo2, tmelodia, pados`}`@buffalo.edu`

Dale Green
Teledyne Benthos
49 Edgerton Drive
North Falmouth, MA 02556 USA
E-mail: `dale.green@teledyne.com`

***Abstract*— In this paper, we present our ongoing work on developing a reconfigurable underwater networking testbed based on the Teledyne Benthos Telesonar SM-75 modem. The testbed is designed with the objective of allowing researchers and developers to advance research activities in the field of underwater networking and communications through a flexible testbed platform. We will first discuss the testbed architecture, with a particular emphasis on architectural considerations, software-hardware separation, and general capabilities. We will then present work in progress on enabling support for IPv6, custom-designed medium access control, and ad hoc routing protocols. Finally, we will discuss integration of the testbed with a channel emulator that allows performing laboratory controlled experiments where the user is able to configure transmission distance, channel effects, and underwater communication parameters. Emulation results for several different scenarios as well as actual experimental results for transmission of custom defined acoustic waveform will be demonstrated at the end. This platform therefore allows playing, processing, and recording custom-defined acoustic waveforms to support reconfigurable physical layer experimentation with arbitrary transmission schemes.**

***Index Terms*—Underwater acoustic sensor networks, Software-defined radio, Experimental evaluation, Channel emulator**

I. INTRODUCTION

Underwater acoustic sensor networks (UW-ASNs) [1] are increasingly becoming the focus of intense research activity driven by commercial and defense applications such as oceanographic data collection, climate change monitoring, pollution tracking, offshore exploration, disaster prevention, and tactical surveillance. Recent research efforts have been directed at studying optimized signaling strategies, signal processing algorithms, and networking protocols specifically tailored for the underwater acoustic propagation environment. Yet, as of today, the research community is essentially lacking a versatile reconfigurable platform to enable experimental evaluation of underwater communication and networking protocols.

Extensive field experimentation is needed to validate underwater transmission schemes and networking protocols. Unfortunately, setting up an experimental platform for underwater acoustic networks is very expensive compared to establishing a radio frequency (RF) wireless sensor network testbed [2]. Not only are acoustic modems expensive, but also the deployment and maintenance of the testbed itself are costly. As a natural consequence, deployment of underwater acoustic sensors in general are less dense, fewer sensors are utilized and with longer communication range compared to deployments in terrestrial wireless sensor networks [3]. A limited number of experimental platforms have been deployed so far.

In this paper, we present our ongoing work on developing a reconfigurable underwater networking testbed based on the Teledyne Benthos Telesonar SM-75 modem [4]. The SM-75 modem is modified to allow the research community to perform advanced networking and communication experiments as follows. First, a programmable Gumstix network processor is integrated with the SM-75 modem through a newly designed interface that defines communication primitives between the modem board and the external processor. A reconfigurable, software-defined protocol stack, including medium access control (MAC), IP network layer with reconfigurable ad hoc routing, network self-configuration primitives (e.g., neighbor discovery, DHCP), is being implemented on the Gumstix board to enable the definition of complex networking experiments with reconfigurable, cross-layer designed protocol stacks. Second, the modified platform is designed such that it allows generating, transmitting and receiving custom defined acoustic waveforms to support reconfigurable physical layer experimentation with arbitrary transmission schemes.

Finally, we discuss integration of the testbed with an underwater acoustic channel emulator we developed that allows the user to perform laboratory controlled experiments. The channel emulator, based on Telesonar SM-75 modems, is designed to allow the user to configure transmission distance, modulation scheme and underwater communication parameters. The acoustic signal is processed to account for path loss, noise, and multipath of a real underwater acoustic environment through the channel emulator. In addition, the channel emulator provides a graphical user interface (GUI) and allows simulation of multi-user multiple-input-multiple-output (MU-MIMO) communication systems, which could by used for developing cooperative diversity protocols [5]. Lastly, we demonstrate the channel emulation results for several different scenarios as well as actual experimental results for transmission of custom defined acoustic waveform and compare it with emulation results.

The remainder of this paper is organized as follows. We first discuss the related work in Section II. We then discuss the developments of a reconfigurable underwater networking testbed in Section III. The underwater acoustic channel emulator is introduced in Section IV, while in Section V, we demonstrate the emulating and experimental results. Finally, in Section VI, we draw the main conclusions.

II. RELATED WORK

To validate new algorithms and networking protocols designed for UW-ASNs it is essential to have a reconfigurable experimental platform. These are in general based on commercial or experimental acoustic modems.

Commercial Acoustic Modems. Some of the leading companies involved in manufacturing commercial acoustic modems include Teledyne Benthos [4], LinkQuest [6], EvoLogics [7], DSPComm [8] and Tritech [9]; as well as a few platforms developed within the research community, most notably the WHOI Micro-Modem [10]. However, currently most of the available off-the-shelf acoustic modems are not readily reconfigurable.

Reconfigurable underwater acoustic modems should facilitate implementation of different protocols and algorithms. Flexible modems range from reconfigurable modems that allow the user to select the modulation method from a finite set of schemes, to fully reprogrammable modems that permit the user to implement any modulation and demodulation scheme in addition to flexible networking protocol in software [11].

Experimental Acoustic Modems. Several experimental acoustic modems have been developed by different research groups. In [12], a compact, low-power acoustic transceiver called Micro-Modem is developed, which is a user-programmable open alternative solution to the available commercial modems. The modulation schemes supported by the modem are frequency-hopping frequency-shift keying (FH-FSK) and phase-shift keying (PSK) with data rates ranging from $80\,\mathrm{bit/s}$ to $5,300\,\mathrm{bit/s}$. Micro-Modem also supports error correction coding (ECC) capability and allows long range communication (2 to $4\,\mathrm{km}$), in very shallow water channels. Moreover, it provides RS-232 serial port user interface and includes some basic built-in networking capabilities. Additionally, it supports four and eight channel receive hydrophone arrays and a flash memory board allowing large raw data capture.

In [13], a reconfigurable acoustic modem (rModem) is developed, which allows the user to reconfigure functionalities across different layers of protocol stack with possibility of cross-layer optimization. It contains a digital signal processor, (DSP) and a field-programmable gate array (FPGA). The FPGA allows the user to operate at any carrier frequency and bandwidth within the $1\,\mathrm{kHz}$ to $100\,\mathrm{kHz}$ range, while the DSP running at $255\ \mathrm{MHz}$ enables floating point arithmetic computation. Moreover, it has $32\ \mathrm{MBytes}$ of internal flash RAM for persistent program and data storage. rModem allows MIMO transmission schemes to be implemented by using the four configurable input and output channels. rModem provides a GUI, which may be used to control the modems's hardware, send and receive packets, and log events and data.

In [14], a Software Defined Acoustic Modem (UW SWDAM) is developed. The general idea is to get the software as close to the antennas as possible so that researchers can implement the entire modem stack in software using general purpose processors. Moreover, common operating systems such as Linux or Microsoft Windows can be implemented on the ITX platforms, which enables researchers to port algorithms from their desktop.

In [15], a software-defined research platform called Underwater Acoustic Networking plaTform (UANT), is designed with the objective to provide a flexible software-defined reconfigurable platform for researchers to experiment new protocols and modulation schemes on a fully functional underwater network. UANT uses GNU Radio, a software-defined framework, for physical layer design configurations and TinyOS for network protocol stack design. UANT allows real time configuration of the acoustic modem, hence, may adapt to constantly changing underwater acoustic environment. UANT may be reconfigured at the physical, MAC, and application layers. However, one of the drawbacks of UANT platform is that it needs to run on a personal computer.

Experimental Testbeds. As mentioned earlier experimental testbeds are essential for evaluating algorithms and protocols in real world scenarios adequately. There are a number of existing experimental testbed platforms as well as ongoing projects developed by several research groups. Seaweb [16] is among the first experimental platforms primarily designed for military applications. Seaweb is a wide-area network with DSP-based telesonar underwater acoustic modems that connects autonomous and fixed nodes together. Throughout the years many networking protocols have been developed and numerous field tests have been carried out to validate the protocols using Seaweb platform.

The Centre for Maritime Research and Experimentation (CMRE) [17], formerly known as the NATO Undersea Research Centre (NURC) [18], [19], [20], is a scientific research and experimentation NATO facility. Among other research areas, CMRE is engaged in conducting research on off-board Low Frequency Active (LFA) sensors that could be used in Cooperative distributed Anti-Submarine Warfare (CASW) [20] to create a scalable and autonomous system that potentially would remove personnel from high risk areas such as deep oceans. Moreover, CMRE is involved in standardizing channel modeling schemes and networking architecture design that supports cross-layer interactions [18]. CMRE also runs and maintains an underwater networking testbed with heterogeneous modems.

In [21], a Testbed for Underwater NEtworks (Aqua-TUNE) composed of several network nodes interconnected through hybrid wireless network system is designed. Each network node contains an acoustic modem, a floating platform for electronic devices, an RF based monitor and remote control system, and a software platform for establishing an underwater network. The modulation schemes supported by the platform are multiple frequency-shift keying (MFSK), frequency-hopping spread-spectrum (FHSS) and direct-sequence spread-spectrum (DSSS). While the software system is based on Linux implementation of Aqua-Net [22]. Aqua-TUNE provides various capabilities including synchronization, localization, link and power control as well as networking modules, which facilitates researchers in conducting underwater network experiments.

In [23], a framework for underwater Simulation Emulation and real-life Testing (SUNSET) is developed by the UWSN Group [24]. SUNSET provides a framework based on open source network simulator ns-2 [25] software. The framework contains a number of commercial acoustic modem models that allows simulation and emulation of actual underwater acoustic channel conditions. Moreover, the simulator code written in ns-2 may be ported onto a small computer-on-module hardware device like Gumstix [26], which may be embedded inside an acoustic modem or AUV's (Autonomous Underwater Vehicles) housing to control their functionalities. In addition to that the framework allows interfacing software communication modules with various hardware and commercial acoustic modems, and at the same time having an open architecture to allow integration with different acoustic modems and AUV's.

Similarly, in [27], an NS-Miracle based framework to DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols (DESERT Underwater) is developed at the University of Padova. The objective of this framework is to realize a complete set of public C/C++ libraries to support the design and implementation of underwater network protocols. DESERT Underwater extends the NS-Miracle [28] simulation software library to accommodate a number of protocol stacks for underwater networks, and to support routines essential for development of new protocols.

The Woods Hole Oceanographic Institute (WHOI) is developing an underwater acoustic network (UAN) testbed [29], which will provide a valuable infrastructure for evaluating and developing network protocols for shallow and deep water communications. The testbed can be made available for collaborative experiments with the UAN research community. The acoustic nodes in the testbed can remotely be controlled through the serial port over the Internet for most of the experimental configurations. Each testbed node includes a WHOI Micro-Modem, which is controlled by a Gumstix, on which network protocols are implemented and executed. Moreover, the testbed includes buoy nodes that are equipped with GPS receivers and Freewave radios to provide gateway routing capabilities.

Although different research groups are moving towards developing reconfigurable underwater platforms, as of now, to the best of our knowledge, there are hardly any shared reconfigurable platforms. To fill that void, we are proposing a reconfigurable underwater networking platform that could be shared among different research groups. Among the available acoustic modems we chose to work with the Teledyne Benthos Telesonar SM-75 modem due to its desirable features that are essential for building a reconfigurable underwater acoustic platform. Among these attributes are playing and recording an arbitrary waveform, high reliability, and providing substantial support for networked operations, which are discussed in greater detail in Section III.

III. TESTBED ARCHITECTURE

The underwater acoustic networking testbed at the University at Buffalo (UW-Buffalo) [30] is designed to bridge the gap between experimentation and theoretical developments in underwater communications and networking by providing the research community with a versatile and shared reconfigurable platform to enable experimental evaluation of underwater communications and networking protocols.

The testbed, shown in Fig. 1, is based on the Teledyne Benthos Telesonar SM-75 modem, which, in its many configurations, is also a key component in multiple U.S. Navy programs and of many wireless tsunami warning systems worldwide. The testbed consists of 11 Telesonar SM-75 modems, one sonar SM-975 modem, and one universal deckbox, UDB-9000, equipped with an acoustic transducer used for monitoring the underwater communications.

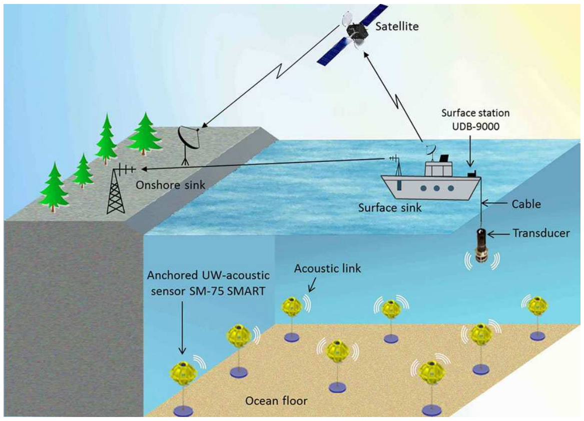


Fig. 1. Architecture of an underwater acoustic sensor network.

A. The SM-75 Modem and UDB-9000 Deckbox

The Telesonar SM-75 modem, shown in Fig. 2(a), is an all-in-one design that provides self-supported flotation, batteries, acoustic release, and RS-232 sensor interface capabilities. The acoustic modem, housed in $13\ \mathrm{inch}$ glass sphere, can be deployed in deep waters of over $6000\mathrm{m}$. Convolutional coding, interleaving and CRC are standard. The modem will sleep with a power consumption of $9\mathrm{mW}$, and will awaken to any incoming acoustic message without loss of content. Ranging accuracy is approximately $0.3\ \mathrm{m}$ and range rates (relative speeds) in excess of $20\ \mathrm{kts}$ (at frequencies between 9 and $18\ \mathrm{kHz}$) are accommodated. The acoustic data modulation methods supported by the modem are MFSK and PSK with data rates of $140-2,400\,\mathrm{bit/s}$ and $2,560-15,360\,\mathrm{bit/s}$ respectively. In addition to the above standard signaling schemes the modem may transmit an arbitrary waveform defined through standard Matlab primitives. A subset of the modems host data recorder with two SD card slots that provide high data capacity logging capability. Monitoring of all acoustic activity is done via the


(a) SM-75. (b) UDB-9000.

Fig. 2. Teledyne Benthos: (a) SM-75 telesonar modem, (b) Universal deckbox UDB-9000 with a transducer.

universal deckbox, UDB-9000, shown in Fig. 2(b), with over-the-side transducer or via Telesonabuoy with RF links to the Internet or ashore facility as illustrated in Fig. 1. The UDB-9000 is used for releasing, communicating and controlling the SM-75 modems. In addition to that it provides two RS-232 serial ports for data/command line interaction, and a large grey-scale touch screen for simple graphical user control and enhanced sunlight readability.

B. External Controller for SM-75 Modem

The SM-75 modem is interfaced with an additional external controller on a daughterboard, a Gumstix residing on the Tobi expansion board, as shown in Fig. 3. The Gumstix is a small computer-on-module, which houses TI OMAP 3530 applications processor (based on ARM Cortex-A8 CPU) clocked at $720\,\mathrm{MHz}$ and natively runs Linux (2.6.33 kernel). The DSP core of the Gumstix is based on C64x+ and delivers 3D graphics acceleration with 802.11b/g and Bluetooth communications on-board the COM. In addition, it houses 512-MB RAM, MicroSD slot, USB host, OTG port, serial ports, GPIO lines, stereo audio in/out, Ethernet 10/100 support, and HDMI interfaces.

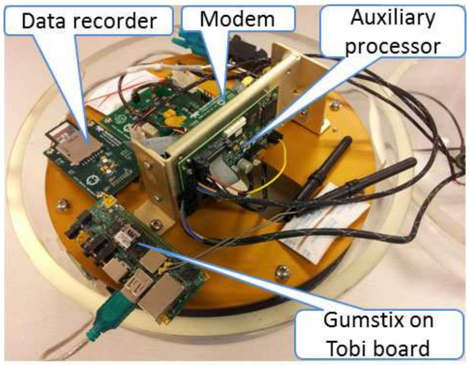
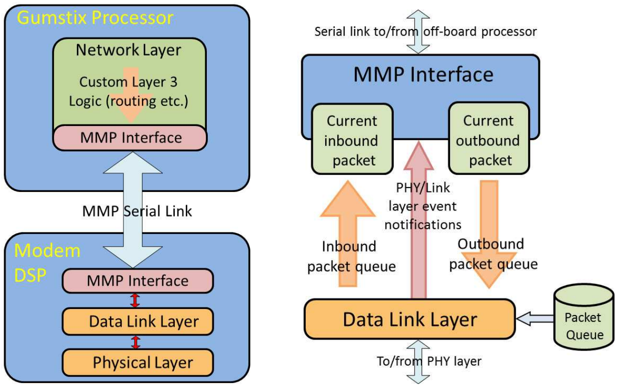


Fig. 3. Open view of SM-75 electronics with auxiliary processor, data recorder and Gumstix residing on Tobi expansion board.

The purpose of the Gumstix is to host the control logic in charge of implementing networking functionalities at all layers of the protocol stack by building on the physical/link layer application programming interface (API) exposed by the Benthos modem (modified as described in Section III-C below). Moreover, the Gumstix will add the capability to process one (or more) channels of analog signal or digital data for purposes defined by the software resident in the signal processing infrastructure. Most notably, the Gumstix will allow storing and processing data from (at least) two different channels, to allow MIMO and cooperative signal processing functionalities. The source of these data could be the modem's transducer used in parallel as a hydrophone or it could be an auxiliary hydrophone.



(a) (b)

Fig. 4. (a) Open PHY/Link networking interface, (b) Packet management and event notification.

C. Exposing PHY/LINK: A Networking API for the SM-75

In the commercial implementation of the SM-75 Benthos modem, all networking functionalities, including channel access negotiation, selective repeat request (SRQ), and waveform selection, reside within the core DSP of the individual modem, and cannot be reconfigured by the end-user. Similarly, the existing network layer implements static routing tables at each node in the network within the main modem board, and is not separable from it. Therefore, in the current on-board networking implementation, all packet processing occurs completely within the modem's CPU and firmware. This does not allow for external implementation of alternate networking and MAC schemes, and this logic is only accessible by Teledyne Benthos personnel.

A networking API software interface has been developed by Teledyne Benthos with the intention to remove the hard-coded bond between the embedded link and network layers in the modem and replace it with a new serial binary control protocol called Modem Management Protocol (MMP). The native network layer is bypassed completely and its duties are passed to an off-board processor (in our implementation it is the Gumstix) whose behavior can be defined by the end user as illustrated in Fig. 4(a). The original data link and physical layers remain unchanged, and implement many of the same peer-to-peer behaviors as SM-75 modem's lower layers do. Only, these can now be controlled through a well-defined API.

D. Development of Networking Protocols through the MMP

The logic in control of the networking functionalities is implemented in the C language and housed on the Gumstix's processor. The Gumstix interfaces with the modem via the established API and may therefore be programmed by project participants, as illustrated in the software architecture in Fig. 5. The modem's DSP will only implement physical layer and logical link control functionalities. The modem will offer an API that abstracts all the main physical and link layer functionalities to the Gumstix. In addition, it will offer access to a predefined set of status monitoring signals. These signals in turn will provide information about the current state of the communication process at lower layers of the protocol stack, i.e., signal-to-noise ratio (SNR), channel impulse response (CIR) duration, round-trip time (RTT) and relative speeds of platforms among others, that is going to be leveraged by higher layers of the protocol stack as discussed in detail below.

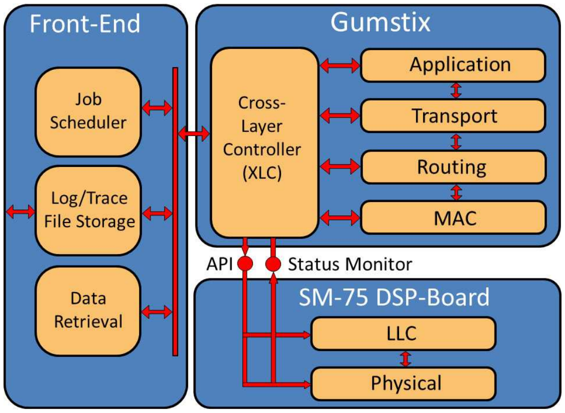


Fig. 5. Software architecture.

The means of passing information back and forth between a Teledyne Benthos acoustic modem and the Gumstix's processor is achieved through the use of MMP. The goal of MMP is to provide a static, unambiguous binary interface for machine-based command and control of the acoustic modem. The entirety of MMP may be used to control a vast array of modem functions and settings. Using the facilities of MMP the external processor on Gumstix, running a network state machine, can communicate with and control the lower protocol layers on the modem.

E. Controlling the Link Layer: Cross Layer Controller and Communication Architecture

A key architectural requirement in the design of communication protocols for underwater networking is to facilitate the use of cross-layer interactions. In underwater networks, the attainable capacity of each wireless link depends on the interference level perceived at the receiver. This, in turn, depends on the interaction of functionalities that are distributively handled by all network devices such as power control, routing, and rate policies. Hence, capacity and delay attainable at each link are location dependent, vary continuously, and may be bursty in nature. Accordingly, making efficient utilization of network resources is a challenging task. Moreover, functionalities handled at different layers are inherently and strictly coupled due to the shared nature of the underwater acoustic channel. For this reason, to develop an optimized protocol, we need to make state information from lower layers of the protocol stack available to higher layers and vice versa, what is referred to as cross-layer information sharing. We accomplish this by designing an additional module, referred in Fig. 5 as cross-layer controller (XLC), in charge of controlling and regulating this information exchange among functionalities handled at different layers of the protocol stack.

Link/Physical Status Monitoring. It is possible for the Gumstix to simply exchange packets with the lower protocol layers on the modem and keep the actions of the data link and physical layers opaque. However, it may be advantageous or necessary to have visibility into the status of the lower layers for cross-layer design. The MMP interface generates spontaneous MMP notifications for many significant events and transitions in the lower layers, which will be made available to the cross-layer controller. The Gumstix's cross-layer control logic will react to these notifications and store them in a standardized, predefined data structure. Protocols may choose to use this information or ignore them, depending on the desired end goals.

Examples of link layer notifications include: RTS sent, RTS received, CTS sent, CTS received, DATA packet sent, DATA packet received, positive ACK sent, positive ACK received, SRQ sent, SRQ received, wait for SRQ timeout period expired, PING sent, PING received, ping ECHO sent, and ping ECHO received.

There are also physical layer notifications such as packet acquisition time stamping and range rate correction measurements (sometimes referred to as Doppler), as well as notifications of packet errors. The latter will be filtered by the XLC to provide protocols with a view of the link packet error rate. Therefore, the functional "state" of the physical layer of the protocol stack will be characterized by measurements averaged in time of the underlying link quality, i.e., perceived SNR, packet error rate, and parameters characterizing the channel such as duration of the impulse response. The state of the physical layer, as described by such predefined set of variables, will be stored and updated in the cross-layer controller. Similarly, the link layer will be characterized by parameters describing link reliability, contention on the channel, transmission range. Equivalently, the network layer will be characterized by variables characterizing the energy efficiency of each neighbor, path latency, route robustness, congestion probability, as well as information describing connectivity robustness and maintenance cost.

Through a predefined set of variables and interfaces, the programmer of a module of the protocol stack (e.g., network layer) will have the possibility to export one or all the state variables to the cross-layer controller. The programmer will additionally have to define a default behavior for the protocol in case the cross-layer information is not available. This design allows keeping modularity and upgradeability of the protocols implemented.

Link/Physical Adaptive Control. While the modems will be responsible for carrying out all data link and physical layer tasks autonomously, the Gumstix may adaptively reconfigure them. Attributes such as maximum number of RTS attempts, whether to use positive acknowledgements, acoustic data rate, and link layer node addresses will all be configurable at each modem node.

Packet Queueing. The modem system implements the concept of current packets: one inbound and one outbound. In addition, it has a queue for inbound packets where they are buffered while waiting for the current packet to be processed, and another for outbound packets where they wait their turn for transmission. As new packets arrive via the link layer, they are placed in the inbound packet queue. If there is no current inbound packet, whatever is at the front of the queue becomes the current packet. A notification is sent via MMP message to signal the arrival of a new current inbound packet as shown in Fig. 4(b). If another packet arrives, it queues up behind the current one. When the current one is disposed of (forwarded to another modem in the network route or accepted locally as the final network destination) the next packet becomes the current one and a notification is generated.

F. Implementation of Baseline Networking Protocols

IPv6 Compatible Network Layer. In the current version of the Benthos modems there is no standard network layer. We are working on making the modems IP-compatible by designing and implementing an IPv4- and IPv6-compatible network header and dynamic routing tables to route network packets. This is going to be implemented based on existing Linux libraries on the Gumstix's network processor. Specific routing information, hop data, and any other desired network metrics will be carried in the network header, and will be defined and populated on the Gumstix's processor. The modems is going to be made IP-addressable, and the Gumstix's controllers will perform key routing functionalities. Communications between the modems and the Gumstix's network processor will be optimized by requesting only the network header instead of the entire packet.

Implementation of Medium Access Control Primitives. As previously discussed, the current implementation of the MMP interface includes primitives for link establishment and negotiation as well as primitives for packet transmission and acknowledgement. However, it does not implement any logic for medium access arbitration (medium access control). A primary objective of this work is to design a set of reconfigurable MAC primitives that will be the basis for experimenting with both contention-based (e.g., CSMA-like) and collision-free (e.g., TDMA-like) protocols for underwater networking. These primitives will include (but not be limited to) clear channel assessment (CCA), back-off for channel arbitration, reconfiguration of link layer acknowledgements, adaptive sleeping, implementation of periodic time slotted structure with guard bands. These primitives will be the basis for the implementation of full MAC protocols. By exposing a set of configurable mechanisms, protocols built on these primitives will make local control decisions to optimize metrics of interest, including power consumption, latency, throughput, network life-span, fairness or reliability. Currently, we are working on developing a cluster-based hybrid MAC protocol in which a TDMA scheduling is used for intra-cluster communication and CSMA/CA scheme is utilized for communication between the cluster heads and the sink node, located on the surface of the sea.

Implementation of Routing Protocols. A significant effort in the development of the testbed will consist of implementing a set of default routing protocols based on the IP-compatible addressing structure discussed before. Basic protocols will include a static (preconfigured routes) protocol, a proactive routing algorithm and a reactive routing protocol. In addition to these basic mechanisms, which will be made available as building blocks for more complex solutions, state-of-the-art specialized protocols for underwater acoustic networking will be implemented. The implementation of these protocols will be made available to the research community, hence, allowing comparison, standardized and repeatable performance evaluation of new networking protocols. Moreover, a researcher willing to experiment a new routing protocol will concentrate on developing a limited portion of the code and will rely on existing implementations of the suite of built-in protocols for the other functionalities. We will promote and encourage an open-source approach for researchers using the testbed in order to rapidly increase the suite of protocols available for research and development, similar to popular simulation platforms such as ns-2.

IV. UNDERWATER ACOUSTIC CHANNEL EMULATOR

In UW-ASNs it is very difficult to conduct repeatable and realistic experiments through a reconfigurable experimental testbed alone. That is because the physical layer is strongly dependent on the exact conditions under which an experiment is conducted and on the underwater acoustic channel environment, which, in general, is highly dynamic in nature. For this very reason in this section we are proposing an underwater acoustic channel emulator that will allow researchers to conduct laboratory controlled experiments by leveraging physical layer underwater acoustic channel emulation. We first introduce the architecture of the proposed channel emulator. We then describe the GUI we created to accompany the channel emulator.

A. Channel Emulator Architecture

The underwater acoustic channel emulator, residing in a personal computer (PC), is interfaced with the SM-75 modems through RS-232 serial port links that may control the modems to transmit and record custom defined acoustic waveforms, as shown in Fig. 6. The transmitted acoustic signals are first captured by an audio input device and fed to the channel emulator. The captured signals are signal processed in Matlab to account for path loss, noise and multipath spread of a real underwater acoustic channel. In our design of the channel emulator we apply widely used channel models and allow the user to select the parameters that have an influence on those factors. (For comprehensive treatment on the models used, the reader is referred to [2] and the references therein). Then, the modified acoustic signal is played by an audio output device and recorded by the receiver modem, which in turn attempts to decode the original transmitted information bits.

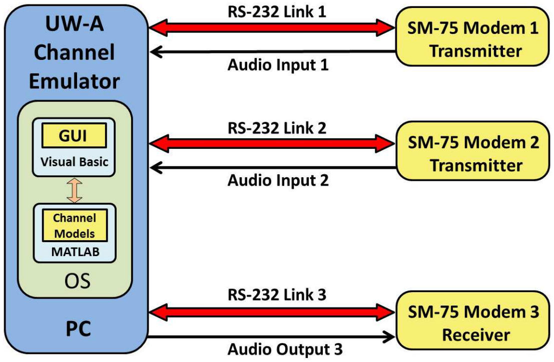
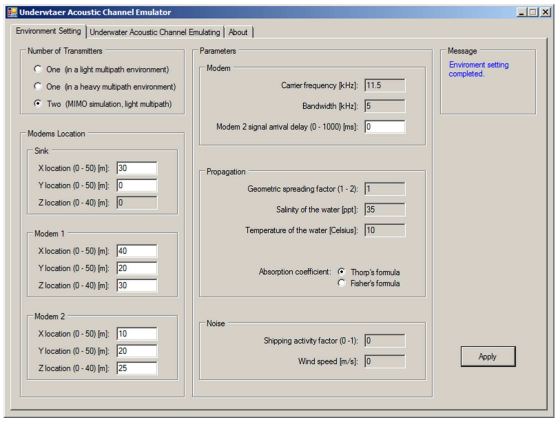


Fig. 6. Architecture of the channel emulator.



Number of Transmitters
○ One (in a light multipath environment)
○ One (in a heavy multipath environment)
● Two (MIMO simulation, light multipath)

Modems Location

Sink
X location (0 - 50) [m]: 30
Y location (0 - 50) [m]: 0
Z location (0 - 40) [m]: 0

Modem 1
X location (0 - 50) [m]: 40
Y location (0 - 50) [m]: 20
Z location (0 - 40) [m]: 30

Modem 2
X location (0 - 50) [m]: 10
Y location (0 - 50) [m]: 20
Z location (0 - 40) [m]: 25

Parameters

Modem
Carrier frequency [kHz]: 11.5
Bandwidth [kHz]: 5
Modem 2 signal arrival delay (0 - 1000) [ms]: 0

Propagation
Geometric spreading factor (1 - 2): 1
Salinity of the water [ppt]: 35
Temperature of the water [Celsius]: 10
Absorption coefficient: ● Thorp's formula ○ Fisher's formula

Noise
Shipping activity factor (0 -1): 0
Wind speed [m/s]: 0

Message
Enviroment setting completed.

Fig. 7. GUI for environment setting.

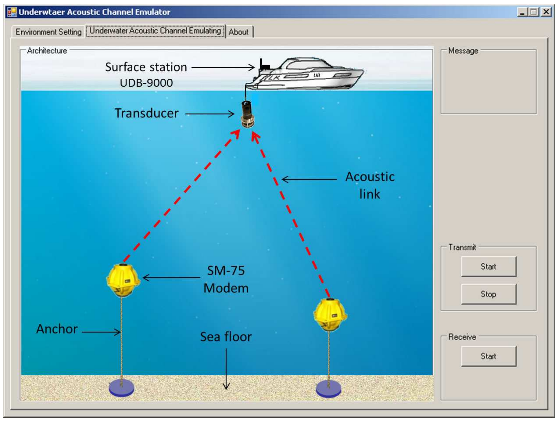


Fig. 8. Underwater acoustic channel emulator in action.

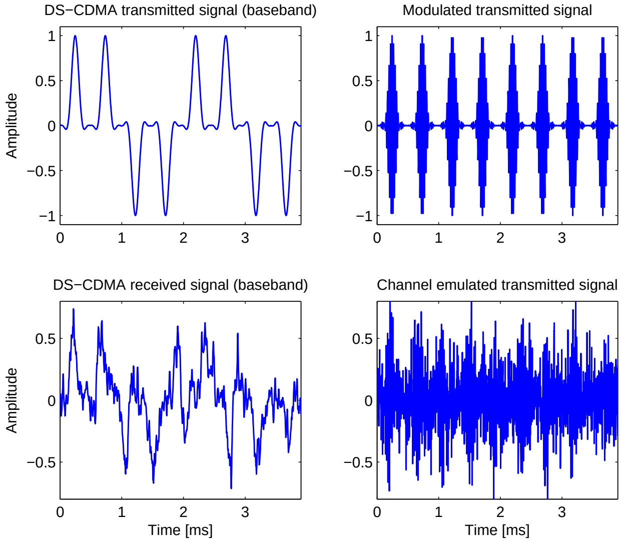


Fig. 9. Transmission and reception of a DS-CDMA 256 bit/s, BPSK raised-cosine pulse shaped acoustic waveform.

B. Graphical User Interface

We have developed a GUI to accompany the underwater acoustic channel emulator. The user may use the environment setting, shown in Fig. 7, to emulate three scenarios: i) single-input-single-output (SISO) under light multipath, ii) SISO under heavy multipath, and iii) MIMO under light multipath environments. Moreover, the user may select the location of the modems and a set of underwater acoustic channel parameters to emulate the real underwater channel conditions for the different scenarios described above. When emulating MIMO scenario, the user may adjust the time delay of the signal transmitted by the second modem such that the two transmitted signals bearing the same information bits arrive at the receiver simultaneously to provide channel diversity [31]. This can also be useful in evaluating the performance of a protocol in scenarios when the two transmitted signals do not necessarily arrive at the receiver simultaneously. An example of an underwater acoustic channel emulation with 2 transmitters and 1 receiver is illustrated in Fig. 8.

The channel emulator also allows generating and transmitting custom defined acoustic waveforms. Figure 9 shows a direct-sequence code-division multiple-access (DS-CDMA) acoustic waveform generated using the third column of a Sylvester-Hadamard matrix of order $L = 8$ as the spreading code. The transmitted DS-CDMA signal is a $256\,\mathrm{bit/s}$, binary phase-shift keying (BPSK), raised-cosine pulse shaped waveform with a roll-off factor $\beta = 0.9$ and a carrier frequency of $11.5\,\mathrm{kHz}$. The transmit power is set to $20\,\mathrm{W}$, which is the maximum transmit power provided by the SM-75 acoustic modem. In this example a SISO scenario in shallow water under heavy multipath is emulated with communication rage of $70\,\mathrm{m}$. The lower left-hand side of Fig. 9 shows the baseband received signal after channel emulation. Note that only one bit, comprised of 8 chips, of the transmitted signal is illustrated and the preamble used for synchronization is not shown.

V. EMULATOR AND EXPERIMENTAL RESULTS

In this section we demonstrate the channel emulation results of the three scenarios discussed in Section IV-B. In addition to that we present actual experimental results for transmission of custom defined acoustic waveform shown in Fig. 9 and compare it with the emulator results.

We will start with the channel emulation results of the three scenarios. A $512\,\mathrm{bit/s}$, BPSK raised-cosine pulse shaped waveform with a roll-off factor $\beta = 0.5$ is transmitted with a transmit power of $2.5\,\mathrm{W}$, which is the second lowest transmit power level provided by the SM-75 acoustic modem. A total

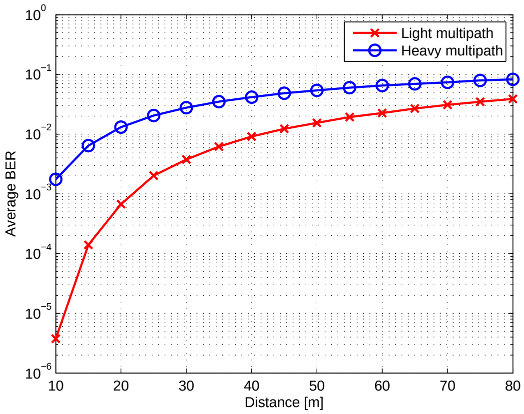


Fig. 10. SISO transmission with different transmission distances.

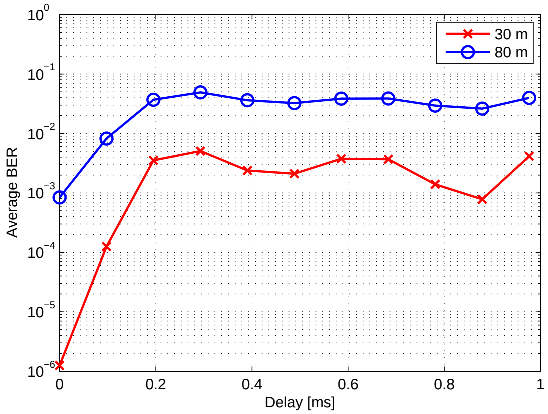


Fig. 11. MIMO transmission with different second signal arrival delay time.

of $1.25\,\mathrm{MByte}$ of data are transmitted in each scenario, and they are decoded by minimum distance decoder. In Fig. 10, we demonstrate the SISO scenario under light and heavy multipath environments for different transmission distances. According to the Thorp's formula [32] the path loss gradually grows as the transmission distance increases, which results in an increase in the average bit error rate (BER). As anticipated, SISO scenario under heavy multipath environment leads to a higher average BER compared to light multipath environment. The heavy multipath environment is an acoustic channel in which the CIR is of relatively longer length compared to light multipath environment. The multipath is modeled as a tap-delay line with CIR expressed as [33]

$$c(\tau, t) = \sum_{p} A_p(t)\delta(\tau - \tau_p(t)), \tag{1}$$

where $A_p(t)$ and $\tau_p(t)$ denote time-varying path amplitude and time-varying path delay, respectively.

In Fig. 11, we present emulation results for the MIMO scenario and evaluate the average BER performance with respect to different arrival delay time of the two transmitted signals at the receiver side. The range of the two transmitters from the receiver are first set at $30\,\mathrm{m}$ then at $80\,\mathrm{m}$. As expected the average BER of $30\,\mathrm{m}$ range is lower than the $80\,\mathrm{m}$ case. Moreover, the average BER increases as the inter-arrival delay time of the signal transmitted by the second modem increases. We may observe that when the two signals inter-arrival delay time is greater than the symbol duration, $0.2\,\mathrm{ms}$, the inter-symbol-interference (ISI) is the dominant factor.

Real underwater experiments were conducted inside the diving pool of the Alumni Arena at the University at Buffalo in order to validate the performance of the proposed channel emulator. The experimental setup is shown in Fig. 12. The dimensions of the diving pool are $22\ \mathrm{m} \times 16\ \mathrm{m}\ \times 4.9\ \mathrm{m}$ (length, width and depth). On one side of the pool an acoustic transduce, connected to the surface station (UDB-9000) and controlled by a laptop through RS-232 interface, was submerged to a depth of $2\ \mathrm{m}$ from the water surface. On the far end of the pool an SM-75 modem with a data recorder was immersed again to a depth of $2\,\mathrm{m}$ from the surface of the pool. The distance between the transmitter and the receiver was $20\ \mathrm{m}$.

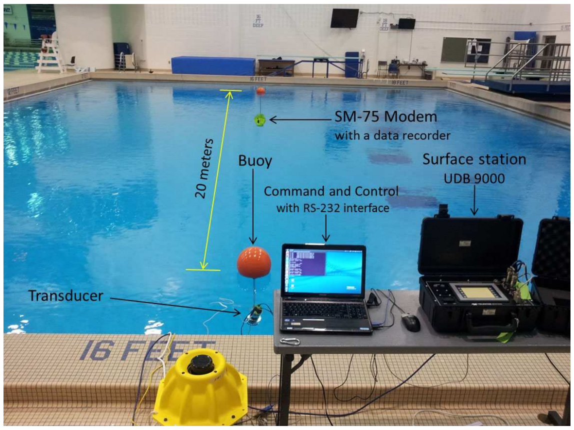


Fig. 12. Experimental setup in the diving pool of University at Buffalo.

A custom defined DS-CDMA acoustic waveform, discussed is Section IV-B and shown in Fig. 9, is transmitted in underwater by the transducer and recorded by the SM-75 modem equipped with a data recorder. Before the actual data transmission a chirp signal of duration $100\,\mathrm{ms}$ sweeping the bandwidth from $10\ \mathrm{Hz}$ to $2.6\ \mathrm{kHz}$ was used as a preable for synchronization purposes followed by a guard-band of $50\ \mathrm{ms}$. A total of $1.25\ \mathrm{kByte}$ of data was transmitted and each experiment was repeated 10 times with different transmit power levels. The transmit power levels provided by the SM-75 modem are from $-21\ \mathrm{dB}$ $(1.78\ \mathrm{W})$ to $0\ \mathrm{dB}$ $(20\ \mathrm{W})$ with $3\ \mathrm{dB}$ increments.

Conventional RAKE-matched-filter (RAKE-MF) was used to decode the transmitted information bits. The average BER performance for experimental and emulation results with different transmit power level is shown in Fig. 13. As the transmit power level increases more bits are decoded correctly and hence the average BER decreases. We may observe that the BER performance for the real experimental results is slightly worse than the emulator results due to the severe multipath effect generated by highly reflective surface of the walls and bottom of the diving pool, which results in inter-symbol-interference. We expect the performance to be closer to the emulation results when conducting the experiments in actual

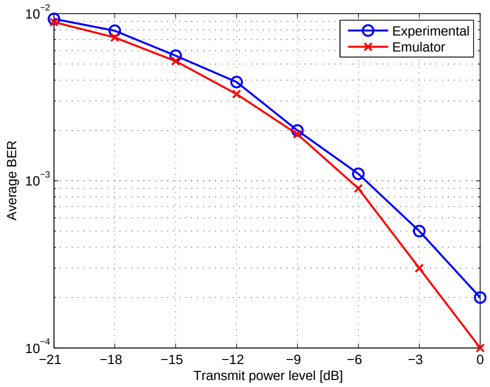


Fig. 13. Average BER performance for experimental and emulation results of DS-CDMA transmission in underwater with different transmit power levels.

shallow water environment (eg., lake, sea).

VI. CONCLUSIONS

In this work we presented our ongoing work on developing a reconfigurable underwater networking testbed based on the Teledyne Benthos Telesonar SM-75 modem. The testbed is designed with the objective of allowing researchers and developers to advance research activities in the field of underwater networking and communications through a flexible testbed platform. Moreover, we have developed an underwater acoustic channel emulator that allows the user to perform laboratory controlled experiments. Our underwater acoustic channel emulator allows developers to advance research activities and evaluate different protocol designs without actually deploying the modems in a real underwater environment (eg., lake, sea, ocean). The emulator also provides MU-MIMO links that could be used for developing cooperative diversity protocols.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under grants CNS-1055945 and CNS-1126357.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks (Elsevier)*, vol. 3, no. 3, pp. 257–279, May 2005.
[2] T. Melodia, H. Kulhandjian, L. Kuo, and E. Demirors, *Advances in Underwater Acoustic Networking*. Inc., Hoboken, NJ: In Mobile Ad Hoc Networking: Cutting Edge Directions, Eds. S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, John Wiley and Sons, 2013.
[3] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater Sensor Networks: Applications, Advances, and Challenges," *Phil. Trans. R. Soc. A*, vol. 370, no. 1958, pp. 158–175, January 2012.
[4] Teledyne-Benthos, Acoustic Modems. [Online]. Available: http://www.benthos.com.
[5] A. K. Sadek, W. Su, and K. J. R. Liu, "Multinode Cooperative Communications in Wireless Networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 341–355, January 2007.
[6] LinkQuest, Underwater Acoustic Modem Models. [Online]. Available: http://www.link-quest.com.
[7] EvoLogics, Underwater Acoustic Modems. [Online]. Available: http://www.evologics.de.
[8] DSPComm, AquaComm: Underwater wireless modem. [Online]. Available: http://www.dspcomm.com.
[9] Tritech, Micron Data Modem. [Online]. Available: http://www.tritech.co.uk.
[10] The WHOI Micro-Modem. [Online]. Available: http://acomms.whoi.edu/umodem/.
[11] R. Otnes, T. Jenserud, J. E. Voldhaug, and C. Soldberg, "A Roadmap to Ubiquitous Underwater Acoustic Communications and Networking," in *Proc. of Underwater Acoustic Measurement: Technologies and Results*, Nafplion, Crete, Greece, June 2009.
[12] 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 *Proc. of MTS/IEEE OCEANS 2005*, vol. 2, Washington, D.C., USA, September 2005, pp. 1086–1092.
[13] E. M. Sozer and M. Stojanovic, "Reconfigurable acoustic modem for underwater sensor networks," in *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, USA, September 2006, pp. 101–104.
[14] A. Gray, P. Arabshahi, S. Roy, N. Jensen, L. Tracy, N. Parrish, and C. Hsieh, "Extended Abstract: Tradeoffs and Design Choices for a Software Defined Acoustic Modem: A Case Study," in *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Berkeley, CA, USA, November 2009.
[15] D. Torres, J. Friedman, T. Schmid, and M. B. Srivastava, "Software-defined underwater acoustic networking platform," in *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Berkeley, California, USA, November 2009.
[16] J. A. Rice, R. K. Creber, C. L. Fletcher, P. A. Baxley, D. C. Davison, and K. E. Rogers, "Seaweb Undersea Acoustic Nets," *Biennial Review 2001, SSC San Diego Technical Document TD 3117*, pp. 234–250, 2001.
[17] NATO S&T Organization: Centre for Maritime Research and Experimentation. [Online]. Available: http://www.cmre.nato.int/.
[18] J. R. Potter, A. Berni, J. Alves, D. Merani, G. Zappa, and R. Been, "Underwater communications protocols and architecture developments at NURC," in *Proc. of MTS/IEEE OCEANS 2011*, Santander, Spain, June 2011, pp. 1–6.
[19] M. A. Rella, A. Maguer, R. Stoner, D. Galletti, and E. Molinari, "NURC within glider sensors calibration, validation and monitoring facilities," in *Proc. of MTS/IEEE OCEANS 2011*, Santander, Spain, June 2011.
[20] R. Been, D. T. Hughes, J. R. Potter, and C. Strode, "Cooperative anti-submarine warfare at NURC moving towards a net-centric capability," in *Proc. of MTS/IEEE OCEANS 2010*, Sydney, Australia, May 2010.
[21] Z. Peng, S. Le, M. Zuba, H. Mo, Y.Zhu, L. Pu, J. Liu, and J.-H. Cui, "Aqua-TUNE: A testbed for underwater networks," in *Proc. of MTS/IEEE OCEANS 2011*, Santander, Spain, June 2011, pp. 1–9.
[22] Z. Peng, Z. Zhou, J.-H. Cui, and Z. Shi, "Aqua-Net: an Underwater Sensor Network Architecture: Design, Implementation, and Initial Testing," in *Proc. of MTS/IEEE OCEANS 2009*, Biloxi, Mississippi, USA, October 2009.
[23] C. Petrioli, R. Petroccia, J. Shusta, and L. Freitag, "From underwater simulation to at-sea testing using the ns-2 network simulator," in *Proc. of MTS/IEEE OCEANS 2011*, Santander, Spain, June 2011, pp. 1–9.
[24] SUNSET: Sapienza University Networking framework for underwater Simulation, Emulation and real-life Testing. [Online]. Available: http://reti.dsi.uniroma1.it/UWSN_Group/.
[25] The VINT Project, The Network Simulator Manual. [Online]. Available: http://www.isi.edu/nsnam/ns/.
[26] Gumstix Inc. [Online]. Available: http://www.gumstix.com.
[27] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, "DESERT Underwater: an NS-Miracle-based framework to DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols," in *Proc. of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May 2012.
[28] The Network Simulator - NS-Miracle. [Online]. Available: http://telecom.dei.unipd.it/pages/read/58/.
[29] L. Freitag, K. Ball, J. Partan, E. Gallimore, S. Singh, and P. Koski, "Underwater Acoustic Network Testbed," in *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Seattle, WA, USA, December 2011.
[30] T. Melodia, S. Batalama, D. Pados, W. Su, J. Atkinson. UW-Buffalo: An Underwater Acoustic Testbed at the University at Buffalo. [Online]. Available: http://www.eng.buffalo.edu/wnesl/underwater_testbed.html.
[31] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.
[32] L. Berkhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics*. New York: Springer, 1982.
[33] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 198–209, April 2008.