Extended Abstract: Development of a Reconfigurable Underwater Acoustic Networking Testbed

Hovannes Kulhandjian, Li-Chung Kuo, and Tommaso Melodia
Department of Electrical Engineering
State University of New York at Buffalo, Buffalo, New York 14260, USA
Email: {hkk2, lkuo2, tmelodia}@buffalo.edu

1. INTRODUCTION

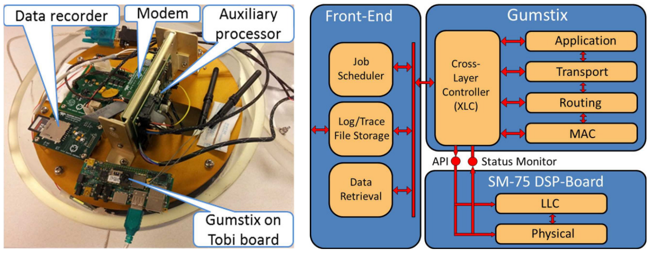
We present our ongoing work on developing a reconfigurable underwater acoustic (UW-A) networking testbed based on the Teledyne Benthos Telesonar SM-75 modem [1]. The SM-75 modem is customized to allow the research community to conduct advanced networking and communication experiments as follows. First, a programmable Gumstix [2] network processor is integrated with the SM-75 modem, as shown in Fig. 1(a), 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 is being implemented on the Gumstix board to enable the definition of complex networking experiments with reconfigurable, cross-layer designed protocol stacks [4]. 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 UW-A channel emulator we developed that allows the user to perform laboratory controlled experiments.

2. TESTBED ARCHITECTURE

The UW-A networking testbed at the University at Buffalo (UW-Buffalo) is designed to provide the research community with a versatile and shared reconfigurable platform to enable experimental evaluation of underwater communications and networking protocols. The testbed is comprised of 11 Telesonar SM-75 modems, one sonar modem and one universal deckbox, UDB-9000, equipped with an acoustic transducer used for monitoring the underwater communications.

2.1 External Controller for SM-75 Modem

The SM-75 modem is interfaced with a Gumstix 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. Moreover, the Gumstix will allow storing and processing data from multiple channels, to allow multiple-input-multiple-output (MIMO) and cooperative signal processing functionalities.



(a) (b)

Figure 1: (a) Open view of the SM-75 electronics with auxiliary processor, data recorder and a Gumstix, (b) Software architecture.

2.2 A Networking API for the SM-75 Modem

In the commercial implementation of the SM-75 modem all networking functionalities reside within the core DSP of the individual modem, and cannot be reconfigured by the end-user. Using the networking API software interface developed by Benthos the native network layer is bypassed and its duties are passed to the Gumstix, whose behavior can be defined by the end user. The API is designed 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 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.

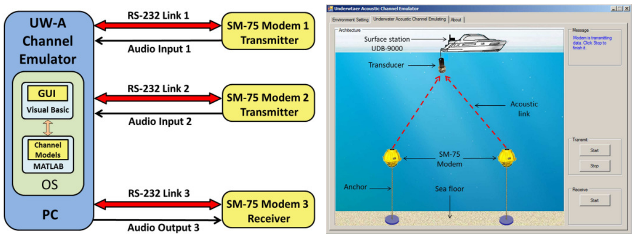
2.3 Networking Protocol Developments

The logic in control of the networking functionalities is implemented in the C language and housed on the Gumstix processor. 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), that is going to be utilized by higher layers of the protocol stack. The communications between the modem and the Gumstix processor is accomplished through the use of MMP.

2.4 Cross Layer Controller Architecture

One of the key requirements in developing optimized communication protocols for underwater networking is to facilitate the use of cross-layer interactions [3]. We accomplish this by designing an additional module, referred in Fig. 1(b) as cross-layer controller (XLC), which controls and regulates information exchange among functionalities handled at different layers of the protocol stack. Through a predefined set of variables and interfaces, the programmer of a module of the protocol stack (e.g., network layer)

WUWNet'12, Nov. 5 - 6, 2012 Los Angeles, California, USA.



(a) (b)

Figure 2: (a) Architecture of the channel emulator with SM-75 modems, (b) UW-A channel emulation in action.

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.

3. UW-A CHANNEL EMULATOR

In underwater acoustic sensor networks (UW-ASNs) it is very difficult to conduct repeatable and realistic experiments through a reconfigurable experimental testbed alone [4]; since the physical layer is strongly dependent on the UW-A channel environment and the exact conditions under which an experiment is conducted. Accordingly, we are developing an UW-A channel emulator that allows conducting laboratory controlled experiments.

3.1 Channel Emulator Architecture

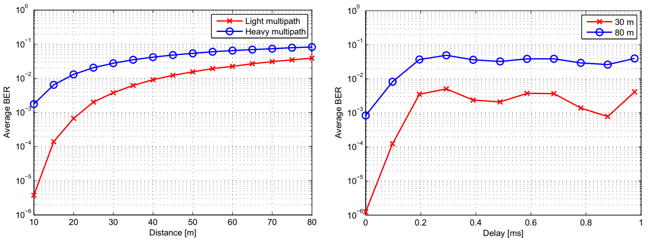
The UW-A channel emulator is interfaced with the SM-75 modems through RS-232 links as shown in Fig. 2(a). The transmitted acoustic signals are first recorded by an audio input device and signal processed by the emulator to account for path loss, noise and multipath spread of a real UW-A channel. Then, the modified acoustic signal is played by an audio output device and recorded by the receiver modem, which decodes the original transmitted data.

3.2 Graphical User Interface (GUI)

We have developed a GUI that allows the user to set up the channel emulation environment. Moreover, it enables the user to emulate single-input-single-output (SISO) and MIMO scenarios under light and heavy multipath channel environments, an example of MIMO senario is shown in Fig. 2(b). The emulator also allows generating and transmitting custom defined acoustic waveforms.

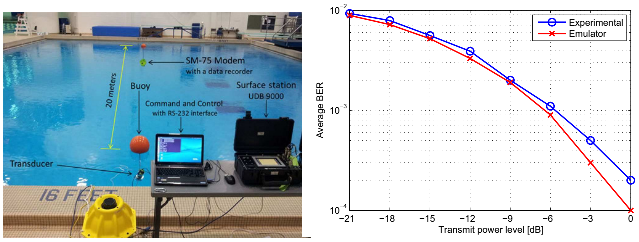
3.3 Emulator and Experimental Results

Channel emulation results for several scenarios including actual experimental results are presented next. A 512 bit/s, BPSK raised-cosine pulse shaped waveform with roll-off factor β= 0.5 is transmitted with a transmit power of 2.5 W. A total of 1.25 MBytes of data are transmitted in each scenario, and they are decoded by minimum distance decoder. In Fig. 3(a), SISO scenario under light and heavy multipath environments are emulated for different transmission range. As the transmission distance increases the path loss gradually grows, which results in an increase in the average bit error rate (BER). As expected, SISO scenario under heavy multipath environment leads to a higher average BER compared to light multipath environment. In Fig. 3(b), 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 (shown in Fig. 2(b)) at the receiver side. The two transmitters from the receiver are first set to 30 m then, to 80 m. As expected, the average BER at 30 m range is lower than in the 80 m case. Moreover, the average BER increases as the inter-arrival delay time of the signal transmitted by the second modem increases and saturates when the delay time is greater than 0.2 ms.



(a) (b)

Figure 3: (a) SISO with different transmission distances, (b) MIMO with different signal arrival delay time.



(a) (b)

Figure 4: (a) Experimental setup in the diving pool of University at Buffalo, (b) Average BER performance for experimental and emulation results of DS-CDMA transmission with different Tx power.

Real underwater experiments were conducted inside the diving pool of the Alumni Arena at the University at Buffalo to validate the performance of the proposed channel emulator. The experimental setup is shown in Fig. 4(a). The dimensions of the diving pool are $22\text{m} \times 16\text{m} \times 4.9\text{m}$ (length, width and depth). On one side of the pool a transducer, connected to the surface station and controlled by a laptop through the RS-232 interface, was submerged to a depth of 2 m from the water surface. On the other side of the pool an SM-75 modem was immersed again to a depth of 2 m. The distance between the transmitter and the receiver was 20 m. A custom defined, binary phase-shift keying (BPSK), DS-CDMA acoustic waveform of 256 bit/s was transmitted by the transducer and recorded by the SM-75 modem equipped with a data recorder. The DS-CDMA waveform was generated using a spreading code extracted from Sylvester-Hadamard matrix of order, $L = 8$. A total of 1.25 kBytes 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 range from -21 dB (1.78 W) to 0 dB (20 W). Conventional RAKE-matched-filter was used to decode the transmitted bits. The average BER performance for experimental and emulation results with different transmit power level is shown in Fig. 4(b). As the transmit power increases more bits are correctly decoded and hence, the average BER decreases. 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 surfaces of the pool.

Acknowledgment

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

4. REFERENCES

[1] Teledyne-Benthos, Acoustic Modems. http://www.benthos.com.
[2] Gumstix Inc. [Online]. Available: http://www.gumstix.com.
[3] H. Kulhandjian, L. Kuo, T. Melodia, D. A. Pados, and D. Green. Towards Experimental Evaluation of Software-Defined Underwater Networked Systems. In *Proc. of IEEE UComms*, Sestri Levante, Italy, September 12-14, 2012.
[4] T. Melodia, H. Kulhandjian, L. Kuo, and E. Demirors. *Advances in Underwater Acoustic Networking.* In Mobile Ad Hoc Networking: Cutting Edge Directions, Eds. S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, John Wiley and Sons, Inc., Hoboken, NJ, 2013.