SEANet: A Software-Defined Acoustic Networking Framework for Reconfigurable Underwater Networking

Emrecan Demirors, Bharatwaj G. Shankar, G. Enrico Santagati, and Tommaso Melodia
Department of Electrical and Computer Engineering
Northeastern University, Boston, MA, U.S.A.
{edemirors, shankar, santagati, melodia}@ece.neu.edu

ABSTRACT

As of today, Underwater Acoustic Networks (UANs) are heavily dependent on commercially available acoustic modems. While commercial modems are often able to support specific applications, they are typically not flexible enough to satisfy the requirements of next-generation UANs, which need to be able to adapt their communication and networking protocols in real-time based on the environmental and application conditions. To address these needs, we present SEANet (Software-dEfined Acoustic Networking), a modular, evolving software-defined framework for UAN devices that offers the necessary flexibility to adapt and satisfy different application and system requirements through a well-defined set of functionalities at the physical, data-link, network, and application layers of the networking protocol stack. SEANet is based on a structured modular architecture that enables real-time reconfiguration at different layers, provides a flexible platform for the deployment of new protocol designs and enhancements, and ensures software portability for platform independence. Moreover, we present a prototype of a low-cost, fully reconfigurable underwater sensing platform that implements the SEANet framework, and discuss performance evaluation results from water tank tests.

Keywords

Underwater acoustic networks, software-defined acoustic networks, reconfigurable platforms.

1. INTRODUCTION

Underwater acoustic networking is the enabling technology for many military and commercial applications, including, but not limited to, underwater surveillance and monitoring, oceanographic data collection, and offshore exploration [1]. While UANs are increasingly attracting interest and are becoming widely used, the application and system needs are also evolving and becoming increasingly complex.

The underwater acoustic (UW-A) channel is notoriously characterized by high path loss, noise, multipath, high propagation delay, and severe Doppler spread. These challenges result in highly *temporally* and *spatially* variable channel conditions. As a consequence, a one-size-fits-all UAN solution is inadequate since next-generation UAN devices need to have the necessary flexibility and capability to reconfigure and adapt their communication and networking protocols in real time based on the environmental and application conditions to guarantee functionality and high performance of the network at all times.

As of today, the majority of existing UANs are based on commercially available acoustic network devices. While these devices may adequately address the needs of specific UAN applications, they cannot guarantee the flexibility and evolvability needed by next-generation UANs because of their inherently fixed-hardware designs and proprietary protocol solutions. Furthermore, size, cost, and energy-efficiency limitations make UANs deployments prohibitive in terms of cost, time, and sustainability.

To address these needs, in this paper, we introduce SEANet (Software-dEfined Acoustic Networking), a modular, evolving software-defined framework for UAN devices that offers the necessary flexibility to adapt and satisfy different application and system requirements through its well-defined set of functionalities at the physical, data-link, network, and application layers. SEANet provides a structured modular architecture that enables real-time cross-layer reconfiguration capabilities, offers a flexible platform to facilitate new protocol designs and enhancements, and provides software portability for platform-independency.

Unlike state-of-the-art software-based framework solutions for UANs [2–7] that are mainly based on interfacing and exploiting commercially available devices, SEANet proposes to transform a general-purpose processor interfaced with acoustic transducers into a fully-functional UAN device. In this way, SEANet aims to obtain a relatively energy-efficient, small-size, low-cost, and fully-reconfigurable UAN device that can overcome the inherent limitations of commercial devices. To demonstrate this, we design and build a prototype of a low-cost, fully reconfigurable underwater sensing platform that implements the SEANet framework and that we tested in a water test tank.

Acknowledgement: This work is based upon work supported in part by the US National Science Foundation under grants CNS-1422874 and CNS-1126357.

WUWNET '15, October 22-24 2015, Washington DC, USA

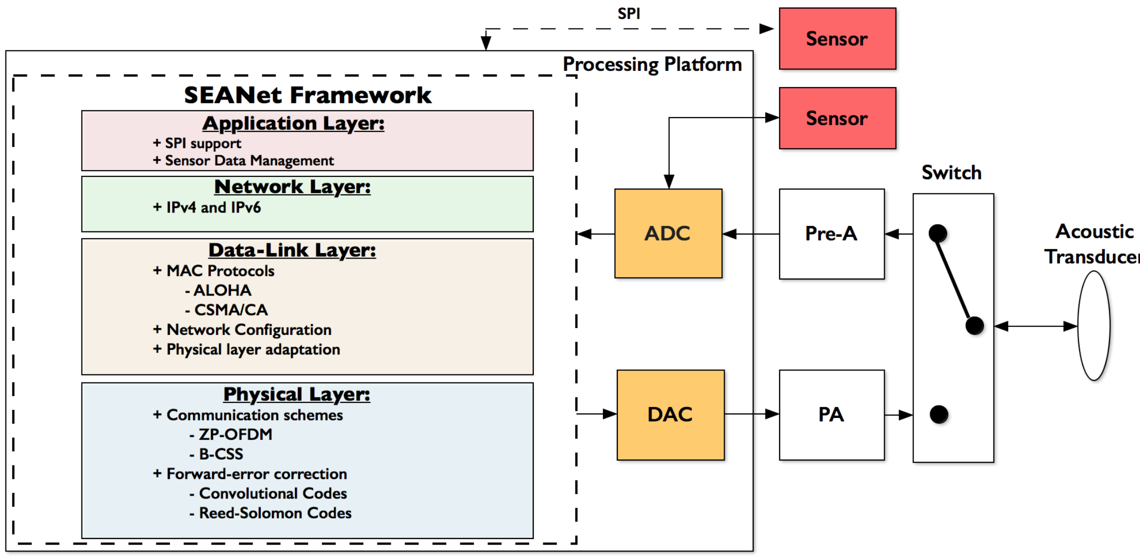

Figure 1: SEANet framework architecture.

The rest of the paper is organized as follows. In Section 2, we describe the general architecture of the SEANet framework. In Section 3, we present the design, implementation, experimental evaluation, and architectural considerations of the proposed SEANet prototype. In Section 4, we briefly review related work while in Section 5 we draw the main conclusions.

2. SEANet ARCHITECTURE

In this section, we describe the architecture of the SEANet framework, which is depicted in Fig. 1. The framework incorporates four main blocks, each corresponding to a specific layer functionality, i.e., physical layer, data-link layer, network layer, and application layer.

2.1 Physical Layer

SEANet provides physical layer libraries for defining different communication schemes and forward error correction (FEC) techniques.

2.1.1 Communication Schemes

SEANet incorporates two communication schemes, i.e., Zero-Padded Orthogonal Frequency-Division-Multiplexing (ZP-OFDM) and Binary Chirp Spread-Spectrum (B-CSS). In addition to that, SEANet includes reusable primitive building blocks, e.g., symbol mapping, Fast Fourier Transform (FFT), filters, which enable the implementation of new physical layer protocols that can be obtained as the result of rearrangement of these blocks.

ZP-OFDM. OFDM is a widely used communication scheme for UW-A systems [8, 9], thanks to its robustness against frequency-selective channels with long delay spreads. Specifically, SEANet adopts an OFDM scheme with zero-padding, where each OFDM symbol is followed by padded zeros, due to its energy-efficiency compared to its counterparts, i.e., cyclic-prefixing (CP). In each OFDM symbol, SEANet accommodates K_P pilot subcarriers to be used in the channel estimation and fine symbol synchronization. Moreover, it also includes K_N null subcarriers for Doppler estimation and K_D data subcarriers, which are conventionally modulated with either M-Phase-Shift-Keying (PSK) or M-Quadrature-Amplitude-Modulation (QAM), for data transmission. SEANet uses a packet format, depicted in Fig. 2, which includes N OFDM symbols preceded with a preamble block that provides packet detection and coarse symbol synchronization. SEANet provides an option to select the preamble block to be either a pseudo noise (PN)-sequence or a chirp signal based on the network and application requirements.

B-CSS. B-CSS is based on chirp signals that are well-known to be resilient against severe multipath and Doppler effects that are the main characteristics of an UW-A channel. B-CSS has been used in UW-A communications, especially in links that require relatively low data rate but high reliability [10, 11] such as feedback links. Furthermore, B-CSS is characterized by a very low complexity correlation-based receiver architecture that dramatically decreases the computational complexity.

2.1.2 Forward Error Correction

SEANet provides a forward error correction FEC functionality based on two different coding schemes, Reed-Solomon (RS) and Convolutional codes.

Reed-Solomon Codes. RS codes are linear block-type error-correction codes designed to correct potential errors caused by channel fluctuations and inter-symbol-interference. An RS encoder converts k information symbols into a n sized symbol block by adding t redundancy symbols. Correspondingly, an RS decoder obtains k information symbols from a n sized symbol block, while being able to correct up to $t/2$ of them.

Convolutional Codes. Convolutional codes are error-correction codes that work with arbitrary sized symbol streams. They generate parity symbols by using a sliding boolean polynomial function to data streams. Convolutional codes can be punctured with different puncturing schemes to decrease the coding overhead and correspondingly reach higher data rates.

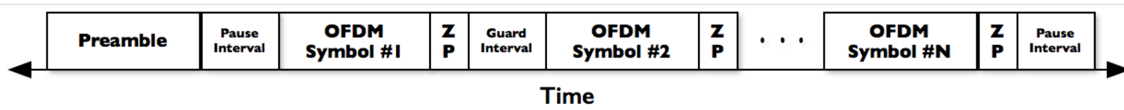

Figure 2: Packet format of ZP-OFDM.

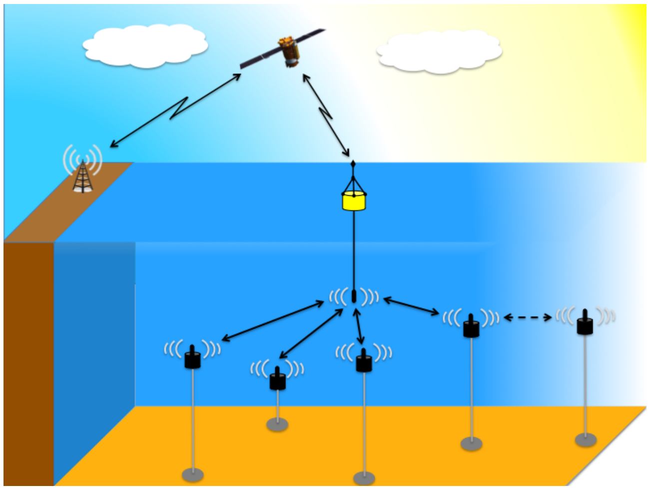

Figure 3: SEANet Hybrid Network Topology. Solid and dashed lines depict centralized and decentralized control, respectively.

2.2 Data-Link Layer

SEANet incorporates a data-link layer design that contains different MAC protocol implementations, network topology configurations, and a physical layer adaptation mechanism.

2.2.1 MAC Protocols

SEANet offers a set of different MAC protocols, including Carrier-Sense-Multiple-Access with Collision-Avoidance CSMA/CA and ALOHA. Moreover, SEANet also provides a set of primitive functions, e.g., retransmission, timer functionalities, check-sum based error control mechanism, and idle listening, that can enable the implementation of different MAC protocol designs.

CSMA/CA. CSMA/CA is a medium access control technique that depends on a carrier sensing mechanism for preventing collisions with on-going transmissions [12]. Specifically, if a node wants to start data transmission, it first senses the medium for a certain amount of time (in the order of inter-frame space duration). When the node senses the medium continuously idle, it starts the transmission process. However, CSMA/CA is known to be less effective because of the long propagation delays in the UW-A channel; it can be a viable option for *very short* or *short* communication links.

ALOHA. ALOHA is a medium access control protocol based on random access [13]. If a node wants to transmit a packet, it accesses the medium without sensing. Each successful transmission is acknowledged by the receiver; otherwise, the transmitter node concludes that a collision has occurred. In this case, the transmitter waits for a random time interval (i.e., backoff time) and retransmits the packet.

2.2.2 Network Topology

SEANet provides support for UAN network devices that can operate in networks with both centralized and decentralized control. In networks with centralized control, one node is assigned to be the central (master) node and it coordinates the rest of the nodes. Typically; the central node may have higher computational and memory resources. Specifically in UANs, the central node might also take the role of gateway between underwater acoustic and terrestrial networks, in which it requires to support RF communications besides its acoustic communication capability. For example, such configuration can be used for UAN applications that require real-time and continuous monitoring, where the data collected in the network is sent to the central node. Subsequently, the central node transfers the collected data to a shore station or a database through its RF communication module. SEANet also supports network configurations with decentralized control, where each node acts as a peer without the control of any central identity. Such configurations can be useful in UAN applications to reach higher ranges or save energy by exploiting multi-hop links. Figure 3 illustrates a hybrid network topology, where both centralized and decentralized control are considered.

2.2.3 Physical Layer Adaptation Mechanism

SEANet incorporates a physical layer adaptation mechanism that provides the UAN devices with the capability to adapt/change in real-time their physical layer scheme, e.g., modulation, FEC coding rate, guard interval size, and symbol duration. Thanks to its software-defined architecture, SEANet can perform cross-layer adaptation and allows network and data-link layers to reconfigure physical layer parameters. As a result, SEANet can define different decision algorithms to adapt its behavior based on the current channel requirements and different application needs.

2.3 Network Layer

SEANet is designed to support IPv4 and IPv6 protocols through the adaptation layer design proposed in [14]. The adaptation layer provides the interoperability of the UAN devices with traditional IP networks. To reach that goal, the adaptation layer provides a set of functionalities, i.e., IP header compression, IP packet fragmentation, which optimize the traditional IPv4 and IPv6 headers for underwater acoustic channels to minimize the overall network delay and energy consumption.

2.4 Application Layer

SEANet is designed to support different operational needs. Specifically, SEANet can configure nodes to operate either as a sink node, which collects data from the network, as a source node, which sends its sensed data to another node, or as a relay node that forwards collected data to another node. Moreover, SEANet provides support for interfacing different sensor units. The sensors can either be interfaced through

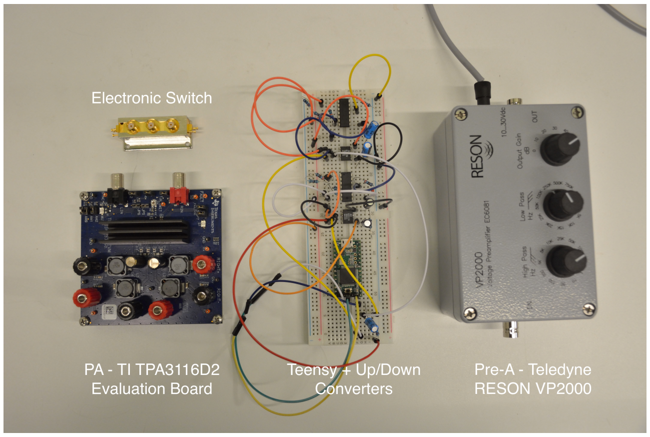

Figure 4: The proposed SEANet prototype.

ADC in the analog domain or through digital domain with standard serial communication protocols, i.e., Serial Peripheral Interface (SPI).

3. SEANet PROTOTYPE

In this section, we illustrate the design and components of a UAN device prototype that implements the SEANet framework. We also present an experimental evaluation and architectural considerations for the proposed prototype.

3.1 Hardware Setup

The SEANet prototype has a modular hardware architecture. It is formed by four permanent modules, i.e., main, acoustic communication, sensor, and battery; and one optional module, i.e., RF communication module, as depicted in Fig. 5. The main module incorporates a processor that executes the SEANet framework, an internal memory unit, and analog-to-digital and digital-to-analog converters. The acoustic communication module contains up and down converters, an electronic switch, an amplification stage that includes power and pre-amplifiers, and an acoustic transducer. The battery module is used to power the prototype and the sensing module may include several different sensors based on the application and system requirements. The RF communication module can be plugged in for enabling the prototype to act as a gateway between UAN and RF network as discussed in Section 2.2.2.

The SEANet prototype provides an evolvable and reconfigurable underwater sensing platform, thanks to its modular design. Each module in the prototype is designed to be swappable since it is connected to others with standard interfaces and has well-described, non-overlapping duties. Therefore, the prototype can support varying hardware requirements dictated by different applications and systems without the need to change design, as further discussed in Section 3.4.

We built a preliminary version of the prototype by using commercial-off-the-shelf (COTS) components, as depicted in Fig. 4. Specifically, we used Teensy 3.1 [15] as the main module with an omnidirectional acoustic transducer, Teledyne RESON TC4013 [16], an audio power amplifier, Texas Instruments TPA3116D2 [17], and a voltage pre-amplifier, Teledyne RESON VP2000 [16]. We included an electronic switch, Mini-Circuits ZX80 − DR230+ [18] to allow a single acoustic transducer to work as transmitter and receiver in a time-division duplex fashion. We also incorporate two voltage-controlled oscillators, ICL8038 [19], and two mixers, AD633 [20], for up- and down-converting the communication signals. Currently, we are employing a power supply instead of a battery unit for experimental purposes. We are also working towards building a compact prototype on a custom PCB design with smaller-size power and pre-amplifier units.

Teensy 3.1. We selected Teensy 3.1, which is a commercially available, 32-bit ARM Cortex-M4 based microcontroller development board. It offers 64 kbytes of RAM, 256 kbytes of Flash Memory, 12 bit digital-to-analog converter (DAC), dual analog-to-digital converter (ADC), and USB connectivity. It has an actual size of 0.7 x 1.4 inches, as illustrated in Fig. 6.

In the process of selecting Teensy 3.1 as the core part of our prototype design, we considered five main factors, i.e., (i) energy-efficiency, (ii) size, (iii) cost, (iv) capabilities, and (v) ease of use (programmability). Based on these factors, we compared different COTS platforms. First, Teensy 3.1 stands out against counterparts like USRP N210 and Raspberry Pi as the most energy-efficient because of its low power consumption. Unlike solutions that execute processes on top of an operating system running on microprocessors, Teensy 3.1 is handling processes on a low-power microcontroller. Second, with the actual size of 0.7 x 1.4 inches, Teensy 3.1 is smaller compared to other microprocessor based platforms. Third, Teensy 3.1 USB development platform costs only $20, which is cheaper than most of the available COTS platforms. Four, Teensy 3.1 offers higher levels of flexibility, because of its microcontroller-based design; and higher memory and computational resources than its base platform Arduino UNO. Moreover, unlike Arduino UNO, it supports Direct Memory Access (DMA), which enables continuos data flow to the DAC without the involvement of the processor. Last, Teensy 3.1 eases prototyping and programming by supporting libraries designed for Arduino and uniquely for Teensy such as the audio library [21]. As a result of all these observations and comparisons, Teensy 3.1 was found to be the ideal choice.

Amplifiers, Converters, and Switch. In the SEANet prototype, we use a class D power amplifier, Texas Instruments TPA3116D2, to extend the communication range. TPA3116D2 is desirable because of its small body size (11.00 mm × 6.20 mm), low price (approximately $2), and operational characteristics, i.e., maximum output power of 100 W, variable gain functionality, and 32 dB maximum gain at 100 kHz. We are currently employing TPA3116D2 with its Evaluation Module as illustrated in Fig. 4. To perform the full-duplex operations with a single acoustic transducer, we incorporate a commercial electronic switch, Mini-Circuits ZX80 − DR230+, which is controlled by Teensy 3.1 through available digital pins. The switch is reported to offer low insertion loss and very high isolation over the frequency range of 0 − 3 GHz. To amplify the received signals we use a voltage preamplifier (PreA), Teledyne RESON VP2000, that provides low-noise performance in selected frequencies through its bandpass filters and adjustable gain selector. Furthermore, we also employ voltage controlled oscillators, ICL8038, and mixers, AD633, for up- or down-conversion of the signals.

Acoustic Transducer. We selected an acoustic trans-

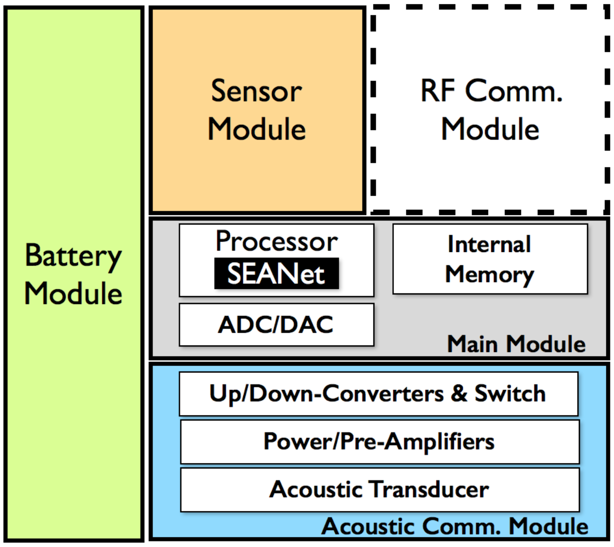

Figure 5: The SEANet prototype contains four permanent and one optional (dashed line) module.

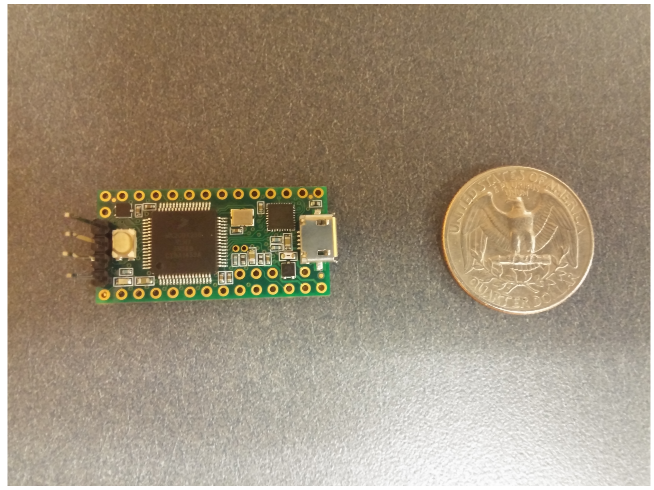

Figure 6: Teensy 3.1 USB Development Board.

ducer, specifically a receiver hydrophone, Teledyne RESON TC4013, that can offer operational frequency range from 1 Hz to 170 kHz. The transducer is reported to provide a flat receiving sensitivity of −211 [dB re 1V /μPa at 1 m] over the operational frequency range, and a maximum transmit sensitivity of 130 [dB re 1μPa/V at 1 m] at 100 kHz. The transducer produces directivity pattern that is omnidirectional in the horizontal axis and 270° in the vertical axis.

3.2 Software Setup

We use the C++ programming language with Teensyduino [22], which is a software add-on for the Arduino integrated development environment (IDE). Teensy 3.1 provides support for Arduino libraries as well as other libraries that are specially designed for Teensy, e.g., audio library. Since the audio library was originally designed to build streaming audio projects, we can exploit it to implement SEANet's physical layer signal processing functionalities.

The audio library is able to record, process, and playback audio signals with a sampling rate of 44.1 kHz. It provides a set of objects for digital signal processing, e.g., finite-impulse-response (FIR) filters, Fast Fourier Transform (FFT), and waveform generators. Furthermore, the audio library provides the ability and space to create new objects for additional digital signal processing functionalities. By connecting these object to each other, different processing chains that correspond to different physical layer designs can be built. The audio library allows its objects to flow data inside the chain in the form of fixed size blocks that contain 128 samples (2.9 ms) of audio data. Considering that each object should finish its processing operations within the 2.9 ms deadline to avoid bottlenecks and having continuous audio data, we implement computational intensive processing blocks, e.g., FEC, channel estimation, doppler estimation, outside of the audio library.

Figure 7 depicts the block diagram of the software implementation of the SEANet framework. Specifically at the physical layer, the implementation of the ZP-OFDM communication scheme is presented. Based on that, the physical layer transmitter chain starts with the FEC encoder object. This object applies convolutional coding operation to the information bits coming from the data-link layer. Subsequently, the coded bits are passed to the symbol mapping object that maps them into symbols, i.e., complex in-phase/quadrature signals, according to a modulation scheme selected by the data-link layer. The symbols are then placed into the subcarrier positions based on a predefined scheme to form OFDM symbols. Afterwards, IFFT and zero padding operations are performed on each OFDM symbol by corresponding objects. Zero-padded OFDM symbols are then passed to the PN sequence and guard interval adding object. Here, PN and guard sequences are inserted in-between the OFDM symbols to form the ZP-OFDM packet structure that is depicted in Fig. 2. The ZP-OFDM packets are then passed to the up-mixer object, where they are up-converted into an intermediate frequency to transform the complex in-phase/quadrature signals into the real outputs. The output of the up-mixer block is later passed to the audio output object, where it is sent to DAC to be converted into the analog signals.

At the receiver side, the audio input object inputs the digital signal that are coming from the ADC into the receiver chain. The received digital signals are first injected into a high-pass filter (HPF) object to eliminate dc-offset and low-frequency noise. The filtered signals are processed by a packet detector object that is based on an energy-level check. After packet detection, the incoming packet is down-converted into a baseband signal by the down-mixer. The output of the down-mixer object, which is in the form of complex in-phase/quadrature signals, is then passed to the low-pass filter (LPF) object to eliminate higher frequency harmonics. Then, the synchronization object performs a preamble correlation to obtain packet synchronization. After packet synchronization, the block partition object partitions input streams into OFDM blocks. Each block (symbol) is then passed through the Doppler scale estimation and channel estimation and equalization operations. The equalized symbols are later mapped onto the bits and decoded with the FEC Decoder before being passed to the data-link layer.

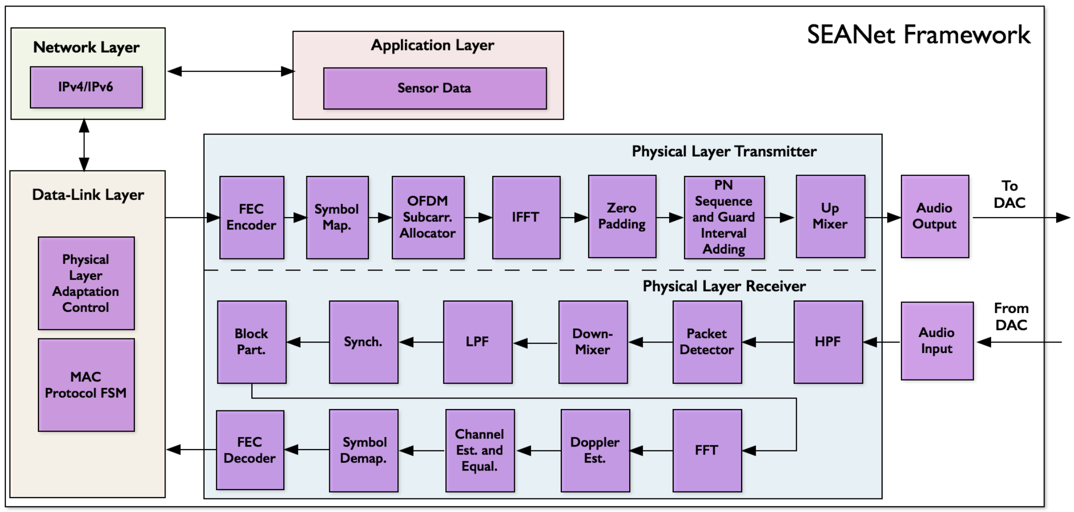

Figure 7: Block Diagram of the software implementation of the SEANET framework.

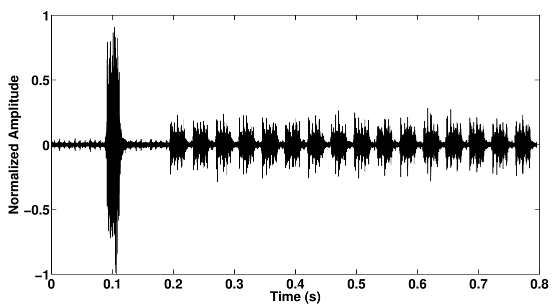

Figure 8: Packet structure.

The data-link layer incorporates finite state machines to implement ALOHA and CSMA/CA MAC protocols similar to [23]. Moreover, it also defines a physical layer adaptation mechanism to select FEC coding rate and modulation schemes based on a user-defined decision algorithm. The network layer implements the adaptation layer that is described in Section 2.3 to support IP operations.

3.3 Experimental Evaluation

We conducted a series of experiments in a water test tank of dimensions 228.6 cm × 55.9 cm × 73.7 cm. Our objective was to (i) assess the functionality of the first SEANet prototype; (ii) evaluate the physical layer implementation and observe its performance. To that end, we use ZP-OFDM signals in the packet form that is described in the Section 2.1. Specifically, we generated signals that occupy a bandwidth of $B = 11.025$ kHz, which can be reliably supported with the sampling rate of the audio library (44.1 ksamples/s). We selected a carrier frequency of $f_c = 121.025$ kHz, since that results in the highest transmit power of the proposed prototype based on the characteristics of the acoustic transducer and amplifiers. We introduced guard intervals of 14.5 ms for each OFDM block. We used $K = 256$ total subcarriers, where $K/4$ of them are used as pilot subcarriers and each subcarrier is either modulated with BPSK or QPSK and is either coded with 1/2 rate convolutional codes or not coded at all.

First, the received signal from the ADC was recorded without any processing to confirm that the prototype is capable of generating the proposed packet structure correctly. Fig. 8 illustrates that the actual packet structure matches the proposed packet structure depicted in Fig. 2. After verfiying the packet structure, we obtained BER results for different modulation schemes, i.e., BPSK and QPSK, and convolutional coding rates, i.e., 1/2 in the implemented ZP-OFDM scheme. Figure 9 illustrates the BER performance of the proposed prototype for different SNR values. As expected, BPSK modulation gives better BER performance than the QPSK modulation and coding improves BER performance at the higher SINR values. We also observe that, despite the unfavorable environment of a water tank that poses excessive number of channel taps (in the order of 300), the prototype can still reach a data rate of 10 kbit/s with a BER performance of 5×10^{-2}. To validate our claim, we observed the estimated channel in the water test tank. As illustrated in Fig 10, the channel creates severe multipath effect, with spread larger than 5.35 ms. Comparing our current results with the results obtained in a similar water tank in our previous work [11], we can expect to reach three orders of magnitude better BER results in a lake or sea environment, where the multipath effect is less severe with the same setup.

3.4 Architectural Considerations

Computational Capacity. The current version of the SEANet prototype incorporates a Teensy 3.1 because of the reasons discussed in Section 3.1. While Teensy's Cortex − M4 microcontroller can support the current implementation needs of the SEANet framework, it may be changed based on the application and system requirements. Specifically, in scenarios where computational capacity can be sacrificed for better energy-efficiency and smaller size, a low-area, low-power, low-cost microcontroller, e.g., Cortex − M0, can be used with the SEANet framework. On the other hand, for a

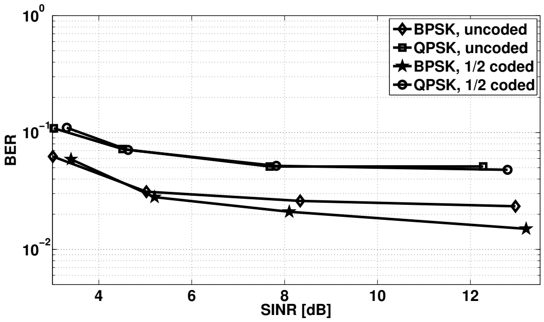

Figure 9: BER versus SNR for BPSK and QPSK mapping.

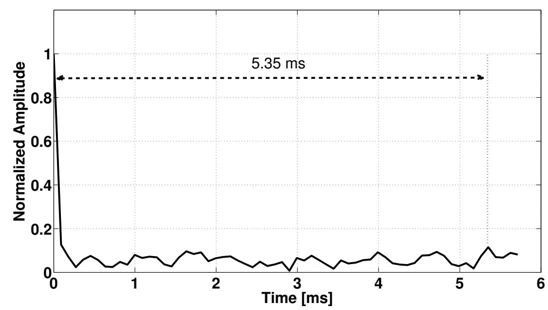

Figure 10: Estimated channel in the water test tank.

system that needs higher computational capacity to perform multiple computationally intensive data processes (e.g., finite impulse response (FIR) filters, FFT), a higher computational capacity microcontroller, e.g., Cortex $-$ M7, would be a better candidate. In existing commercial devices and experimental prototypes, the aforementioned switch between processor/controller units might be either infeasible or prohibitive in terms of time and effort. On the contrary, the SEANet prototype supports such a switch, in the hardware domain, through its modular structure and standard interfaces as described in Section 3.1, and in the software domain, with the software portability of the SEANet framework.

Data Rate Support. The current implementation of the SEANet prototype is able to support data rates up to 41 kbit/s by occupying a bandwidth of 22.05 kHz. While such a data rate is sufficient for most UAN applications, it is significantly lower than what the proposed hardware architecture can reach as an upper limit. The main factor that prevents the proposed prototype to reach higher data rates is the audio library, which is used in the physical layer implementation of the framework. Specifically, the audio library imposes that the ADC (which natively supports up to 818.330 kHz) sample at 44.1 kHz and accordingly determines the maximum bandwidth that can be used through Nyquist rate as 22.05 kHz. Considering that the sampling rate limitation can be eliminated by replacing the audio library with another custom DSP library, which is a feasible solution thanks to the modular architecture of SEANet, the data rate of the SEANet prototype can be increased significantly. Given the current hardware specifications, we estimate that the prototype should be able to reach a data rate of 250kbit/s by utilizing a bandwidth of 100 kHz. Moreover, such an increase in the sampling rate can remove the need of having up and down converter units at the acoustic communication module of the prototype. Consequently, a more robust and compact prototype can be achieved.

4. RELATED WORK

Recent years have witnessed a growing interest in software-defined radio (SDR) solutions for various RF and in-air ultrasonic applications. SDRs enable the implementation of cognitive, flexible, and reconfigurable wireless networks [23, 24]. Considering the capabilities and features of SDR in RF and in-air ultrasonic networks and the needs of next-generation UAN devices, i.e., flexibility and reconfigurability, SDR stands out as a promising solution for next-generation UANs. To that end, there have been some studies [25, 26] that forecast the potential benefits of a possible paradigm shift from hardware-based to SDR based networking devices.

In light of the anticipated benefits, there have been research efforts towards developing software-defined networking frameworks for UANs. SUNSET [2] and DESERT [3] are two software-defined networking frameworks that specifically simulate, emulate and test policies by building on the open-source network simulators ns$-$2 and ns$-$2 miracle and interfacing with COTS acoustic modems. In the SUNRISE project [4], the goal is to create a federated underwater communication infrastructure by using a software-defined open-architecture modem. Similarly in [5], the authors propose a framework with multi-layer architecture to deploy networks that incorporate cross-layer optimizations by using WHOI micro-modems [27] and Teledyne Benthos modems [28]. In UNET [6], the authors introduce an agent-based protocol stack to ease the transition from simulation studies to real-world deployments, specifically with supported acoustic modems. In [7], the authors proposes a framework that considers a cross-layer design with general modules, which will be determined based on a generalization of a classical OSI communications stack, to ease the adaptation of existing modems with software-defined architectures. While the SEANet framework follows similar goals, it proposes to transform a general-purpose processor interfaced with acoustic transducers into a fully-functional UAN device, instead of interfacing with COTS acoustic devices.

In recent years, researchers proposed several studies that consider the hardware development of new experimental modems with partially or fully software-defined protocol stacks. In [29], the authors propose a USRP-based [30] network device that exploits open-source software tools, i.e., GNU Radio, TinyOS, and TOSSIM, for implementing physical and data-link layer functionalities. Similar works in [26, 31, 32] introduce new modem prototypes, that are based on FPGA/DSP or FPGA-only solutions in alliance with software-defined physical and data-link layer for reaching parameter adaptation. In [11, 33], the authors present a software-defined modem based on USRP device that offers real-time reconfiguration at the physical layer. While SEANet is based on the same idea of using custom developed hardware for software-defined UANs, SEANet goes beyond the lower-layer protocol building blocks. We proposed a framework that offers functionalities that span physical, data-link, network, and application layer.

5. CONCLUSIONS

We introduced a modular, evolving software-defined framework, called SEANet, to provide flexibility to adapt and satisfy different application and system requirements of next-generation UANs. SEANet offers a structured modular architecture that spans physical, data-link, network, and application layers. Moreover, SEANet introduces a structured modular architecture, which enables real-time reconfiguration at different layers, provides a flexible platform for the deployment of new protocol designs and enhancements, and provides software portability for platform independence. We designed and built a prototype of a low-cost, fully reconfigurable underwater sensing platform that implements the SEANet framework, and discussed performance evaluation results from water tank tests.

6. REFERENCES

[1] T. Melodia, H. Kulhandjian, L. Kuo, and E. Demirors, "Advances in Underwater Acoustic Networking," in *Mobile Ad Hoc Networking: Cutting Edge Directions*, second edition ed., S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Inc., Hoboken, NJ: John Wiley and Sons, 2013, pp. 804–852.

[2] C. Petrioli, R. Petroccia, and D. Spaccini, "SUNSET Version 2.0: Enhanced framework for simulation, emulation and real-life testing of underwater wireless sensor networks," in *Proc. of ACM Intl. Conf. on UnderWater Networks and Systems (WUWNet)*, Kaohsiung, Taiwan, 2013.

[3] 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*, Yeosu, Republic of Korea, May 2012.

[4] "SUNRISE: Building the internet of underwater things," 2015. [Online]. Available: http://fp7-sunrise.eu/

[5] 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*, Biloxi, MS, October 2009.

[6] M. Chitre, R. Bhatnagar, and W.-S. Soh, "UnetStack: an Agent-based Software Stack and Simulator for Underwater Networks," in *Proc. of MTS/IEEE OCEANS*, St. John's, Newfoundland, September 2014.

[7] J. Potter, T. Furfaro, A. Vermeij, and N. Jourden, "Software Defined Open Architecture Modem Development at CMRE," in *Proc. Underwater Communications and Networking Workshop (UComms)*, Sestri Levante, Italy, September 2014.

[8] S. Zhou and Z.-H. Wang, *OFDM for Underwater Acoustic Communications.* John Wiley and Sons, Inc., 2014.

[9] A. Radosevic, R. Ahmed, T. Duman, J. Proakis, and M. Stojanovic, "Adaptive OFDM modulation for underwater acoustic communications: Design considerations and experimental results," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 357–370, April 2014.

[10] L. LeBlanc, M. Singer, P.-P. Beaujean, C. Boubli, and J. Alleyne, "Improved chirp FSK modem for high reliability communications in shallow water," in *Proc. of MTS/IEEE OCEANS*, September 2000.

[11] E. Demirors, G. Sklivanitis, G.E. Santagati, T. Melodia and S. N. Batalama, "Design of A Software-defined Underwater Acoustic Modem with Real-time Physical Layer Adaptation Capabilities," in *Proc. of ACM Intl. Conf. on Underwater Networks & Systems (WUWNet)*, Rome, Italy, November 2014.

[12] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, December 1975.

[13] N. Abramson, "THE ALOHA SYSTEM: another alternative for computer communications," in *Proc. of the Fall Joint Computer Conference (FJCC), AFIPS*, Houston, TX, November 1970.

[14] Y. Sun and T. Melodia, "The Internet Underwater: An IP-compatible Protocol Stack for Commercial Undersea Modems," in *Proc. of ACM Intl. Conf. on UnderWater Networks and Systems (WUWNet)*, Kaohsiung, Taiwan, November 2013.

[15] "Teensy USB Development Board," 2015. [Online]. Available: https://www.pjrc.com/teensy

[16] "Teledyne Reson products," 2014. [Online]. Available: http://www.teledyne-reson.com/products/

[17] "Texas Instruments TPA3116D2 Evaluation Module," 2015. [Online]. Available: http://www.ti.com/tool/tpa3116d2evm

[18] "Mini-Circuits: RF/IF & Microwave components DC to 40 GHz." 2014. [Online]. Available: http://www.minicircuits.com

[19] "ICL8038 Datasheet," 2015. [Online]. Available: http://www.intersil.com/content/dam/Intersil/documents/icl8/icl8038.pdf

[20] "AD633 Datasheet," 2015. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD633.pdf

[21] "Teensy Audio Library," 2015. [Online]. Available: https://www.pjrc.com/teensy/td_libs_Audio.html

[22] "Teensyduino," 2015. [Online]. Available: https://www.pjrc.com/teensy/teensyduino.html

[23] E. Demirors, G. Sklivanitis, T. Melodia, and S. N. Batalama, "RcUBe: Real-time reconfigurable radio framework with self-optimization capabilitites," in *Proc. of IEEE Intl. Conf. on Sensing, Communication, and Networking (SECON)*, Seattle, WA, June 2015.

[24] G. E. Santagati and T. Melodia, "U-Wear: Software-defined ultrasonic networking for wearable devices," in *Proc. of IEEE Conference on Mobile Systems, Services and Applications (MOBISYS)*, Florence, Italy, May 2015.

[25] E. Jones, "The application of software radio techniques to underwater acoustic communications," in *Proc. of IEEE OCEANS*, Aberdeen, Scotland, June 2007.

[26] E. M. Sözer and M. Stojanovic, "Reconfigurable acoustic modem for underwater sensor networks," in *Proc. of ACM Intl. Workshop on Underwater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[27] "WHOI Micromodem," 2015. [Online]. Available: http://acomms.whoi.edu/micro-modem/

[28] "Teledyne Benthos Acoustic Modems," 2015. [Online]. Available: https://teledynebenthos.com/product_dashboard/acoustic_modems

[29] Y. Noh, D. Torres, and M. Gerla, "Software-defined underwater acoustic networking platform and its applications," *Ad Hoc Networks (Elsevier)*, 2015.

[30] "USRP products," 2015. [Online]. Available: http://www.ettus.com

[31] N. Nowsheen, C. Benson, and M. Frater, "A high data-rate, software-defined underwater acoustic modem," in *Proc. of MTS/IEEE OCEANS*, Seattle, WA, September 2010.

[32] M. Chitre, I. Topor, and T. Koay, "The UNET-2 modem - An extensible tool for underwater networking research," in *Proc. of MTS/IEEE OCEANS*, Yeosu, Republic of Korea, May 2012.

[33] G. Sklivanitis, E. Demirors, S. N. Batalama, T. Melodia and D. A. Pados, "Receiver Configuration and Testbed Development for Underwater Cognitive Channelization," in *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 2014.