Telecommun Syst
DOI 10.1007/s11235-008-9075-3

Mote-based underwater sensor networks: opportunities, challenges, and guidelines

**Raja Jurdak · Antonio G. Ruzzelli ·
Gregory M.P. O'Hare · Cristina Videira Lopes**

Abstract Most underwater networks rely on expensive specialized hardware for acoustic communication and modulation. This has impeded wide scale deployments of underwater sensor networks and has forced researchers to use simulations to investigate these systems. To address these issues, this paper examines a system that integrates off-the-shelf acoustic hardware built-in to sensor modules with software modems for establishing underwater acoustic links. Because the hardware in our system is readily available, we have conducted several rounds of field experiments to evaluate it. Building on our recent field experiments in a river, canal, pond, and swimming pool, this paper outlines the technical and logistical challenges for deploying software-driven underwater sensor networks. The design choices include methods for signal modulation at the sender, and symbol synchronization, signal filtering, and signal demodulation at the receiver. We also discuss higher layer communication protocol issues, with a focus on cross-layer optimization, as well as practical solutions to logistical deployment challenges, such as waterproofing and casing, calibration, and fouling. The design guidelines in this paper lay the groundwork for further development of software-driven of underwater sensor networks.

Keywords Software-driven · Software modem · Acoustic · Sound · Underwater · Off-the-shelf · Communication · Sensor networks

R. Jurdak's work is supported by IRCSET Embark Grant.
A.G. Ruzzelli and G.M.P. O'Hare's work is supported by Science Foundation Ireland under Grant No. 03z/IN.3/1361.
The work of C.V. Lopes was partially supported by US National Science Foundation Grant No. CCF-0347902.

R. Jurdak (✉) · A.G. Ruzzelli · G.M.P. O'Hare
School of Computer Science and Informatics, University College Dublin, Belfield 4, Dublin, Ireland
e-mail: rjurdak@ieee.org

A.G. Ruzzelli
e-mail: ruzzelli@ucd.ie

G.M.P. O'Hare
e-mail: gregory.ohare@ucd.ie

C.V. Lopes
University of California, Irvine, CA 92697, USA
e-mail: lopes@uci.edu

1 Introduction

Acoustic underwater communications is a well-established field that has been used by the military for almost half a century. Recently, civilian applications for underwater communications have emerged, including oil prospecting and water quality monitoring.

The design of underwater communication systems has so far relied on expensive specialized hardware for acoustic communication and modulation. The conventional reliance on hardware acoustic modulation has stemmed from low processing speeds that did not allow the modulation of acoustic signals in software.

Software modulation and demodulation [9, 13] is a recent alternative approach which overcomes most of the drawbacks of hardware modems. Recent advances in miniaturization and circuit integration have yielded smaller and more powerful processors that are capable of efficiently running acoustic modulation and demodulation software. Software modulation also provides a higher level of flexibility for on-the-fly tuning of modulation parameters, such as the data transfer rate and symbol duration, to suit the variable conditions of a particular deployment environment.

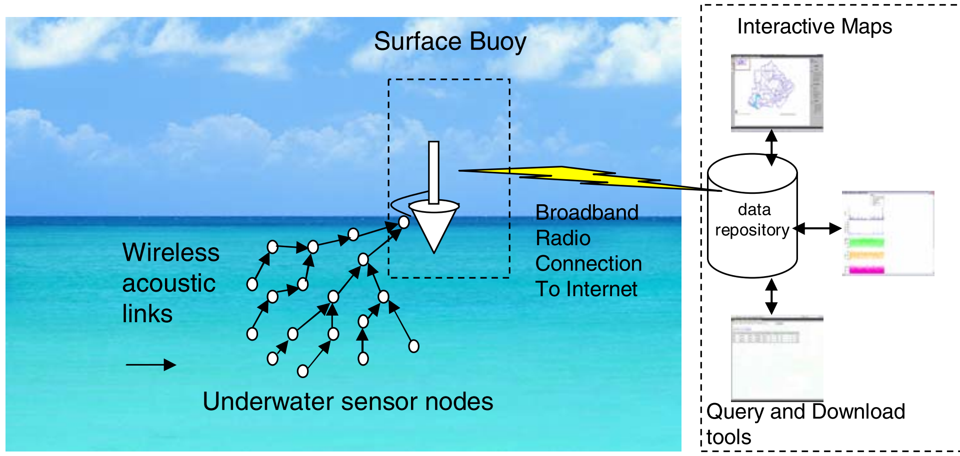


Fig. 1 Target network application

Coupling software modems with generic microphones and speakers that are built-in to off-the-shelf sensor modules eliminates the need for specialized communication hardware and reduces system cost, facilitating the dense deployment of sensor nodes to form underwater acoustic sensor networks. The use of low power generic acoustic hardware and software modems for underwater communication yields a low bit rate in the order of tens to hundreds of bits per second. The achievable bit rate of the proposed system is sufficient for monitoring sensor networks, such as for environmental or habitat monitoring, where the nodes sample their sensors and send the data once during each update period, typically in the order of minutes. Since each node must send only a handful of sensor values during each update period, a data transfer rate in the order of tens of bits per second provides more than enough throughput to communicate all the sensor values during an update period.

The target network application consists of general purpose sensor modules that use software modems and generic hardware to communicate acoustically in shallow water and send the data to the base station, as shown in Fig. 1. We expect the network deployment to consist of tens to hundreds of sensor modules in a shallow water environment. The sensor modules can communicate acoustically through wireless multi-hop links. The modules periodically sample their sensors, collecting physical indicator data such as temperature and salinity, which influence pollution levels in the water. After sampling their sensors, the nodes report their data to a surface node nearby. The surface node, known as the base station, includes a water-immersed acoustic transceiver that communicates with the underwater nodes. It is also equipped with a long range wireless broadband communication card that uses a cellular or satellite connection to stream the network data towards a central server.

The server includes a data repository that archives historical data from the monitored area. The proposed network deployment will stream near-real time data from the aquatic environment into the data repository, providing professionals in the water management and research communities with access to sizeable and timely data from the water. The application will also leverage existing environmental information management platforms [2] for providing the tools necessary for analyzing, displaying, and sharing the collected data.

The work described in this paper outlines the design considerations for the development of software-driven underwater sensor networks for monitoring pollution indicators in rivers, lakes, estuaries, and coastal areas and subsequently providing the data to environmental engineers in near real-time. While recent surveys cover underwater acoustic networks [1] and practical issues in these networks [14], the focus here is on design and deployment issues of software-driven underwater sensor networks in shallow water. In particular, this paper leverages our recent field experiments to identify suitable design choices for the acoustic communication link including signal modulation, symbol synchronization, filtering, and demodulation. The paper also visits medium access control (MAC) and routing layer issues and proposes cross-layer design to cope with limited bandwidth and energy resources of underwater sensor networks. Our field experiments have also revealed several logistical issues for underwater sensor network deployments, including waterproofing, casing, calibration, and fouling.

2 Related work

This section first reviews previous work on hardware acoustic modulation. The second part of the section focuses on software modulation as an enabler of wireless underwater communication.

2.1 Hardware modems

Earlier efforts in acoustic communication have focused on using specialized and dedicated hardware for underwater acoustic modulation and demodulation. Acoustic underwater communication is a mature field and there are several commercially available underwater acoustic modems [3, 12]. The commercially available acoustic modems provide data rates ranging from 100 bps to about 40 Kbps, and they have an operating range of up to a few km and an operating depth in the order of thousands of meters. The cost of a single commercial underwater acoustic modem is at least a few thousand US dollars. The prohibitive cost of commercial underwater modems has been an obstacle to the wide deployment of dense underwater networks, until the recent development of research versions of hardware acoustic modems.

Researchers at the Woods Hole Oceanographic Institution are developing a Utility Acoustic Modem (UAM) as a completely self-contained, autonomous acoustic modem capable of moderate communication rates with low power consumption [19]. This modem uses a single specialized DSP board with on board memory and batteries. The purpose of developing the UAM is to make a more affordable acoustic modem available for the research community. Researchers at UC, Santa Barbara are also developing a hardware acoustic underwater telemetry modem [7] for ecological research applications, using a DSP board with custom amplifiers, matching networks, and transducers. Their modem is intended for interfacing to nodes in an underwater ad hoc network, and it achieves a 133 bps data rate. Whereas both of the above efforts [7, 19] aim at making underwater acoustic modems cheaper and more accessible by developing specialized affordable hardware, our work aims at driving the cost even lower and at making acoustic underwater communications even more pervasive through the development of software acoustic modems that can operate on generic hardware platforms.

In a more recent article, Wills et al. [21] propose their design for an inexpensive hardware modem for dense short-range underwater sensor networks. Their work aims at borrowing communication concepts, such as wake-up radio, from terrestrial sensor networks. Although we share the same end goal as Wills et al. (inexpensive acoustic modems for dense short-range wireless networks), our approach differs in its emphasis on modulation through software rather than through specialized hardware.

One of the few attempts to deal with generic microphones and speakers is Vasilescu et al. [20]. These authors propose a network that combines acoustic and optical communications, stationary nodes and AUV's for monitoring coral reefs and fisheries with ranges in the order of hundreds of meters. Their work uses generic microphones and speakers along with a specialized integrated circuit that generates ASK or FSK modulated sound signal in order to demonstrate the acoustic communication capability underwater. Vasilescu et al. achieve a bit rate in the order of tens of bits per second up to about 10 to 15 meters. Although our work resembles their work in the use of generic microphones and speakers for acoustic communications, it differs in its proposal and implementation of software modems for off-the-shelf mote platforms rather than the use of specialized integrated circuits for communication.

2.2 Software modems

With the rapid increase in processor speeds, the idea of implementing acoustic modems in software becomes feasible and even attractive due to the low cost processing power. Coupling software acoustic modems with the use of microphones and speakers for transmission and reception can eliminate the need for specialized hardware for acoustic communication, trading off cheap computational power for expensive communication hardware. The cost of software acoustic modems is limited to the development cost, after which the per unit cost is zero.

Because of these attractive features, Lopes and Aguiar [13] have investigated using software modems for aerial acoustic communications in ubiquitous computing applications. Building on their work, software acoustic modems can also eliminate the need for specialized hardware in underwater acoustic communications, thereby encouraging wider deployment of underwater sensor networks. In preliminary experiments, we profiled the underwater acoustic spectrum and data communications capabilities with software acoustic modems [9]. Our work used waterproofed generic microphones and speakers, connected to laptops on the surface, for sending and receiving software modulated acoustic signals. The achieved bit rates were in the order of tens of bits per second for distances up to 10 meters. More recently, we extended the work in [9] by coupling software modems with Tmote Invent modules. Our recent study [10] presents underwater experiments results that confirm the communication capability of software modems on Tmote Invent hardware [16]. This chapter further investigates the application opportunities and the technical and logistical challenges of software-driven underwater sensor networks with Tmote Invent modules.

3 Fundamentals of underwater acoustics

This section reviews the main concepts of underwater acoustic communications and how they relate to software-driven underwater networks. The passive sonar equation [18] characterizes the signal to noise ratio (SNR_u) of an emitted underwater signal at the receiver:

$$SNR_u = SL - TL_u - NL_u + DI \tag{1}$$

where SL is the source level, TL_u is the underwater transmission loss, NL_u is the noise level, and DI is the directivity index.

The directivity index DI for our network is zero because we assume omnidirectional hydrophones. Note that this is a conservative assumption, since the use of a directive hydrophone [5] can reduce power consumption and increase the data rate or transmission range of software-driven acoustic communications.

3.1 Source level

Typically, the specifications of audio speakers indicate the speaker's maximum emitted signal power. The transmitter source level (SL) of underwater sound relates to signal intensity I_t, which in turn depends on the transmission power. Given the transmission power P_t, the transmitted intensity of an underwater signal at 1 m from the source can be obtained through the following expression [18]:

$$I_t = \frac{P_t}{2\pi \times 1\ \mathrm{m} \times H} \tag{2}$$

in Watts/m^2, where H is the water depth in m. The following equation determines the source level SL in dB:

$$SL = 10 \log\left(\frac{I_t}{10^{-12}}\right) \tag{3}$$

In order to determine the SL of our generic speakers, we simply use the output power value P_t of the speakers in (2), which yields the value of I_t. We then compute the SL through (3).

3.2 Transmission loss

The transmitted signal pattern has been modelled in various ways, ranging from a cylindrical pattern to a spherical one. The following expression governs acoustic signals propagation in shallow water [18]:

$$TL_u = 10 \times \mu \log d + \alpha d \times 10^{-3} \tag{4}$$

where d is the distance between source and receiver in meters, α is the frequency dependent medium absorption coefficient in dB/km, and TL is in dB. The variable μ depends on the signal spreading pattern. In shallow water cases that are semi-constrained by the floor of the water body, the value of μ lies somewhere between 1 and 2, depending on the depth.

Equation (4) indicates that the transmitted acoustic signal loses energy as it travels through the underwater medium, mainly due to distance dependent attenuation and frequency dependent medium absorbtion. Fisher and Simmons [4] conducted measurements of medium absorbtion in shallow seawater at temperatures of 4°C and 20°C. We derive the average of the two measurements in (5), which expresses the average medium absorption at temperatures between 4°C and 20°C for our frequency range of interest:

$$\alpha = \begin{cases} 0.0601 \times f^{0.8552} & 1 \le f \le 6 \\ 9.7888 \times f^{1.7885} \times 10^{-3} & 7 \le f \le 20 \end{cases} \tag{5}$$

where f is in KHz, and α is in dB/km.

Through (5), we can compute medium absorbtion for any frequency range of interest. We use this value for determining the transmission loss at various internode distances through (4) which enables us to compute the source level in (3) and subsequently to compute the power needed at the transmitter.

3.3 Noise level

Factors contributing to the noise level NL_u in shallow water networks include waves, shipping traffic, wind level, biological noise, seaquakes, volcanic activity, and rain, and the impact of each of these factors on NL_u depends on the particular setting. For instance, shipping activity may dominate noise figures in bays or ports, while water currents are the primary noise source in rivers. In a swimming pool environment, where we conducted our experiments, the main sources of underwater noise are swimmers, vibrations from people walking near the pool, water pumps, and drains.

4 Technical considerations

Building on the acoustic fundamentals of Sect. 3, we proceed to the design of our software-driven wireless acoustic communication system.

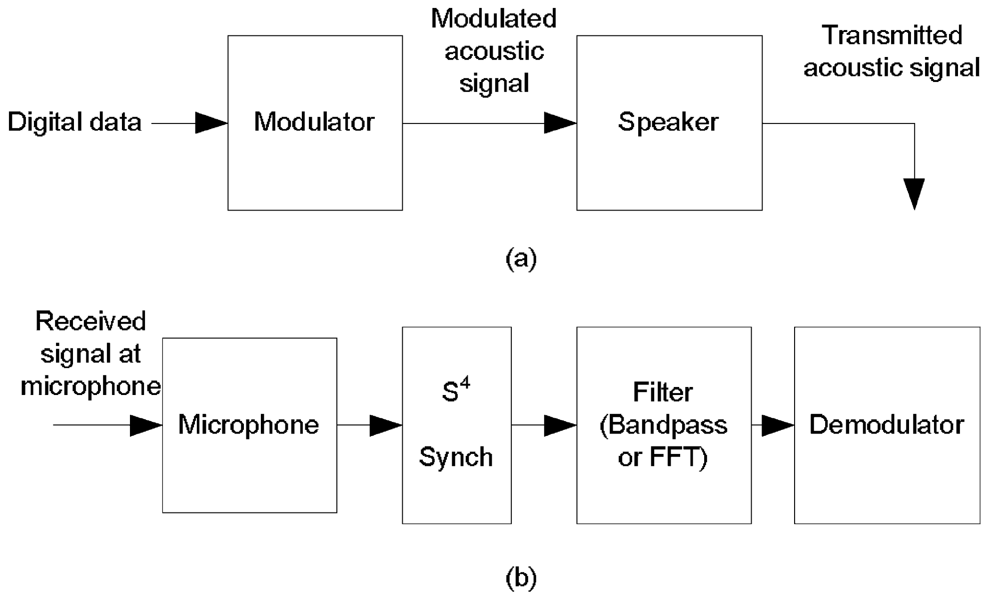
4.1 Physical layer

The physical layer design for our system involves both communication and modulation. The communication hardware is simply the microphone and speaker built-in to sensor nodes, serving as receiver and transmitter respectively. The focus then, is on signal modulation, which takes place through software resident on the motes. Figure 2 shows the block diagram for the communication system. The following discussion explores the main components of Fig. 2 separately.

4.1.1 Modulation

The first component of the acoustic communication system is software modulation, that takes digital data as input and modulates an acoustic signal with the data. The potential choices of modulation schemes for software modems

Fig. 2 Block diagram for software modem (**a**) Modulator/Transmitter (**b**) Demodulator/Receiver



include amplitude shift keying (ASK), phase shift keying (PSK), and frequency shift keying (FSK). We have selected FSK as the lowest complexity mechanism to run on the resource-limited mote platforms.

In general, FSK uses 2^N frequencies to encode N bits per frequency. The signal demodulation at the receiver can use low complexity techniques, such as the Fast Fourier Transform (FFT), to determine the frequency content of the signal. Choosing a number of frequencies that have high signal-to-noise ratio (SNR) for the channel and ensuring sufficient spectral separation between the frequencies for FSK provides robust underwater communication for low power transmissions with minimal processing complexity at the receiver.

We recently performed an empirical study to investigate the spectral properties of the underwater channel in a controlled water environment [10]. The study used the Tmote Invent module speaker as the transmitter of acoustic signals and a generic PC microphone as the receiver. The components were waterproofed using off-the-shelf elastic latex membranes that vibrate sufficiently to preserve most the acoustic properties of the speaker and microphone. The study evaluated the signal-to-noise ration (SNR) of frequency tones between 400 and 6700 Hz at 100 Hz increments. The selection of the 100 Hz band separation between frequencies provides for low complexity frequency detection at the receiver. Note that narrower separation bands enable the use of more frequencies within the same available bandwidth, which increases bit rates but also reduces signal quality at the receiver. In general, there is a tradeoff in digital modulation techniques between the number of signal levels (in phase, amplitude, or frequency) and the quality of the signal. In our case, increasing the number of frequencies by using narrower frequency bands requires the use of higher order filters or fast Fourier transform (FFT) at the mote to decode the signal. This is not feasible with the limited processing power of the motes.

The study revealed that the channel, which includes the speaker, latex membranes, the water, and the microphone, exhibits the highest SNR at frequencies in the range of the 1000 Hz to 2500 Hz. The SNR drops steadily at frequencies above 3 KHz. The results of these empirical experiments have enabled us to identify the frequencies with the highest SNR for our system are: 1000, 1200, 1300, 1500, 16000, 1700, 1800, and 2100 Hz. Consequently, these are the frequencies that we have selected for our 8-frequency FSK software modulation scheme.

The testing of the modem in the water investigated 4 different bit rates from 6 bps to 48 bps, by varying the symbol duration from 500 ms to 62.5 ms. The experiment results revealed a low bit error rate for all data transfer rates within a transmission distance of 17 m, which was the size of the testing area.

The output of the modulator block is a modulated acoustic signal that is then transmitted over the wireless medium through the built-in speaker.

4.1.2 Symbol synchronization

The transmitted signal is received by the microphone at another node, on which the software then proceeds in the decoding process. Essential to acoustic communication with software modems is the ability of the receiver to synchronize to the first symbol of an incoming data stream. Traditional symbol synchronization approaches rely on the transmission of a predefined sequence of symbols, often referred to as a training sequence. The conventional approach makes two assumptions about the communication channel that do not

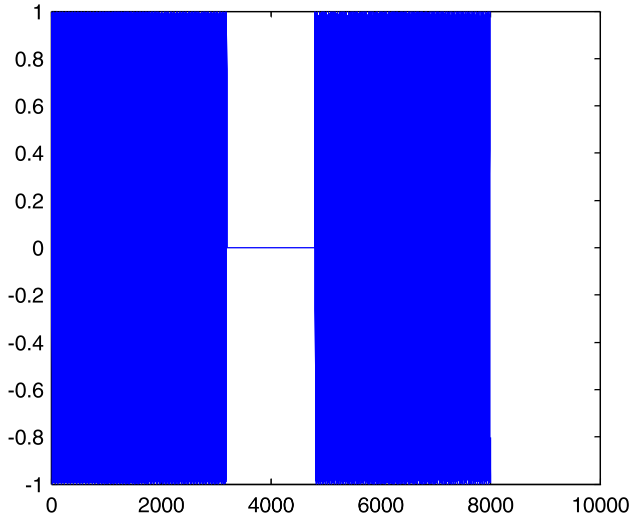


Fig. 3 Synchronization symbol structure as produced at the PC

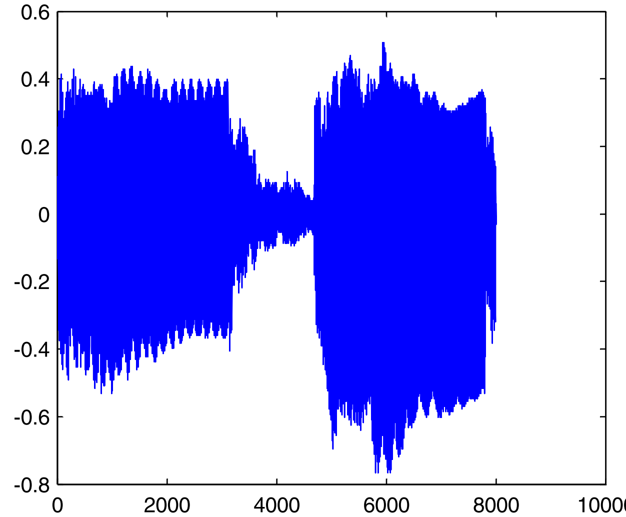


Fig. 4 First reference synchronization symbol captured from Tmote Invent speaker

hold for software-driven acoustic communications, namely: (1) a bit rate at least in the order of tens of kilobits per second; (2) a bit error rate (BER) in the order of 10^{-6} or lower. Software-driven acoustic communication, both aerial and underwater, supports lower bit rates that range between tens to hundreds of bits per second [9, 10]. In addition, the bit error rate of software-driven acoustic communication is several orders of magnitude higher than the radio frequency bit error rate. The higher bit error rate in acoustic communications tends to cause loss of training sequence symbols, preventing proper symbol synchronization. Furthermore, providing high redundancy in the training sequence to mitigate training symbol losses is not an option for the narrow usable bandwidth of software-driven acoustic communications.

We recently proposed and tested the S^4 synchronization mechanism [11], that uses Short Signature Synchronization Symbols (S^4) to align symbol boundaries at the receiver. The design of the signature symbol aims at a high probability of correlation at the receiver even in cases of partial loss of the symbol and at low probability of false synchronization with ambient noise or data symbols.

The proposed synchronization technique considers a signal composed of a preamble, data symbols, and a post-amble. The preamble and post-amble have the same structure, combining two square waves of frequencies f_1 and f_2, and separated by a silence of duration T. Figure 3 shows the synchronization symbol structure, as generated at a PC. Figure 4 shows the structure of the synchronization symbol generated by the Tmote Invent speaker, which is not as symmetric as the same symbol generated by the PC, due to the hardware imperfections of the mote speaker. Furthermore, the Tmote Invent speaker generates non-deterministic variations in signals captured in different media (e.g. air, water, etc.), as discussed in Sect. 5.2. These properties of the Tmote Invent speakers have dictated the following signal design choices:

1. The signature of the preamble and post-amble symbol cannot rely solely on amplitude. As a result, the signal signature includes the two square waves of predetermined frequencies, the duration of the square wave signals, and a predetermined guard time between them. This signal shape allows the correlation function at the receiver to detect the surge at the beginning of the first square signal, the drop at the end of the first square signal, the surge at the beginning of the second square signal, and the drop at the end of the second the square signal. In addition, the synchronization technique selects a guard time duration that is not a constant multiple of the inter-symbol guard times to avoid any correlation or misalignments with data symbol sequences.
2. The non-deterministic variation in signal amplitude within different media motivates matching the incoming signal to more than one reference signal at the receiver. The receiver can then correlate each of the N reference signals with the incoming signal, select the reference signal with the best correlation to the received signal, and synchronize to the sample with the highest correlation between the selected reference signal and the received signal. Each of N reference signals are captured from within the target deployment area. For example, prior to deploying the nodes in a river, a calibration phase is performed during which N reference signals are transmitted in the river and stored at the receiver, providing better correlation.

We illustrate the receiver synchronization process through an example with $N = 2$. Upon receiving an incoming signal, the algorithm correlates the received signal with the two

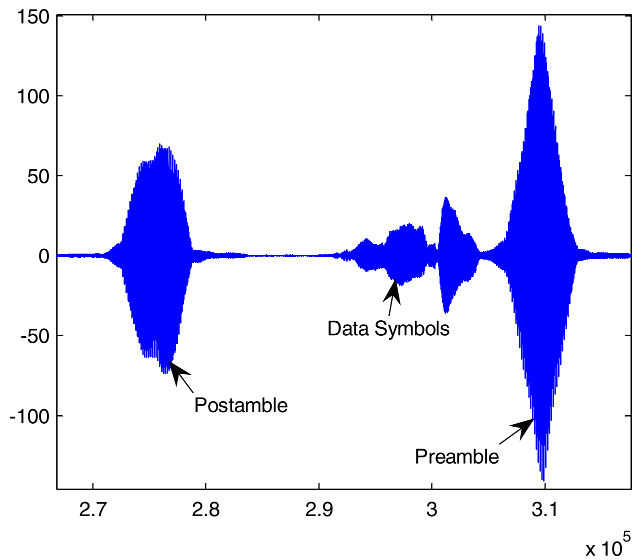


Fig. 5 Correlation output with the first reference signal

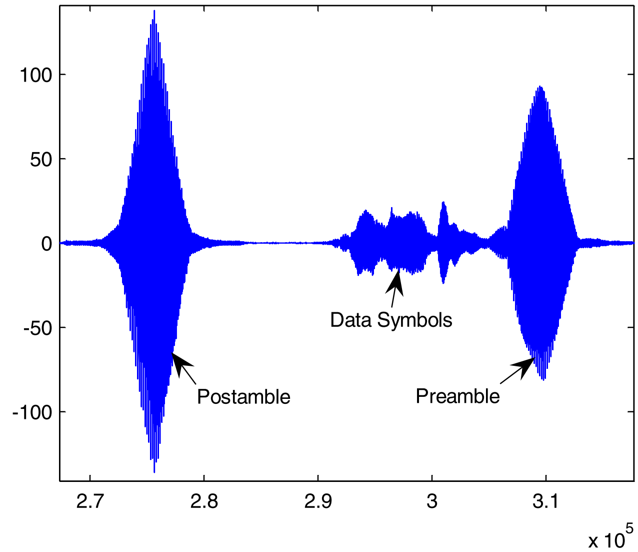


Fig. 6 Correlation output with the second reference signal

reference signals. In our current example, Figs. 5 and 6 represent the correlation outputs of the received signal with the two reference signals, yielding 4 correlation peaks: 2 peaks at the preamble, and 2 peaks post-amble. Visual inspection of the 4 peaks in these figures reveals that the preamble in the first correlation output and the post-amble in the second correlation output have the sharper peaks, but it does not indicate which of the 2 sharper peaks is a better reference for synchronization. The synchronization algorithm stores the amplitude of each peak a_i and the sample corresponding to each peak t_i, where i is the peak index. The algorithm then checks each peak's properties in order to determine the best peak to consider. First, the synchronization algorithm isolates each peak by storing $P/2$ signal samples around each peak into a vector S_i, where P is number of samples in each square signal of the preamble and post-amble. Next, the algorithm computes the envelope of the signal around each peak through the following expression:

$$\begin{aligned} Q_i &= Hilbert(P_i) \\ R_i &= Q_i * conj(Q_i) \end{aligned} \tag{6}$$

which first takes the Hilbert transform of each peak and then retains the real part of the transform. The slope of the envelope around each peak is then computed through the following expressions:

$$\begin{aligned} R_i^1(i) &= dR_i/dt \\ LR_i &= average[R_i^1(t_i - \alpha)] \\ LF_i &= average[R_i^1(\alpha - t_i)] \end{aligned} \tag{7}$$

where LR_i represents the average slope of the envelope prior to peak i, LF_i represents the average slope of the envelope after peak i, and α is a constant that specifies the number of samples to check around the peak. The receiver then estimates the sharpness of the peak PS_i through the following equation:

$$PS_i = \frac{a_i}{average(S_i)} \tag{8}$$

which is essentially the ratio of the peak amplitude to the average amplitude of the samples in S_i. The variable PS_i is proportional to the peak sharpness, since the higher the value of PS_i, the sharper the peak i. Next, the receiver computes the symmetry of each peak through the following equation:

$$SYM_i = \frac{LR_i + LF_i}{LR_i} \tag{9}$$

The SYM_i variable is inversely proportional to the symmetry of a peak i. In the extreme case, LR_i and $-LF_i$ are equal and SYM_i is zero. An ideal peak has a high degree of sharpness and a high degree of symmetry, so the peak quality can be quantified by the following expression:

$$PQ_i = \frac{PS_i}{SYM_i} \tag{10}$$

After determining each PQ_i, the algorithm selects the peak with the highest peak quality and uses the peak sample t_i to synchronize to the received signal. In case t_i corresponds to the post-amble, the algorithm subtracts the fixed frame length from t_i to compute the start of the data symbols.

In short, upon receiving an acoustic signal through its microphone, the receiver executes the following steps in the S^4 algorithm:

Correlate signal to each of the 2 reference signals
ForEach: Peak i
  Record t_i and a_i
  Store S_i
  Determine envelope
  Compute R_i^1
  Compute LR_i and LF_i
  Compute PS_i, SYM_i, and PQ_i
Select Peak m with highest PQ_i
Synchronize to Peak m

The output of the S^4 provides the index of the first data sample in the received signal, which is used by the filtering block.

4.1.3 Filtering

The receiver then proceeds to filter the received signal. Filtering the acoustic signal at the receiver minimizes the effect of out-of-band noise on the decoding process. A suitable choice of filtering method depends highly on the processing capability of at the receiver. Our system currently supports two filtering methods, with the first method employing narrow bandpass filters at the relevant frequencies. The second filtering method applies an FFT to the received signal and examines the signal amplitude at the FFT samples corresponding to the relevant frequencies.

The narrowband filtering method provides a finer signal quality as it only focuses on the frequencies of interest and excludes all interference outside this spectrum. The superior performance of the narrowband filtering method comes at the cost of higher processing activity at the receiver. In fact, the narrowband filtering method is suitable for running on a PC or a PDA that acts as a base station for the underwater network, as in Fig. 1.

The FFT method can run on the motes themselves as it has lower processing complexity. Running the FFT method on motes enables the deployment of a multi-hop network where nodes can autonomously decode, process, and relay received signals. The FFT method provides a coarser signal quality than filtering since it does not exclude interference from outside the frequency spectrum of interest.

4.1.4 Demodulation

The filtered signal then proceeds to the demodulator component. The demodulator begins examining the signal at the first data sample, which has been determined by the S^4 block. Starting at the first data sample and taking the number of samples that corresponds to one symbol, the demodulator determines the frequency component with the highest amplitude within this window, and outputs the data symbol corresponding to the highest frequency. For subsequent bits, the demodulator shifts the start sample of the previous symbol by the symbol length, and repeats the process of determining the strongest frequency component.

4.2 Communication protocols

The design of the higher layer communication protocols for this system should adopt a minimalist approach to avoid creating high overhead in this bandwidth and energy limited system. The MAC protocol design for underwater sensor networks cannot exploit traditional low duty cycle protocols for terrestrial sensor networks. The reason is that the energy cost of transmission is generally much higher than the cost of signal reception in underwater networks, whereas the cost of transmission and reception is almost the same in terrestrial sensor networks [6]. Thus, signal transmissions dominate the energy consumption profile of underwater nodes. As such, the MAC protocol design should minimize control overhead messages, such as request to send (RTS) and clear to send (CTS), rather than implementing sleep policies. One possibility for our system is to use burst tones at the beginning of transmission to reserve the channel. In particular, the S^4 preamble can serve as a burst tone for reserving the channel at the MAC layer.

For routing data towards the surface node, our communication system advocates the use of short multi-hop links for deploying dense underwater sensor networks. Keeping in line with the system's minimalist approach, the network should make use of simple multi-hop routing techniques, such as the directed broadcast with overhearing method of MERLIN [15]. The relatively low cost of receiving signals actually favors the use of overhearing. Our recent study on the upper bounds for transmissions in directed broadcast networks with overhearing has shown that this method has only few redundant packet, resulting in low overhead for large networks.

The proposed communication protocol design concepts exploit cross-layer interactions [8] to optimize the use of bandwidth and energy resources for this system. We expect that further cross-layer optimization opportunities will arise as the project advances.

5 Logistical considerations

5.1 Waterproofing and casing

The most common waterproofing method for underwater communications hardware is to place the hardware in a custom-designed waterproof case with a special air-locked hole for the hydrophone and transducer that are in contact with the water. The cost associated with the material and design of the custom-designed casing strategies significantly

Table 1 Comparison of the percentage of symbols correctly received for the PC speakers and Tmote Invent experiments

Transfer rate (bps)	6	12	24	48	96
Latex mem. up to 17 m	≥95%	≥90%	≥81%	≥79%	N/A
Vinyl mem. up to 10 m	N/A	≥90%	≥78%	≥35%	≥10%

increases system cost and discourages large scale deployments of underwater sensor networks.

Our project's design strategy advocates the use off-the-shelf components not just for communication and sensing, but also for the protection and waterproofing of the components. As such, our design places the sensor nodes in elastic latex membranes that are cheap and readily available on the market. The latex membranes take the form of a balloon that is sealed to waterproof the sensor nodes. Since the sensor module includes the speaker and microphone, these acoustic communications components are also fully enclosed within the latex membranes. The elasticity of the membranes ensures that the acoustic waves transmitted by the speaker are transferred to the water through the elastic membrane. At the receiver side, the membranes vibrate upon the reception of an acoustic signal, transferring the signal to the interior of the membrane where the microphone can detect it.

The use of the latex membranes causes relatively small reductions in signal amplitude. We recently compared the suitability of two membranes for waterproofing the sensor nodes: (1) a vinyl membrane; and (2) a latex membrane. Table 1 illustrates the percentage of correctly received symbols for bit rates ranging between 6 bps and 96 bps for both membranes. The vinyl membrane experiments were conducted with a generic PC speaker as a transmitter, while the latex membrane experiments were conducted with Tmote Invent speakers. The output power rating of the two speaker types is the same, providing solid ground for comparing the results of the two experiment sets. The results in Table 1 show that the latex membrane has a better coupling with the water, yielding a notably lower bit error rate at the higher transmission rate of 48 bps.

5.2 Calibration

Our field tests in different bodies of water have revealed a distinct background noise and interference pattern in each case. For instance, the primary noise source in swimming pools is water pumps, whereas the noise sources in a river include currents and wildlife activity. The dependence of the noise profile on the deployment environment requires calibration steps, which could be manual or automatic, prior to placing the sensors in the water. Fortunately, the implementation of modulation and communication in software provides maximum flexibility for on-the-fly calibration.

A central issue for calibration is the frequency-selective noise in the deployment environment. The choice of frequencies for the S^4 synchronization symbol must avoid frequencies with high noise in a particular deployment environment. The selection of the S^4 frequencies is critical for proper system operation, as choosing unsuitable frequencies causes large synchronization errors, resulting in many bit misalignments. Similarly, it is also important to choose data symbol frequencies that avoid the high noise frequencies.

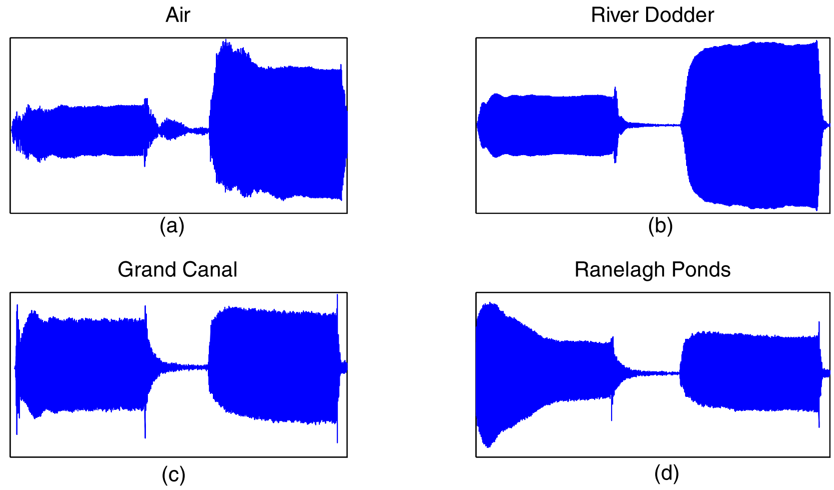
The selection of the suitable frequencies for S^4 synchronization and the data symbols can be done automatically. During an initial setup phase, one node, typically the base station, can be designated as a calibration receiver, and another as a calibration transmitter. Upon deployment, the designated transmitter sends an apriori known calibration signal that includes a diverse set of S^4 symbols with different frequency combinations, followed by a sequence of frequency tones that covers all the possible symbol frequency tones. The designated receiver processes the calibration signal by comparing the processed signal against a locally stored version of the reference signal. The receiver then selects the frequencies that have been received with the highest SNR, and transmits a short broadcast message indicating these frequencies to the other nodes.

Another calibration issue is ensuring that the stored S^4 symbol at each node is representative of the symbol as it is received in the current deployment environment. Our field experiments have shown that the structure of a received S^4 symbol with the Tmote Invent microphone in air is different than the structure in water. As Fig. 7 shows, the structure even varies across different water media, depending on the current, depth, and suspended solids in the water. For instance, the envelope of the S^4 symbol differs significantly when the signal is received in the river (Fig. 7b) and in the pond (Fig. 7d). The plots in Fig. 7 illustrate that the relative amplitude of each of the square signals changes depending on the medium. Other signal artifacts, such as impulses at the beginning or end of the signal, are also dependent on the deployment environment. As such, each node should store at least one instance of the S^4 symbol *as it is received in the current deployment environment*. This maximizes the probability of successful correlation and synchronization through the S^4 mechanism.

5.3 Fouling

Fouling is a process by which marine wildlife, such as barnacles, zebra mussels, weeds, and algae attach themselves to still object in the water. For this project, avoiding fouling effects is especially important since the attachment of

Fig. 7 Shape of S^4 symbol as received in (**a**) Indoor aerial channel; (**b**) River Dodder; (**c**) Grand Canal in Dublin; (**d**) Ranelagh pond



organisms to the membrane could limit or change the membrane's vibration characteristics. Traditional anti-fouling approaches include the use of special copper-based paints to prevent the attachment of organisms to boat bottoms. Current research focuses on developing alternatives [17] to paint-based solutions, which are harmful to the ambient environment.

Our project aims at protecting the environment, and not damaging it in the process, so we intend to adopt one of the emerging anti-fouling techniques. One interim solution under consideration is to place the nodes in the latex membranes and then to fix the membranes inside a resilient cubic plastic box whose purpose is to shield the nodes from fouling and other hazards in harsh underwater environments. The surfaces of the plastic box would be perforated to maintain acoustic coupling between the box contents and the water.

6 Conclusion

This paper has presented the lessons learned so far from our field experiments with software-driven underwater sensor networks. The main technical challenges include the design and integration of the modulation, synchronization, filtering, and demodulation techniques for the software modems. The design of higher layer communication protocols for this system should also adopt a minimalist approach to minimize control overhead. Logistical considerations for underwater networks depend on the deployment environment. We have identified common logistical considerations low-power underwater networks, which include the need for resilient waterproofing and casing, calibration, and anti-fouling measures. Our initial experiments in various bodies of water have exposed the benefits of using software modems for underwater communications capable of functioning on off-the-shelf multi-purpose sensors. The design guidelines in this paper lay the groundwork for further development of software-driven of underwater sensor networks.

References

1. Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, *3*(3), 257–279.
2. California Sustainable Watershed/Wetland Information Manager available at: http://www.calswim.org.
3. DSPCOMM available at: www.dspcomm.com.
4. Fisher, F. H., & Simmons, V. P. (1977). Sound absorption in sea water. *Journal of Acoustical Society of America*, *62*, 558.
5. Fruehauf, N., & Rice, J. A. (2000). System design aspects of a steerable directional acoustic communications transducer for autonomous undersea systems. In *OCEANS* (Vol. 1, pp. 565–573). New York: IEEE.
6. Harris, P., Stojanovic, M., & Zorzi, M. (2006). When underwater acoustic nodes should sleep with one eye open: idle-time power management in underwater sensor networks. In *Proc. WUWNet*.
7. Iltis, R. A., Lee, H., et al. (2005). An underwater acoustic telemetry modem for eco-sensing. In *Proc. Oceans'05*, September, 2005.
8. Jurdak, R. (2007). *Wireless ad hoc and sensor networks: a cross-layer design perspective*. New York: Springer.
9. Jurdak, R., Lopes, C. V., & Baldi, P. (2006). Software acoustic modems for short range mote-based underwater sensor networks. In *Proc. of IEEE Oceans*, Singapore, May, 2006.
10. Jurdak, R., Aguiar, P. M. Q., Baldi, P., & Lopes, C. V. (2007). Software modems for underwater sensor networks. In *Proc. of Oceans'07*, June, 2007.
11. Jurdak, R., Ruzzelli, A. G., O'Hare, G. M. P., & Lopes, C. V. (2007). Reliable symbol synchronization in software-driven acoustic networks. In *Proc. of IEEE GlobeCom*, Washington DC, November, 2007.
12. Linkquest Inc available at: www.link-quest.com.
13. Lopes, C. V., & Aguiar, P. (2003). Acoustic modems for ubiquitous computing. *IEEE Pervasive Computing*.
14. Partan, J., Kurose, J., & Levine, B. N. (2006). A survey of practical issues in underwater networks. In *Proc. WUWNet*.

15. Ruzzelli, A. G., O'Hare, G. M. P., & Jurdak, R. (2008). Merlin: cross-layer integration mac and routing for low duty-cycle sensor networks. *Ad Hoc Networks*. Special issue on energy efficient design in wireless ad hoc and sensor networks, February, 2008.
16. Sentilla Corporation available at: www.sentilla.com.
17. The Maya2 Project, http://www.maya-net.org/.
18. Urick, R. J. (1983). *Principles of underwater sound*. New York: Mcgraw-Hill.
19. Utility Acoustic Modem available at: auvlab.mit.edu.
20. Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., & Corke, P. (2005). Data collection, storage, and retrieval with an underwater sensor network. In *Proc. Sensys'05*, San Diego, CA.
21. Wills, J., Ye, W., & Heidemann, J. (2006). Low power acoustic modem for dense underwater sensor networks. In *Proc. IEEE (WUWNet'06)*.

Raja Jurdak is currently a research fellow at University College Dublin. He received the BE degree in Computer and Communications Engineering from the American University of Beirut in 2000, the MS degree in Computer Networks and Distributed Computing from University of California Irvine in 2001, and the Ph.D. degree in information and computer sciences at the University of California, Irvine in 2005. From 2005 to 2006, he was a postdoctoral researcher at the University of California, Irvine. He received the Irish Research Council on Science Engineering and Technology Embark fellowship in 2006. Dr. Jurdak's research interests include sensor network communication protocol design and application development. His current and past research projects include acoustic positioning systems, underwater sensor networks, patient monitoring, pet tracking, and fire prevention in granaries. Much of his work has incorporated cross-layer protocol design, which led him to author a book that was published by Springer in 2007 titled "Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective". Dr. Jurdak has also authored or co-authored over 35 peer-reviewed journal and conference articles, and 1 pending patent. He is a member of IEEE Computer Society and IEEE Communication Society.

Antonio G. Ruzzelli is currently in the final stage of Ph.D. writing at University College Dublin (UCD), School of Computer Science and Informatics. For the last 2 summers he has been a visiting scientist of Philips research in The Netherlands. At UCD, he is involved in the Adaptive Information Cluster, AIC, which addresses the area of sensor networks and ambient intelligence. In particular, he is responsible for the implementation of MAC and routing protocols for energy-efficient and adaptive communication of sensor nodes. At Philips, his work focuses on 802.15.4 optimizations and networking capabilities 6LowPan complaint for sensor based-medical systems. He authored a patent on load balancing in sensor networks and about 17 papers on the area of wireless sensor networks in peer reviewed conferences and journals. He is a delegate of the Management Committee European COST Action 2100 for the Republic of Ireland, 2007–2011.

Gregory M.P. O'Hare is a Senior Lecturer in the School of Computer Science and Informatics. Previously, he was a member of faculty at the University of Manchester Institute of Science and Technology (UMIST), UK. He is director of the PRISM (Practice and Research in Intelligent Systems & Media) Laboratory within the School. His research focuses upon Multi-Agent Systems (MAS) and Mobile and Ubiquitous computing. He has published over 185 journal and conference papers and 2 books.

Cristina Videira Lopes is an Associate Professor at the University of California, Irvine. She conducts research in Programming Languages, Software Engineering, and Pervasive Computing. She is the recipient of an NSF CAREER Award.