A Low-cost and Flexible Underwater Platform to Promote Experiments in UWSN Research

Niaz Ahmed, Waqas bin Abbas, Affan A. Syed
National University of Computer and Emerging Sciences
Islamabad, Pakistan
{niaz.ahmed, waqas.abbas, affan.syed}@nu.edu.pk

ABSTRACT

Underwater acoustic sensor networks (UWSN) is a relatively new research area, and remains quite challenging due to limited bandwidth, low data rate, severe multipath, and high variability in the channel conditions. These complicated and non-linear channel characteristics render incorrect most simplifying assumptions used in simulations. We believe that, while researchers have proposed several novel protocols, their use of models and simulations as the *only* form of validation and intra-protocol comparison remains removed from reality. We argue that research experimentation is hindered by two fundamental constraints: high cost of underwater networking experiments, and lack of a single, easily-replicable platform for evaluation. We present here Underwater Platform to Promote Experimental Research (UPPER): a low-cost (about $25/node) and flexible underwater platform designed to enable cost-effective and repeatable experimentation. We utilize COTS components to provide a HW/SW integrated solution that interfaces our custom hydrophones ($5 ea.) with laptops that act as an SDR-based physical layer, while allowing higher layer protocols to interact via a plug-and-play interface. We show that our platform can communicate over small (5-10m) distances and over a range of data rates (100-600bps). We believe our platform removes the barrier to validating simulation results in underwater environments and also allowing a fair comparison with related protocols.

1. INTRODUCTION

The underwater world has enormous impact on human civilization, with it affecting climate change, food security, minerals and natural resources. However, despite technological strides, we still know very little about this region of our planet. Underwater acoustic sensor networks (UWSN) provide a promising window of insight and observation that hopes to fill this void.

The UWSN domain is, however, extremely challenging due to the large and variable propagation delay, limited bandwidth, low data rate, severe multipath, and doppler spread of the underwater acoustic (UWA) channel [7]. Most importantly, many of the channel conditions are highly variable and non-linear; thus, much like (and perhaps more so than) the wireless networking community [9], many channel assumptions used for simulations are rendered inaccurate and far-removed from any practical scenario.

We thus argue that network protocol evaluation, whether MAC, routing, or transport layer (and beyond), suffer from two deficiencies. First, in most cases we observe these evaluations using *only* simulations to validate a novel idea or protocol. We believe that, due to the complex nature of the underwater channel, these results do not practically validate or provide insights, unless coupled with underwater experiments. Secondly, these evaluations rarely have any concrete comparison with related protocols. If any comparisons are done, they are "apples-to-oranges" because of the different environmental conditions/assumptions in which the original work and comparison is performed.

We believe these deficiencies have two major reasons: prohibitive cost of building a network of underwater nodes where an individual platform cost can range from $2-5K [8, 4]; and the lack of an easily accessible, uniform, platform where experiments and protocols can be easily replicated in the same environment allowing a fair and accurate comparison.

We present here a solution, in the form of an easy-to-replicate Underwater Platform to Promote Experimental Research (UPPER) that can be used to build an experimental testbed (Figure 1). Our platform is, at $25 per platform (Table 3), *two orders of magnitude* cheaper than alternatives with an easily-to-replicate design thus removing the first roadblock to experimenting with a network of underwater nodes. We achieve this ultra-low price point by sacrificing energy consumption and range, and choosing to build a software-defined modem over commodity computers and components. We also provide a flexible and easy to integrate software stack via a simple API. Thus, we envision that existing simulation based protocol implementations can be easily ported to work over our phy-layer using this API. This easy integration will allow experimental validation of existing protocols, and will also allow the research community to compare different protocols over the same physical channel. We envision that with remote accessibility our system can be used by researchers to build shared testbed in several different underwater environments.

Our work has three contributions: first we present a DIY design of an underwater platform with a sub $30 price point and an API for easily accessing the software stack thus en-

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

abling experimental evaluation and comparison. Second, we release the code and design of our system to the research community to replicate and evolve the platform. Finally, we perform micro and macro evaluation of our platform to define its specification, and while in its alpha phase, we currently achieve nearly 6-10m in a narrow water channel, using FSK modulation, and over a wide range (100-600bps) of user selectable data rates. While we have a short range, it can be increased proportional to the cost of our hydrophone. However, we envision our platform to prototype new underwater protocols and systems, providing greater realism with packet loss and interference representative of a real underwater environment. Real application deployments should resort to using longer range and robust solutions.

2. RELATED WORK

Underwater research has, recently, made several efforts in designing low-cost modems and hydrophone. Thus, Benson et al. propose a hardware based design of a low cost modem for short range sensor networks, primarily focusing on the design of a low-cost hydrophone [3]. Similarly, Sanchez et al. proposed hardware based acoustic modem design with the focus on low-cost and power consumption, using commercial echo sounders to reduce the cost of the hydrophone [14]. Willis et al. also propose the design of a low cost hardware modem for short range (100-500m) communication [19]. Borowski and Duchamp implemented a software modem, in which standard TCP or UDP transport protocol runs on top of IP stack that runs on top of custom datalink layer using the computer's sound card and Linux TUN drivers [6]. Torres et al. presented a software defined Underwater Acoustic Networking platform (UANT) [17] that uses GNU Radio, and integrates with TinyOS to provide a sensor network stack. Recent work, similar to ours, promotes smooth transition between simulation and experimentation by providing a transition path from NS-2 to deployment on real underwater modems [13, 11]. However, both efforts require separately purchasing the most expensive components: acoustic modem and hydrophone.

We believe ours is the first work to build on these previous efforts but with a singular focus to promote underwater experiments by significantly lowering the cost barrier. We achieve this goal by a novel design that drastically lowers cost for an experimental platform, albeit exchanging the cost for a shorter range and power consumption. This platform includes the hydrophone, modem, and a network stack allowing easy integration with existing or new protocols.

3. DESIGN GOALS AND DECISIONS

We identify two major goals, to promote underwater experiments, that will guide the design of our underwater platform .

- Build a low cost (sub $50) underwater communication platform, using COTS components. Moreover the design should be simple enough that any research group can rebuild this platform locally. This goal is essential to enable cost-effective and easy reproduction of the platform.

- Build a software stack that provides an API to the physical layer of our platform but also allows easy integration with existing and new protocols. The physical layer should also be remotely accessible for wider usage of a testbed built with our platform.

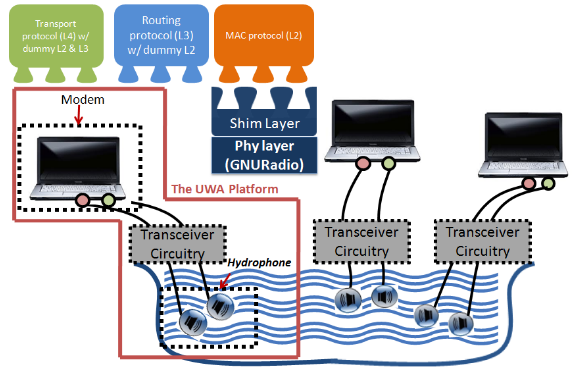

Figure 1: Our vision of an experimental testbed consisting of our Underwater experimental platform (UPPER). Different protocol stacks can use our shim layer to easily integrate with physical layer.

We now explore the design decisions dictated by each of these two high level goals. We assume abundance of power in a lab or experimental environment and thus power-efficiency is *not* a consideration in our design.

3.1 Low Cost Design

Commercial platform are designed for long-range and underwater communication are very expensive, with price point in thousands of dollars [4, 10]. Thus, buying them as off-the-shelf components for multi-node, shorter-range, underwater sensornet experimentation becomes cost-prohibitive. An underwater acoustic platform has two major components in terms of its costs: the modem and an acoustic transducer (Figure 1). Our design seeks to individually reduce the cost of both components.

Acoustic Modem

To implement the acoustic modem we use commodity computers, generally already available, with an SDR (software-defined-radio) implementation to perform the necessary acoustic modulation and demodulation. With all commodity computers possessing a sound card, and the fact that the audible frequency range is appropriate for underwater communication, we get all conversion between the analog and digital domains done by using these sound cards. Thus, using existing compute infrastructure along with a software-defined approach gives us enormous cost savings. Another advantage of using SDR is that it gives the system more flexibility in terms of changing the physical layer parameters.

Hydrophone and Interfacing Circuit

An acoustic hydrophone itself is a very costly component (perhaps *the* most costly [3]) with commercial versions in the range of $500-1,000. We aim to design an acoustic hydrophone with interface circuitry to sound card with a price-point below $50. In line with our goal to use COTS components, we decide to use a commonly available piezo-tweeter used for in-car audio systems that generally cost less than a dollar.

With such a generic piezo-tweeter, we encounter a host of derivative issues. Foremost among them is the need to interface the tweeter with sound card of a computer. Similarly with this choice we have to deal with the directivity of such tweeters and the fact that they are designed as a transmit but not a receive element. We later explain in detail our implementation choices dealing with these issues (Section 4.2).

DIY Design

We also make a conscious decision that our platform design should follow the spirit of do-it-yourself (DIY). This decision requires us to carefully choose the simplest and most common components and publish the circuit diagrams as well as the code for public consumption [1]. We believe that while this decision also lowers the costs some-what, the potential impact for others to build a better platform makes this a sound decision.

3.2 Flexible Stack Integration

A second goal for us is to build a platform with which existing, and new, protocol can be easily integrated. This goal is important for two reasons as it allows: one, existing protocols to seamlessly move their evaluations from simulation to actual experiments and, two, easy and fair comparison of related protocols in the *same* environment. Moreover, we also aim to provide remote accessibility to our platforms. We do so as we envision several testbeds at different locations (ocean, lake, river, pools) formed with collective funding for large scale testing.

We next describe the design decisions that allow us to achieve these goals.

Phy-layer Abstraction using a Shim Layer

Our first design decision to support easy stack integration has been to abstract our SDR based physical layer with a shim layer. This shim layer is responsible for providing a half-duplex, packet interface to the acoustic modem and exports this interface in the form of a well-defined API. Using this API, e.g., the simulation code for any protocol (existing or new) can be adapted to call into our physical layer instead of a simulated environment.

Since several recent protocols employ the concept of tone in their protocol coordination [16, 12], we provide an API to transmit and receive tone (beyond simply data communication).

We believe that the above two decisions will help integrate nearly all existing, and most future, experiments to our modem.

Remote Accessibility using RPC

We extend the shim layer API to be remotely accessible by providing a remote procedure call (RPC) interface to the shim layer. With our well-defined RPC interface, we can achieve universal access to any testbed created using our platform.

We decide to allow only a single remote user to access a platform at a time through the RPC interface. We thus limit the interface as it greatly simplifies the design of the queuing mechanism at the shim layer for potentially concurrent requests.

4. IMPLEMENTATION DETAILS

Our UWSN platform is a complete HW/SW solution and as such we now describe its implementation details along this generic categorization. We consider the implementation of the SDR modem and an easy-to-integrate interface as the software component, while the implementation of the low-cost hydrophone and the interface circuit to computer as the hardware component.

9[1]Detailed code and design document at `http://sysnet.org.pk/w/Code_and_Tools`

Table 1: Our phy-layer API provided as RPC functions.

API Calls	Description
SendData(*String* msg)	transmit *msg*
String ReceiveData()	Returns data received
SendTone()	transmit tone
int ReceiveTone()	Returns tones received
Config(*double* DataRate, *double* CentreFreq, *double* Amplitude)	Configure modems data rate, center frequency, and amplitude

4.1 Software Components of our Platform

Figure 2 shows an over view of the software components of our experimental platform, clearly divided into two parts. The first part implements a *shim layer* to provide a simple, remotely accessible, packet interface to the physical layer while managing congestion using queues. The second is a GNU Radio based *software modem* that acts as a physical layer and is responsible for packetizing data as well as modulation and demodulation of the acoustic signal.

We next describe each of these components in detail.

4.1.1 Shim Layer

Our shim layer presents the external interface that allows users to interact over an underwater medium. This layer is responsible for interaction with both the external users and our software modem, while employing some mechanism for flow and rate control. This functionality is implemented by shim layer in three parts: an RPC server, a GNU Radio interface, and a queue management system.

The RPC server presents a packet interface with an API shown in Table 1. Thus any layer can send packets over our platform using RPC calls subscribing to our API. The API functions encode whether the packet is tone or data; a tag is added to a special packet header and pushed to the transmit queue. We have made the receive API's asynchronous; thus a user can setup a receive call which will be notified when a tone or data packet is pushed into the Queue by the GNU Radio interface. Any configuration parameters are directly channeled to the software modem.

We implement transmit and receive queues inside the shim layer to provide rate control. Thus, if a user sends more packets than can be handled by the phy-layer, our shim layer buffers them and transmits when available. There is a single transmit queue for both tone and data, and separate ones on the receive side so that the async implementation of RPC server can directly poll the respective queues.

Finally, the GNU Radio interface communicates with the software modem over a UDP socket. This part is responsible for ensuring the appropriate fetching of packet from the transmit queue, deciding if it is a tone or data transmission, and then forwarding it to the modem. We also ensure that we dequeue packets after the packet's transmission time. This part also receives tone or data from the modem and puts them in their respective receive queues from where the RPC server can pull them.

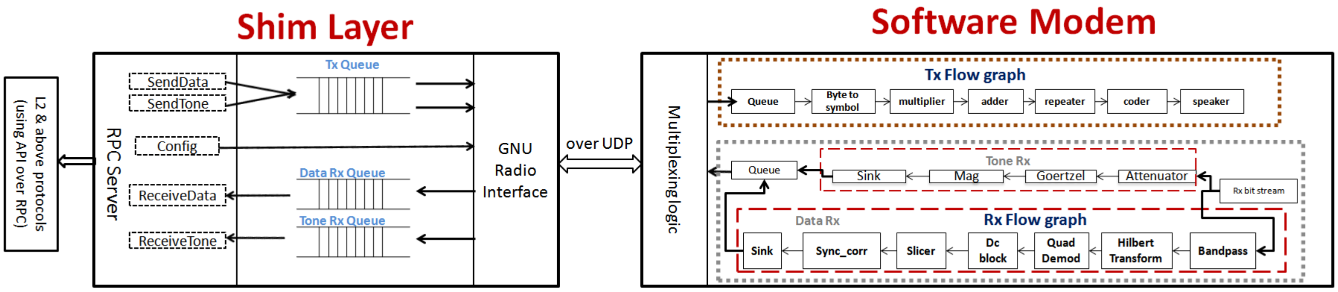

Figure 2: Overview of the Software Components

Table 2: Parameters of Modem

Properties	Assignment
Modulation	FSK
Frequency Spacing (Data)	2KHz
Data/Tone Frequency	15.5KHz, 17.5KHz/ 16.5KHz
Data Rate	100bps-600bps
Communication scheme	Half Duplex

4.1.2 *Software Defined Acoustic Modem*

Our software-defined acoustic modem consists of three subparts. The first is the multiplexing logic that interacts with the shim layer and also handles data/tone and transmit/receive multiplexing. The other two parts are the transmit and receive flow-graphs for which we use GNU Radio, a free and open-source implementation of and SDR [5]. GNU Radio provides all signal processing blocks needed for modulation and demodulation. We first briefly describe our modem parameters and then discuss these parts in detail.

Modem Parameters

Table 2 presents key modem parameters that represent our platform's physical layer. We choose FSK (frequency shift keying) modulation in our modem implementation for its implementation simplicity and because it is robust to multipath for the short distances intended for our experimentation [19]. The choice 15.5 to 17.5KHz for data and tone is dictated by the need to stay within both the soundcard frequency range and best utilize the bandwidth of our hydrophone (Section 5.1.1). Both modulation frequency and data rate are user configurable allowing flexibility to different hydrophone designs. The modulation scheme is not currently configurable; however with an opensource code-base researchers will be free to experiment with different modulation schemes for physical layer experimentation.

Multiplexing logic

One of the major constraints we faced was to time-multiplex the access to sound-card, a shared resource, by the transmit and receive flow-graphs. At the same time we also want to ensure half-duplex communication to prevent our receiver hearing our transmission.

To solve this problem, we can only have one flow-graph active at any given time. Thus when we get a packet for transmission from the shim layer we stop the receive flow-graph, otherwise listening, for the duration of the packet transmission. Once transmission is finished, we stop the transmit flow-graph and restart the receive flow graph. This part is also responsible for implementing the configuration parameters received directly from the shim layer.

A final responsibility of this part is to take packet from the receive queue and present them with the proper tags to the shim layer.

Receive flow-graph

Here we have a similar problem as above, i.e., two flows (data and tone receivers) that need to access the sound card simultaneously. Using the hierarchical block mechanism present in GNU Radio, we simultaneously present the bit stream received from the sound card to the two branches, for tone and data, of the main flow-graph. Data receive flow-graph processes the bit stream via the bandpass, demodulation, and correlation modules provided by GNU Radio itself. The result is finally presented to a modified version of framer-sink example from GNU Radio that packetizes the bit stream. The packet is then passed to the multiplexing logic via a queuing mechanism.

We implement tone detection by declaring a tone if a sufficient signal strength is received at the tone frequency. This detection is implemented using the Goertzel's algorithm. We observed that our tone detection was triggered even when data was being received. This false detection occurs as we position the tone frequency in the center of the mark and space frequency for our modem to most efficiently use our limited bandwidth. We use an attenuation block that precedes the tone sub-graph to reduce such detections. We set the attenuation factor empirically by ensuring the we detect data at close range but not detect tones.

Transmit flow-graph

Like the receive flow-graph, we experience the problem of handling tone and data flow graphs simultaneously accessing the sound card. We observe, however, that both data and tone flow graphs use identical signal processing blocks but with different specifiers for the multiplier and adder block. Thus instead of using the more complicated solution of a hierarchical block to split a flow graph, we use a single transmit flow-graph. We set different parameters for the multiplier and adder block values in the multiplexing logic which sets these based on whether the request is to transmit data or tone.

4.2 Hardware Components of our Platform

The hardware design is a major concern for us as it dictates the cost of our overall platform. Our hardware design goals primarily focus on achieving low cost design as well as providing an effective acoustic communication plat-

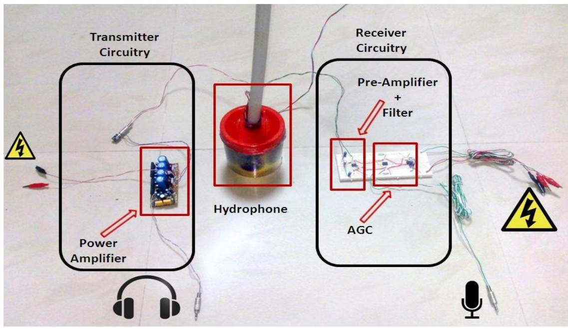

Figure 3: The hardware components of our platform; a low-cost hydrophone with sound-card interface circuitry

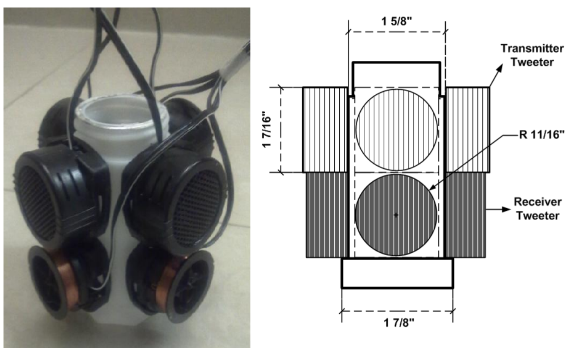

Figure 4: Hydrophone with eight tweeters mounted on a plastic housing.

form. Our hardware design consists of a DIY hydrophone, along with transmitter and receiver circuitry as shown in Figure 3. Transmitter is getting input from the headphone jack of sound card while the receive circuit presents the signal to the sound card over the mic-in jack. Currently our circuitry is operating at +12V at transmitter and ±12V at receiver side.

We next explain the goals and resulting choices of these two major sub-components.

4.2.1 A low-cost Hydrophone

Hydrophone design is a major cost component for an underwater experimental platform. Our hydrophone design (the final design in shown in Figure 4) should:

1. Be suitable for underwater communication.
2. Be significantly low cost (sub-$10).
3. Match sound-card/audible frequency range.
4. Behave as an omni-directional transducer.
5. Remain water-proof at short depths.

Our first goal influences the choice of one of electrostatic, piezoelectric. or magnetostrictive materials to be used for underwater communication. Our research shows that from the above, piezoelectric material is best suited for underwater communication as it produces high pressure in response to applied electric signal (and vice versa) while matching the high acoustic impedance of water [15]. Similarly piezoelectric materials have a high electro-mechanical coupling factor ensuring high power efficiency, and exhibit linearity in energy conversion over a wide range of input signal.

The second and third goals simultaneously affect our choice of the piezoelectric element we use for our hydrophone. As the resonant frequency of the piezoelectric ceramic reduces, its size and, therefore, cost increases. We thus have the choice to either purchase an expensive piezoelectric ceramic resonant at 20KHz; or purchase a lower efficiency piezotweeter whose operating frequency is shifted down to 20KHz range by adding an LC circuit to piezoelectric ceramic.

Since our prime focus is to lower cost and enable experimental underwater research, we chose the latter — in effect trading the range of our system for a significantly lower cost. We thus choose a COTS, car-audio piezo-tweeter (Semtoni Tsp-003,$0.50/unit) as the acoustic element for our hydrophone [2].

The fourth goal for an omni-directional hydrophone is to enable networked, and not just point-to-point, experiments. However, our tweeter element is not omni-directional. Furthermore, it includes a step up transformer to boost the voltage applied to the piezoelectric ceramic. This optimization however results in receive signal being stepped down, and thus lowering receive sensitivity. We propose a design where we combine four of these tweeters to provide a capability to communicate in all directions (details in Section 5.1.2). Furthermore, we utilize another set of four piezo-tweeter elements where the transformer has been removed. We mount these 8 piezo-tweeter elements (4 for transmit and 4 for receive) on a commonly available plastic container to setup a hydrophone. The final result and the dimensions are shown in Figure 4.

Our final design goal was to enable a waterproof and electrically safe packaging. We chose, inline with our cost lower and DIY approach, to use a plastic container, filled with vegetable oil, in which the 8 element structure was immersed and screwed in. We chose vegetable oil as it is a cheap, and commonly available, potting material with density and impedance similar to water, thus allowing a high energy transfer from the tweeter to the aquatic environment. We extract the wires through a hole at the top of the container, and use silicon to water proof the container. As our focus is in-lab experimental setup, the above design suffices for the depths and controlled scenario we expect to run experiments. However, we recommend a better quality container with stronger water proofing for deep water experiments.

4.2.2 Sound-card interface circuit

Our circuitry that interfaces to the sound card (through the headphone and microphone ports) consists of a power amplifier on the transmit path. On the receive path we have built a low pass filter, a pre-amplifier, and an automatic gain control (AGC).

We next describe each of these in detail.

Transmitter Circuitry

We first observe that the signal output from the sound card is, by design, low powered. Thus, this signal needs to be amplified to get any decent communication range.

The choice of a power amplifier, in general, is dependent on two factors: linearity and efficiency. We choose a class AB amplifier which achieves good linearity and efficiency while delivering around 19W to a 4Ω load. While we could

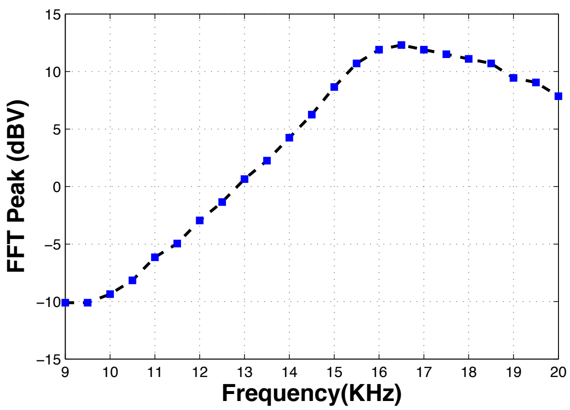

Figure 5: Frequency Response of our chosen low-cost acoustic tweeters. (FFT at 100KSamples/sec)

have followed this up with a class D amplifier that increases efficiency (like Benson et al. [3]), we chose not to as we assume abundant power availability and also to keep the cost down.

Receiver Circuitry

We face several problems on the receive path from our hydrophone. First, the voltage of signal received from our piezo tweeters at large distance is quite low (less than the 0.5mV sensitivity of our oscilloscope). This signal also has lots of out-of-band noise present. Finally, since the sound-card input is saturated for any voltage exceeding 37mV RMS, we have to deal with distance related variation in received signal strength.

We utilize a combination of a pre-amplifier and a low-pass-filter (LPF) to handle the first two problems. Our designed pre-amplifier provides a 27dB gain to the received signal. We then use an LPF with a cutoff frequency of 20KHz (audible/sound-card range) to filter off any out-of-band noise.

The amplified signal, due to distance dependent signal strength cannot be directly tuned to the saturation capacity of the sound-card. We deal with this issue by inserting an automatic gain control (AGC) circuit between the LPF and sound-card input. We design the AGC to maintain a constant 35mV RMS at its output, even if the input signal is much larger. Thus, a received signal amplified greater than 35mV is automatically attenuated, while any signal below 35mV is not boosted by the AGC.

5. PLATFORM EVALUATION

We now evaluate our underwater platform to validate our claims. We first answer some questions regarding the designed low-cost hydrophone. Then we evaluate the overall system in terms of its communication range and cost.

5.1 Hydrophone Evaluation

In this section we perform some micro-experiments to understand the characteristics of both our low-cost hydrophones.

5.1.1 *Frequency Response of our piezo-tweeter*

We first find the frequency response of our modified (without transformer) piezo tweeter. Note that the frequency response of the tweeter, as a transmitting element, is provided

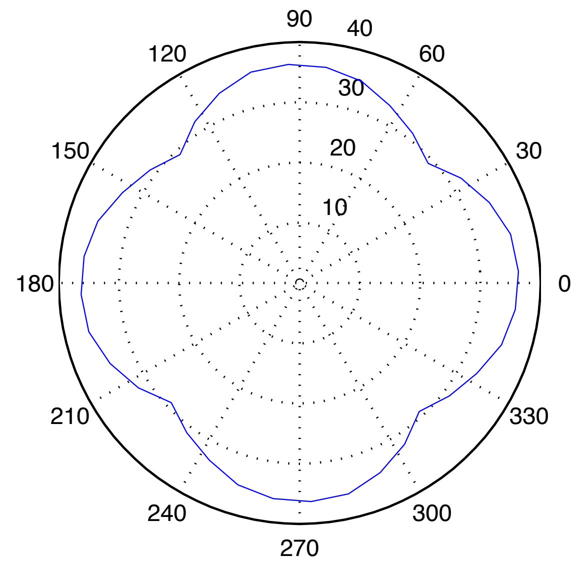

Figure 6: Transmitter Directivity Pattern of our hydrophone with four piezo elements (in dBm and at 15.6KHz)

in its spec sheet and shows a nearly flat response between 15-19KHz.

We connect a tweeter to a signal generator whose output frequency is changed in increments of 0.5KHz. We then observe the FFT of the received signal at another tweeter placed in front of the transmitting element on our oscilloscope. Figure 5 shows the result, where we can see that the maximum frequency response is at around 16.5KHz. This frequency response validates our choice of 15.5-17.5 KHz as the frequency band for our modem (Table 2).

5.1.2 *How Omni-directional is our Hydrophone?*

We have mentioned in Section 4.2.1 the design of our hydrophone where we mount four piezo elements to provide an omni-directional communication capability. We now evaluate this capability by measuring the directivity of our low-cost, custom built hydrophone.

We measure the pattern by applying 15.6KHz to a hydrophone tweeter with a receiver tweeter right in front of it. We complete a rotation of the hydrophone in steps of 5°, observing the receiver output voltage, on an oscilloscope, at each increment. We then plot the attenuation of the received voltage on a radial plot. A similar procedure is followed for receiver directivity pattern by simply swapping the transmitter and receiver hydrophones.

Figure 6 shows the resulting transmit directivity pattern of our hydrophone. It is quite apparent that the hydrophone has nearly omni-directional communication capability with just -6dBm difference between the maximum response to the minimum value between two adjacent hydrophones. Transmitter and receiver hydrophone exhibit similar directivity patterns. We can further improve this response, if the situation requires, by employing more than four elements.

5.2 Overall Platform Evaluation

We now perform some macro or system level experiments to evaluate our underwater platform that presents a packet-

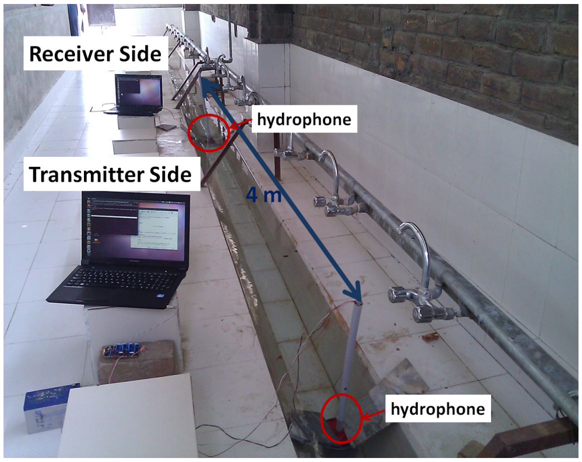

Figure 7: Experimental Setup to evaluate our platform.

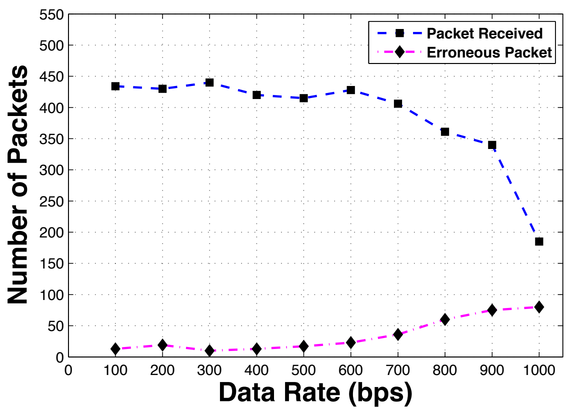

Figure 8: Impact of varying data rate (@ 4.5m).

based physical layer. We begin by reporting the impact of distance and data rate on packet reception. We also evaluate the total cost of our platform.

5.2.1 *Evaluation setup*

Our platform's evaluation setup is shown in (Figure 7) We perform platform evaluation experiments in a water channel (used for purpose of ablution, but conveniently located in our university) measuring $10 \times 0.12 \times 0.3$ m in length, width, and height, respectively. For each data point in our experiments, we send 500 packet and compute the packet received successfully, packet received erroneously (demodulation fails), and packets lost.

5.2.2 *How does data rate impact packet reception?*

We first evaluate the possible data rates for our platform. We fix the distance between our two nodes, and configure the system to send packets at data rates from 100-1Kbps, using increments of 100bps.

Figure 8 shows the result of our experiment. We observe that from 100-600bps our system consistently receives about the same number (90%) of packets. Thereafter the performance degrades, with about 35% packets being received at 1Kbps. This result is encouraging, and for this reason we next perform range testing at both 100 and 600bps.

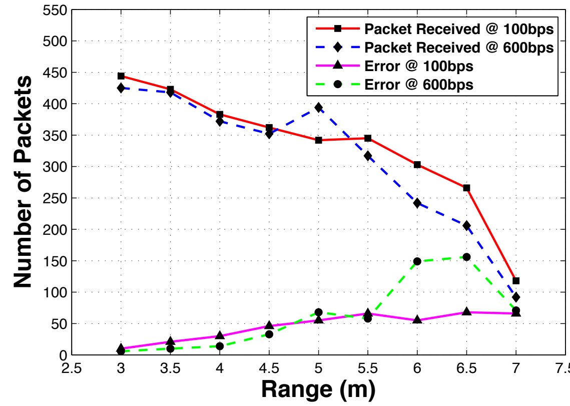

Figure 9: Range testing of our experimental platform (@100bps and 600bps)

5.2.3 *What is the range of our Platform?*

We now try to ascertain the communication range of our system. We test range of our platform at 100 and 600bps by looking at packet receive rate (PRR) and packet error rates at different distance between our nodes. We start at a distance of 3m and, with increments of 0.5m, perform our experiments until 7m.

Figure 9 shows the result of our experiment. We see that for both data rates, we have similar performance. We have nearly 90% PRR at short end of the range that gradually decreases to just 20% at 7m.

We do point out that these results are for our very linear water channel, that should exhibit very high multipath. We believe that in a more open environment, like a swimming pool, the range of our system will be much greater. Similarly, we plan to improve our understanding of our system by repeating these experiments at different environments.

5.2.4 *Cost Evaluation*

We now present a cost evaluation of our platform using a bill of materials (BOM) analysis. As a comparison, available modems range from \$2-5K (research and commercial) while just the hydrophone themself cost around \$600 [3]. As Table 3 shows, our design choices lead to a nearly *two orders-of-magnitude* cheaper underwater platform. We have achieved our drastically low cost by designing a COTS based, ultra-cheap hydrophone and interfacing it to a commodity (and already available) computer's sound-card. We can also forsee implementing GNU Radio-based modem implementation over a Raspberry Pi platform (\$25/unit) to marginally increase the cost if such computers are not freely available [1].

6. CONCLUSION AND FUTURE WORK

In this paper, we argue that the current focus of researchers on evaluating new underwater protocols using *only* simulations is — while understandable due to cost constraints — insufficient due to the vagaries of the underwater acoustic channel. We present here a solution, in the form of an easy-to-replicate Underwater Platform to Promote Experimental Research (UPPER) in UWSN. We design a hydrophone using COTS elements and interface with a computer's sound

Table 3: BOM cost for our complete platform.

Hardware Components	Cost
Laptop	Already available = $0
Modem	implemented in GNU Radio = $0
Hydrophone	8 Tsp310=$7
Transceiver circuit	$15
Miscellaneous	$3
Total	$25

card, where we implement a GNU Radio based, software-defined acoustic modem. Using COTS components and assuming availability of computers, our platform costs around $25. With its low cost and easily replicable design we hope researchers will locally reproduce this platform and perform multi-node experiments. Furthermore, with an easily accessible physical layer that allows us to easily integrate existing and new protocols, simulation results can be strengthened and we can also have fair protocol comparisons.

We hope that in the future the community will use, and extend, the design of our low-cost underwater experimental platform. We are currently working on improving the transceiver circuit and also using better (albeit more expensive @ $35/unit) piezo-elements to increase the range of our system to 25-50m range. We are also working on a web-interface that will allow, much like Emulab [18], researchers needing experimental data to remotely login and be assigned specific nodes from a large testbed consisting of UPPER nodes. We envision such testbeds being deployed in different underwater environment, thus allowing even higher confidence in protocol evaluation.

Acknowlegment

We would like to thank our colleagues in the SysNet Lab (especially Naveed Anwar) for their co-operation in carrying out underwater experiments. We also thank Andrew Goodney and acknowledge modifying his original GNU Radio code in our implementation.

7. REFERENCES

[1] *Raspberry Pi: An ARM GNU/Linux box for $25.* http://www.raspberrypi.org/.

[2] *Semtoni Products [Online] Available.* `http://www.semtoni.com/product.jhtm?id=129&cid=27l`.

[3] B. Benson, Y. Li, R. Kastner, B. Faunce, K. Domond, D. Kimball, and C. Schurgers. *Design of a low-cost, underwater acoustic modem for short-range sensor networks.* IEEE, 2010.

[4] Benthos, Inc. `http://www.benthos.com/pdf/Modems/ModemBrochure.pdf`.

[5] E. Blossom. Exploring gnu radio, April 2009. `http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html`.

[6] B. Borowski and D. Duchamp. The softwater modem: a software modem for underwater acoustic communication: short paper. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, WUWNet '09, pages 12:1–12:4, New York, NY, USA, 2009. ACM.

[7] J. Catipovic. Performance limitations in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering*, 15(3):205–216, Jul 1990.

[8] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092 Vol. 2, 2005.

[9] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '04, pages 78–82, New York, NY, USA, 2004. ACM.

[10] LinkQuest, Inc. *Underwater Acoustic Modems.* `http://www.link-quest.com/html/uwm_hr.pdf`.

[11] 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 *Proceedings of IEEE OCEANS 2012*, pages 1 – 10, Yeosu, Korea, May 2012.

[12] D. Mirza, F. Lu, and C. Schurgers. TB-MAC: Efficient mac-layer broadcast for underwater acoustic sensor networks. In *5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 249–254, dec. 2009.

[13] C. Petrioli, R. Petroccia, J. Shusta, and L. Freitag. From underwater simulation to at-sea testing using the ns-2 network simulator. In *Proceedings of IEEE OCEANS 2011*, pages 1 – 9, Santander, Spain, June.

[14] A. Sanchez, S. Blanc, P. Yuste, and J. J. Serrano. A low cost and high efficient acoustic modem for underwater sensor networks. In *OCEANS 2011 IEEE Spain*, pages 1–10. IEEE, 2011.

[15] C. Sherman. Underwater Sound - A Review I. Underwater Sound Transducers. *IEEE Transactions on Sonics and Ultrasonics,*, 22(5):281 – 290, September 1975.

[16] A. A. Syed, W. Ye, and J. Heidemann. Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks. *IEEE Journal of Selected Areas in Communication*, 26(12):1731–1743, December 2008.

[17] D. Torres, J. Friedman, T. Schmid, and M. B. Srivastava. Software-defined underwater acoustic networking platform. *Proceedings of the Fourth ACM International Workshop on UnderWater Networks WUWNet 09*, pages 1–8, 2009.

[18] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002. USENIX Association.

[19] J. Wills, W. Ye, and J. Heidemann. Low-power acoustic modem for dense underwater sensor networks. In *Proceedings of the First ACM International Workshop on UnderWater Networks (WUWNet)*, pages 79–85, Los Angeles, California, USA, September 2006. ACM.