Low-Power Acoustic Modem for Dense Underwater Sensor Networks[*]

Jack Wills, Wei Ye, and John Heidemann
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{jackw, weiye, johnh}@isi.edu

ABSTRACT

Significant progress has been made in terrestrial sensor networks to revolutionize sensing and data collection. To bring the concept of long-lived, dense sensor networks to the underwater environment, there is a compelling need to develop low-cost and low-power acoustic modems for short-range communications. This paper presents our work in designing and developing such a modem. We describe our design rationale followed by details of both hardware and software development. We have performed preliminary tests with transducers for in-air communications.

Categories and Subject Descriptors

B.4.1 [**Input/Output and Data Communications**]: Data Communications Devices

General Terms

Design, Measurement

Keywords

Underwater networks, wireless sensor networks, low power, acoustic modems

1. INTRODUCTION

Sensor networks are beginning to revolutionize data collection in the physical world, with examples including monitoring of micro-habitats, soil, buildings, and machinery. While these applications are changing science and industry on the surface, relatively little work has been done to explore how sensor networks apply underwater.

We are interested in bringing the benefits of terrestrial sensor networks underwater: wireless communication, dense deployments (each sensor may have eight or more neighbors), self-configuration and local processing, and maximizing the utility of any energy consumed. Our primary application is seismic monitoring, with alternative applications including assistance during underwater construction, pipeline and leak monitoring, biological data collection, or underwater robot communication [8]. Sensor networks typically consist of many battery-powered nodes, densely deployed in an area for close observation and long-term monitoring.

The underwater acoustic channel presents strong challenges to the design of data communication networks (see [13] for a survey of details). Besides severe multipath reflections, there can be curved propagation paths due to uneven temperature distribution and various interference, such as bubbles and noise from man-made objects. One approach to addressing these challenges is to use increasingly sophisticated coding and other physical-layer techniques. However, a potential penalty of this approach is that individual modems become quite expensive and power-hungry, making use of hundreds of modem-equipped sensors economically infeasible. We therefore explore a complementary path that emphasizes *simple but numerous* devices that benefit from dense sensing (*e.g.*, eight or more neighbors per node, rather than one or two) and shorter-range communication. In addition to simpler node-to-node channels due to shorter range, higher-level approaches can compensate for channel problems through approaches such as routing, link-layer retransmission, and application-layer coding. For example, routing helps to avoid bad links in relaying data.

In addition to the unreliable communications, there are a number of other research challenges that must be met to accomplish real-world applications, including new work in high-latency network protocols (MAC, time synchronization, etc.), application design to accommodate very low bandwidth, long-term operation, and low-power hardware design. We have previously described these challenges [8]; this paper examines in detail the design of a new underwater acoustic modem targeted particularly at supporting short-range acoustic communication for dense underwater networks. We describe our design rationale (Section 2), hardware and software designs (Sections 3 and 4), and preliminary tests (Section 5). Although many other acoustic modems exist (we review related work in Section 6), we believe we are the first to focus on underwater communications hardware that captures the characteristics of terrestrial sensor networks.

---

[*] This research is partially supported by the National Science Foundation (NSF) under grant number NeTS-NOSS-0435517, “Sensor Networks for Undersea Seismic Experimentation (SNUSE)”. It is also partly supported by Chevron Co. through the USC Center for Interactive Smart Oilfield Technologies (CiSoft).

WUWNet’06, September 25, 2006, Los Angeles, California, USA.

2. DESIGN RATIONALE

Our overall goal in the design of our underwater modem is to bring the characteristics that are being exploited in terrestrial sensor networks underwater. “Traditional” wireless platforms such as 802.11 and cellular telephones use sophisticated radios with custom DSPs and assume relatively powerful CPUs (32-bit x86 or XScale). By comparison sensor network platforms like the Mica-2 [5] adopt very simple radios (the CC1000, 38.4kb/s baseband speed) and 8-bit microcontrollers (the Atmel Atmega128L) with much of the link layer and higher implemented in the host CPU. This shift in design allows each sensor platform to be very inexpensive (today, less than \$100/node), allowing dense deployments of hundreds of sensors, each with eight or more radio neighbors.

Thus, our primary goal is that the *modem be inexpensive* to make it feasible to purchase and deploy many sensor nodes. Our target price point is that the modem cost as much as the CPU and packaging, around \$30–100. A corollary is that we need only *short-range communication*, since long-range communication can be accomplished by multihop routing over many individual nodes. Fortunately, these choices reinforce each other, because focusing only on short-range communication means we expect to avoid many of the challenges of long-range communication (for example, acoustic ducting and multi-path effects due to surface reflections and temperature gradients), greatly simplifying the modem design. Our target communication range is 50–500m.

We describe the implications of cheap, short-range communication on our hardware design below in Section 3. Most importantly, it allows us to use a simple modulation scheme (FSK) and simple, non-coherent detection.

Our secondary goals stem from the application: *low-power operation* to allow long-lived monitoring, *support for higher-level protocols* in software, and *design for expected channel characteristics*.

Our design uses several techniques to accomplish low-power operation. Our main power-saving innovation is to use a dedicated, very low-power, all-analog *wakeup tone receiver* to trigger the more expensive data receiver. When there is no communication activity, nodes can turn off most components, and only leave the wakeup receiver on. This idea has been explored before based on the concept of a low-power radio [16], but is not currently used in most underwater or terrestrial sensor networks. Second, we assume the radio will be duty cycled, alternating on-and-off periods frequently when active, and with long-term (hour or days) off periods when inactive. We design for rapid activation to support duty cycling to avoid idle listening and overhearing when active [23]. Finally, we support software control of transmit power since transmit power is a significant cost with underwater transducers.

Our modem design includes several features to support higher-level protocols. We plan to integrate the wake-up tone receiver with duty-cycling network protocols and MAC design. We export information about receive signal-strength (RSSI) to support power control and channel quality estimation. In addition to saving power, a controllable transmit power supports software control over network density, allowing nodes to moderate the traffic in its neighborhood. Finally, we provide both analog and digital signal output from the modem to allow high precision time synchronization (for protocols such as TSHL [21]).

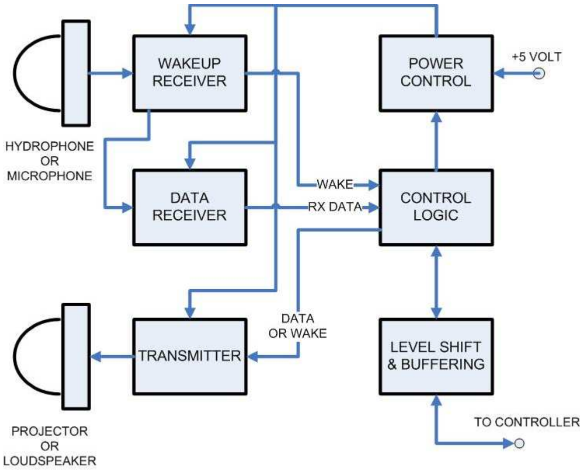

Figure 1: Block diagram of the modem hardware.

Finally, we of course match our design to the expected characteristics of the underwater acoustic channel. Since all coding is in software, we can provide variable bit-rate communication to trade off bit-rate against reliability. Potentially we can add software-based forward-error correction as well.

Since our modem is designed for short-range, dense sensor networks, it does not directly apply to applications that require long-range, reliable, point-to-point communications. For such applications, one should either use existing work on more powerful acoustic modems, or use our modem with complementary, multi-hop communication.

3. CIRCUIT DESIGN AND IMPLEMENTATION

The modem hardware is split into three main portions: a wakeup receiver, a data receiver, and a single transmitter. The transmitter has three output frequencies, which correspond to the data mark, data space, and wakeup tone. It is not possible to transmit data and the wakeup tone simultaneously.

Figure 1 is a block diagram of our modem hardware. The entire circuit operates from a 5 volt power supply. The power control block allows software control of power distribution so that the wakeup receiver, data receiver and transmitter can be independently turned off or on. Level shifters are used to provide compatibility with CMOS logic levels between 2.8 and 5.0 Volts.

Our current prototype contains all the hardware on a single printed circuit board measured as 4 by 5 inches. (The board was designed to simplify debugging and testing; a production design could be much smaller.) Figure 2 is a picture of the board with the wakeup receiver and data receiver installed. We next describe the details of each major part of the modem.

3.1 Wakeup Receiver

The principal goals for the wakeup receiver are good sensitivity and very low power consumption. The only purpose of the receiver is to monitor the total energy level present in a narrow band of frequencies, and to produce an interrupt

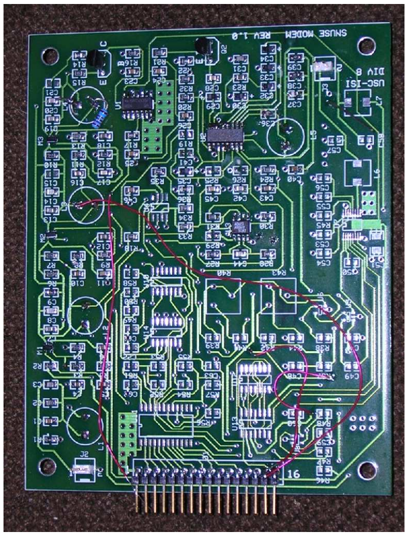

Figure 2: A picture of our prototype board with the wakeup receiver and data receiver installed.

signal when the energy exceeds some preset threshold level. The optimum threshold level depends on the bandwidth of the receiver, and the operational system noise floor. The noise floor is normally set by the background noise of the ocean, rather than the receive electronics [11].

The receiver must be compatible with standard hydrophone design. Because of the high acoustic impedance of water, piezoelectric transducers are the norm. They have a very large source impedance and must be operated into a high impedance amplifier input.

We have chosen 18 kHz as the frequency for the wakeup tone. This is an attractive frequency based on the background noise levels, as well as the attenuation characteristics in the ocean; both factors are frequency dependent [3].

This frequency also lies in the normal audio band (20-20kHz) and allows use of standard audio hardware and software.

Our chosen bandwidth for the wakeup receiver is about 300 Hz. At a carrier frequency of 18 kHz this corresponds to a quality factor (Q) of 60. There are several possible ways to produce such a filter

- **L/C** with passive inductors and capacitors
- **Active RC** using operational amplifiers
- **Digital**—an ADC followed by a DSP

The need for very low power argues against the active RC and digital designs. We choose to apply L/C filters in an architecture reminiscent of the early days of radio. Before that advent of the Superheterodyne radio in the 1930's, common practice was to use a Tuned Radio Frequency (TRF) design. Each amplifier stage had a resonant L/C load to provide both amplification as well as filtering.

We have used this architecture using MOSFET transistors so we have the very high input impedance need for piezoelectric transducers. We cascade stages to get the necessary gain. Miller feedback caused by the parasitic capacitance from base to drain in the FET will cause unwanted interactions between the filter stages. We avoid this by using a cascade amplifier stage, in which a second FET acts as a common base amplifier which improves isolation between stages.

We wish to use commercially available, off the shelf, semiconductors. The most suitable transistor we have been able to locate is the BF998 dual gate FET. This is a N-type depletion mode transistor, normally used as an AGC RF amplifier in TV tuners. In this application it is biased around 10 mA of drain current. We want to use much less power, so we are using a self bias design with a 100 kOhm source resistor, which operated the transistor at about 30 microamps. So far we have not had to do any special selection of transistors.

The filter performance is limited by the loss (or Q) of the inductor. The best device we have found is a 33 milliHenry inductor made by Fastron, part number 09P-333J-50. We resonate this with a nominal 2700 picoFarad capacitor. Measured Q at 18 kHz is about 50.

We use three stages of amplification and then pass the signal into an AM detector. The AM detector is the base-emitter junction of a 2n3906 bipolar transistor. This transistor is normally biased off, but when the peak input amplitude reaches 0.7 volts it begins conducting, and the collector voltage of the transistor moves from ground up to about 4.5 volts. This signal is buffered through 2 stages of CMOS inverters (74HC04) to provide standard logic levels and fanout.

The complete circuit draws 100 microamps at 5 volts when the wakeup tone is not present. When the wakeup tone is present, the current drain increases as the bipolar transistor turns on.

3.2 Data Receiver

The data receiver is a conventional design based on a commercial FM intermediate frequency demodulator chip, the Philips SA604A. The input to the SA604A comes from the first stage of the wakeup receiver, which provides the necessary selectivity. Whenever the data receiver is turned on, the first stage of the wakeup receiver is also powered.

The first stage of the wakeup receiver has considerable gain variation across the data bandwidth, but the SA604A has internal limiting amplifiers prior to the demodulator, so the gain variation is eliminated. The limiting amplifiers also provide receive signal strength information (RSSI). The RSSI signal is delivered to the microcontroller.

The SA604A is designed for use in narrow band FM systems. Due to the channel characteristics in the underwater environment we are sending wideband FM. This requires several changes in the way we apply the SA604A. First we use a simple, single pole low pass and single pole high pass filter to couple between the stages of the SA604A. A narrow band design typically uses an LC resonator or ceramic bandpass filter. The demodulator use a quadrature coil as a phase shift network. In our design we add an external resistor in parallel with the quadrature coil to increase the bandwidth of the demodulator. Finally we see significant carrier feedthrough at the data output terminals of the SA604A. The SA604A has two differential output terminals. We have added an LC notch filter tuned to 18 kHz between these terminals. The output signals are then combined in an AD623 instrumentation amplifier to obtain a single ended analog output. This output is made available to the microcontroller, which normally will sample the signal with an analog-to-digital converter (ADC). The modem also includes a comparator operating on the analog output which acts as a data slicer. The output of this comparator is an unclocked CMOS logic signal. It is also made available to the microcontroller, and allows operation without an ADC. The data receiver consumes less than 4 milliamperes at 5 volts.

3.3 Transmitter

The transmitter uses a Linear Technology LTC6900 low power oscillator as a voltage controlled oscillator (VCO). The circuit design is based on Linear Technology Design Note 293 [17]. The oscillator output feeds into a Texas Instruments TPA2000D1 Class D Audio Power Amplifier. This is capable of delivering 2 watts into a 4 Ohm load. The amplifier gain is set by two digital control signals, which can select gains of 6, 12, 18, or 23.5 dB. By selecting lower gains we reduce the output power level, but extend battery life. We hope that the combination of RSSI and variable output power will encourage development of energy efficient communication protocols. The transmitter efficiency ranges from 80 to 90 percent.

3.4 Transducers

In the ultimate application of underwater communications, we will use piezoelectric transducers. These are high impedance devices, and the modem circuity is designed for high impedance operation.

For lab tests and software development it is convenient to communicate through air instead of water. At the present time we are using Audax brand hi-fi tweeters, both as transmitter and as microphones. Switching over to hydrophones will only require changing the input and output impedance matching networks

3.5 Power Control

The modem operates from a single 5 volt supply. The choice of supply voltage is driven by the dual gate FETs used in the wakeup receiver. These are operated from a 12 volt supply in their intended application. We have been able to use them successfully at 5 volts but the gate threshold voltages of the transistors make 3 volt operation impractical. There is no technical reason that prevents one from building transistors with reduced threshold voltages, but the market demand is too small at the present time to interest semiconductor manufacturers.

While the modem is basically a 5 volt design we need to interface with microcontrollers such as the Mica2 mote. The mote nominally operates on 3 volts. The precise operating voltage of the mote is rather unpredictable, as it runs directly from a battery, without any form of voltage regulation. So the actual battery voltage depends on the charge/discharge state of the battery.

The modem design includes two features to allow interfacing to any voltage level from 2.8 to 5 volts. Digital input and outputs are tied through a Texas Instruments SN74TVC3010 voltage clamp which limits all digital output signals to the microcontroller supply voltage. The modem also includes a voltage regulator which is referenced to the microcontroller supply voltage. This regulator drops 5 volts down to the microcontroller voltage, and is used to power all of the analog output circuitry of the modem. This ensures that the analog outputs will never exceed the microcontroller supply voltage.

Power distribution in the modem is digitally controlled. There is independent control of power to the wakeup receiver, data receiver, transmit VCO, and transmit output amplifier.

State	**Control Pins**		
	Rx/Tx Mode	Data Channel	Wakeup Channel
Everything is off	0	0	0
Wakeup receiver is on	0	0	1
Data receiver is on	0	1	0
Both receivers are on	0	1	1
Transmitter is ready	1	0	0
Sending wakeup tone	1	0	1
Sending data	1	1	0

Table 1: Power control and valid modem states

4. CONTROL AND COMMUNICATION SOFTWARE

The hardware described in section 3 provides the basic functionality of transmission and reception of raw bits. In order to provide packet-level communication, we implement other essential components in software on a general purpose microcontroller that runs TinyOS [9, 7]. This approach leverages on-going efforts in sensor networks, and gives us the flexibility to implement and switch between different low-level components, such alternative coding schemes (simple Manchester code or error-correction codes). This section describes the interfaces of the modem and the microcontroller and basic control and communication software.

4.1 Interface with Microcontroller

We first describe the control and data interfaces between the microcontroller and the modem. We promote simplicity in the interfaces, which uses a minimal set of I/O pins. The control interface includes state control, transmission power control and a wakeup signal. The data interface includes digital data output, both digital and analog data input, and the received signal strength (RSSI) measurement. Our initial prototype directly connects the modem to a Mica2 mote through an interface card. The Mica2 mote has an Atmel Atmega128L microcontroller running at 7.3MHz.

We provide individual power control over the transmitter, the wakeup receiver and the data receiver. Each of them can be powered on or off, which forms 7 valid states, as listed in Table 1. These states are controlled by three output pins, each specifying receive/transmit mode, data channel powered on/off, and wakeup channel powered on/off. Data and wakeup tones cannot be transmitted at the same time, because they share the same transmitter. For simplicity we currently support four discrete power levels for transmission from 15dBm to 33dBm at a 6dBm step.

The wakeup receiver is dedicated to detecting incoming wakeup tones. The microcontroller can be put into sleep while leaving the wakeup receiver on. When a tone is detected the wakeup receiver will generate an interrupt, activating the microcontroller, which then will turn on the data receiver. The wakeup receiver and the data receiver can be turned on at the same time. However, in our current design, the wakeup channel and data channel are not independent. When there are signals in both channels, they may interfere with each other.

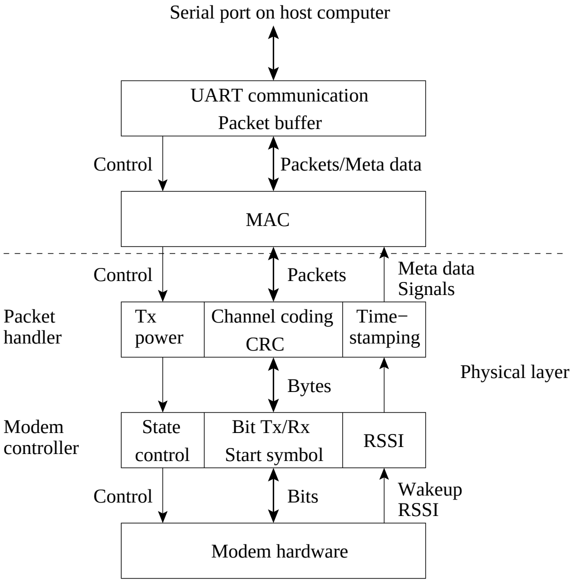

Figure 3: Communication stack on the microcontroller and major components in the physical layer to support packet-level communication.

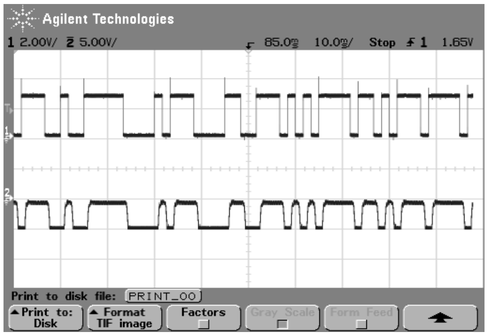

Figure 4: The transmitted waveform (upper) and the received waveform (lower) during BER tests.

We next describe the interfaces for data transmission and reception. For transmission, the microcontroller connects to the modem through a digital output pin. To send bits to the modem, the microcontroller uses an internal hardware timer to interrupt the CPU at the specified baud rate of the modem. Each time when the timer fires, the microcontroller pushes one bit to the output pin. Similarly, on reception, the microcontroller clocks in bits from a digital input pin, which is also driven by the same timer. The RSSI value is measured through an A/D converter.

4.2 Data Communication

Based on the interface defined above, we have implemented major components in TinyOS on the microcontroller to support packet-level communication and upper-layer protocols. This implementation largely borrows the design and source code from our prior work on the Mica mote communication stack [22]. The major principle in our design is to promote clear layering and modularity. For example, the physical layer (PHY) is MAC-neutral, which is able to support different types of MAC protocols, such as CSMA, TDMA or schedule-based MACs [15].

Figure 3 shows the communication stack on the microcontroller and the major software components to support packet-level communication. The communication stack includes the physical layer, MAC, packet buffer, and UART communication. We expect that higher layers will be implemented on the host computer. The host computer communicates with the microcontroller through its serial port managed by the universal asynchronous receiver-transmitter (UART). When sending a packet, the host first passes it to the microcontroller's buffer, which in turn passes it to MAC and then to PHY. PHY has two modules—the packet handler at higher-level and the modem controller below it. The packet handler performs channel coding and passes each encoded byte to the modem controller, which in turn passes each bit to the modem hardware. We are currently using the Manchester code, which is a simple, robust and DC-balanced code. Other coding schemes, such as error-correction codes, can be easily implemented and used in the stack.

On the receiver side, when the data receiver is turned on, it samples the incoming digital signal at a speed of twice the baud rate of the modem. When doing so, it keeps searching for preamble and the start symbol of a potential packet, which are specially designed to minimize false detection and provide good synchronization. When the start symbol is detected, the modem controller will reduce the sampling rate to be the same as the baud rate, and try to align its sampling time to the middle of each data symbol. At this moment, the packet handler is prepared to receive the incoming packet and take readings on the RSSI and timestamp.

Besides the data path, the communication stack has a control path and a signal/meta-data path. The control path goes from an upper layer to lower layers, which can be used to set the transmission power and the modem state, etc. The signal and meta data path goes from a lower layer to upper layers. For example, PHY generates signals such as "wakeup tone is detected" or "start symbol is detected". Meta data such as RSSI and timestamps are collected for all received packets. RSSI values can be used by the MAC layer in its carrier sense algorithm. The timestamps can be used for distributed time synchronization among nodes [21].

5. PRELIMINARY TESTS

Testing of our low power modem is ongoing. So far we have only transmitted and received bits over the air. Underwater testing is planned in the near future. We have performed closed loop bit error rate tests (BER) when the transmitter and the receiver are close to each other in our lab. This test was done using a pair of HP1645A Data Error Analyzers. The bit error rate we have obtained was better than 10^{-5}, based on continuous data streams. We are in the process of interfacing the modem to the Mica2 mote for packet-level testing. Figure 4 shows the transmitted and received waveforms during BER tests.

We have also measured the RSSI output on the data receiver and the sensitivity of the wakeup receiver. The RSSI output from the Data Receiver is roughly linear over a 50 dB range. The wakeup receiver sensitivity is less than 10 μV.

6. RELATED WORK

Underwater acoustic communication has been investigated by many researchers, such as [13, 4, 19, 20]. The major focus in this area is the transmission range, bandwidth utilization and reliability in dealing with multi-path propagation. There are also experimental and commercial off-the-shelf (COTS) acoustic modems available today, such as [14, 2, 10]. However, they are designed for long range communication (1–90km), and are usually bulky and expensive. Long range communication typically requires much higher power than short-range, multi-hop communications [12]. In our modem design, we focus on short-range communication using very low power. This capability is an enabling factor for long-lived sensor networks over battery-powered nodes. Another difference of our work to available COTS modems is that we emphasize low-cost hardware in a small package, enabling large-scale and dense sensor deployment. Densely distributed sensors have the potential advantage of close observation (better signal-to-noise ratio) and multi-sensor fusion (high-fidelity detection and estimation).

Sozer and Stojanovic have developed a reconfigurable acoustic modem (rModem) [18]. Due to its flexible structure, rModem provides a powerful research environment for rapidly testing different communication algorithms, especially at the physical layer. rModem is based on a digital signal processor (DSP) from Texas Instruments, which runs at 225MHz with floating point computing. Combining with Matlab tools, rModem is able to quickly turn simulation code into real-time code and run it on the DSP. Compared to our work, rModem is much more advanced for physical layer research. However, it is not designed for low-power operation, and thus not suitable for long-term deployment over batteries.

Freitag, *et al*, have developed a compact and low-power acoustic modem, called Micro-modem [6]. The Micro-modem is based on a Texas Instruments fixed-point DSP, consuming about 180mW when fully active. The transmission power is fixed and built into the hardware for each application. A unique feature of the Micro-mode is that it has two operating modes: 1) low-power, low-rate and non-coherent (frequency hopping-FSK) and 2) high-power, high-rate and coherent (phase-shift keying). The Micro-modem also incorporates basic acoustic navigation functionality. In comparison, our modem has a much simpler design, and only targets for short range communication and low power operation. For example, our microcontroller only consumes 25mW when fully active, and our transmission power is software adjustable with a maximum value of 2W. When the modem is in sleep mode, only the wakeup receiver is turned on, which consumes about 500μW.

The idea of low-power, short-range, and dense sensor networks comes from the radio-based sensor network research, where several such platforms have been developed. The widely used UC Berkeley motes [9, 5] are based on 8-bit microcontrollers and short-range radios. 32-bit platforms are normally embedded PCs, such as PC/104s (x86 processors) [1] and Stargates (Xscale processors) [5]. Although the radio propagation in water is very bad, the motes are still used by researchers in marine microorganism monitoring applications [24]. Our modem has a similar architecture to motes, but it replaces the radio with our acoustic communication hardware. In fact, our current prototype directly uses the Mica2 mote, which runs TinyOS. Our modem has similar power consumption as Berkeley motes when the lowest transmission power is used. We expect that users of our modem can leverage prior networking research over motes and TinyOS.

7. CONCLUSIONS

This paper describes our work on designing and developing a low-power acoustic modem for underwater sensor networks. The rationale behind our design is to support large-scale, long-lived, and dense sensor networks powered by batteries. We have presented details of our hardware and software design with some preliminary test results. However, the entire work is still in progress. Especially, our current work only uses transducers for in-air communication. We plan to test our modem with real underwater communication in the near future.

8. REFERENCES

[1] `http://www.pc104.org/`.

[2] Benthos, Inc. *Fast And Reliable Access To Undersea Data.* `http://www.benthos.com/pdf/Modems/ModemBrochure.pdf`.

[3] William S Burdic. *Underwater Acoustic System Analysis.* Prentice Hall, 1984.

[4] J. Catipovic, D. Brady, and S. Etchemendy. Development of underwater acoustic modems and networks. *Oceanography*, 6:112–119, March 1993.

[5] Crossbow Technologies, Inc. `http://www.xbow.com/`.

[6] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Balldf. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *Proceedings of the IEEE/MTS OCEANS Conference*, Washington DC, USA, September 2005.

[7] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesc language: A holistic approach to networked embedded systems. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation PLDI*, San Diego, CA, June 2003.

[8] John Heidemann, Wei Ye, Jack Wills, Affan Syed, and Yuan Li. Research challenges and applications for underwater sensor networking. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, page to appear, Las Vegas, Nevada, USA, April 2006.

[9] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, Cambridge, MA, USA, November 2000.

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

[11] D.G. Pincock and F.A. Voegeli. *Quick Course in Underwater Telemetry Systems.* Vemco Inc., `http://www.vemco.com`, January 2002.

[12] Gregory J. Pottie and William J. Kaiser. Embedding the internet: wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[13] A. Quazi and W. Konrad. Underwater acoustic communications. *IEEE Communications Magazine*, 20(2):24–30, March 1982.

[14] Joseph A. Rice. Undersea networked acoustic communication and navigation for autonomous mine-countermeasure systems. In *Proceedings of the 5th International Symposium on Technology and the Mine Problem*, Monterey, CA, April 2002.

[15] Volkan Rodoplu and Min Kyoung Park. An energy-efficient mac protocol for underwater wireless acoustic networks. In *Proceedings of the IEEE/MTS OCEANS Conference*, Washington DC, USA, September 2005.

[16] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Optimizing sensor networks in the energy-latency-density space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, 2002.

[17] Nello Sevastopoulos. *Using the LTC6900 Low Power SOT-23 Oscillator as a VCO*. Linear Technology Co., `http://www.linear.com`, 2002.

[18] Ethem M. Sozer and Milica Stojanovic. Time synchronization for high latency acoustic networks. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Lee, New Hampshire, USA, August 2005.

[19] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21:125–136, April 1996.

[20] M. Stojanovic, J. G. Proakis, J. A. Rice, and M. D. Green. Spread-spectrum methods for underwater acoustic. In *Proceedings of the IEEE OCEANS Conference*, pages 650–654, Nice France, September 1998.

[21] Affan Syed and John Heidemann. Time synchronization for high latency acoustic networks. In *Proceedings of the IEEE Infocom*, Barcelona, Spain, April 2006.

[22] Wei Ye, John Heidemann, and Deborah Estrin. A flexible and reliable radio communication stack on motes. Technical Report ISI-TR-565, USC Information Sciences Institute, September 2002.

[23] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *ACM/IEEE Transactions on Networking*, 12(3):493–506, June 2004.

[24] Bin Zhang, Gaurav S. Sukhatme, and Aristides A. Requicha. Adaptive sampling for marine microorganism monitoring. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 2004.