Research Article

Proteus II: Design and Evaluation of an Integrated Power-Efficient Underwater Sensor Node

Wouter A. P. van Kleunen, Niels A. Moseley, Paul J. M. Havinga, and Nirvana Meratnia

University of Twente, 7500 AE Enschede, Netherlands

Correspondence should be addressed to Wouter A. P. van Kleunen; w.a.p.vankleunen@utwente.nl

We describe the design and evaluation of an integrated low-cost underwater sensor node designed for reconfigurability, allowing continuous operation on a relatively small rechargeable battery for one month. The node uses a host CPU for the network protocols and processing sensor data and a separate CPU performs signal processing for the ultrasonic acoustic software-defined Modulator/Demodulator (MODEM). A Frequency Shift Keying- (FSK-) based modulation scheme with configurable symbol rates, Hamming error correction, and Time-of-Arrival (ToA) estimation for underwater positioning is implemented. The onboard sensors, an accelerometer and a temperature sensor, can be used to measure basic environmental parameters; additional internal and external sensors are supported through industry-standard interfaces (I2C, SPI, and RS232) and an Analog to Digital Converter (ADC) for analog peripherals. A 433 MHz radio can be used when the node is deployed at the surface. Tests were performed to validate the low-power operation. Moreover the acoustic communication range and performance and ToA capabilities were evaluated. Results show that the node achieves the one-month lifetime, is able to perform communication in highly reflective environments, and performs ToA estimation with an accuracy of about 1-2 meters.

1. Introduction

Underwater Acoustic Sensor Networks (UASNs) can be used in many applications, such as monitoring underwater pipelines, monitoring underwater drilling, and performing environmental monitoring [1]. Traditional approaches of underwater measuring involve deploying underwater sensors with data logging capabilities and retrieving the sensors after recording several months of data. Such approach of monitoring provides no real-time monitoring, no reconfiguration, and no failure detection capabilities. By providing communication capabilities to the sensors, it is possible to perform online monitoring and reconfiguration and realize underwater networks of nodes.

Figure 1 gives an example of how an underwater network can look like. Different communication links are used; for example, in the underwater clusters, acoustic communication can be used to communicate data from different sensors to a cluster-head. The cluster-head may then use a long-range link to forward the (possibly aggregated) data towards a surface station. A wireless RF link will forward the data to the shore,

or a satellite link may be used to communicate to the shore. Next to static sensors, Autonomous Underwater Vehicles (AUVs) may be used in the network to provide dynamic links.

One of the first realizations of UASNs is the Seaweb program [2]. This program has performed several experiment deployments with networks of underwater sensors and has performed experiments with modulation, communication, time synchronization, and positioning.

For our research purposes we require an underwater sensor node we can reconfigure to perform experiments with physical, Medium Access Control (MAC), networking, and underwater monitoring applications. In addition, we require the node to be flexible in terms of the internal and external sensors that can be connected. To minimize network maintenance, maximizing battery lifetime is crucial. Our aim is to have a node lifetime of more than one month in low transmit duty cycle scenarios. Existing available commercial underwater sensor nodes do not fit these requirements.

In this paper we describe our design of an underwater acoustic sensor node which combines sensing, processing, and communication powered by an integrated rechargeable
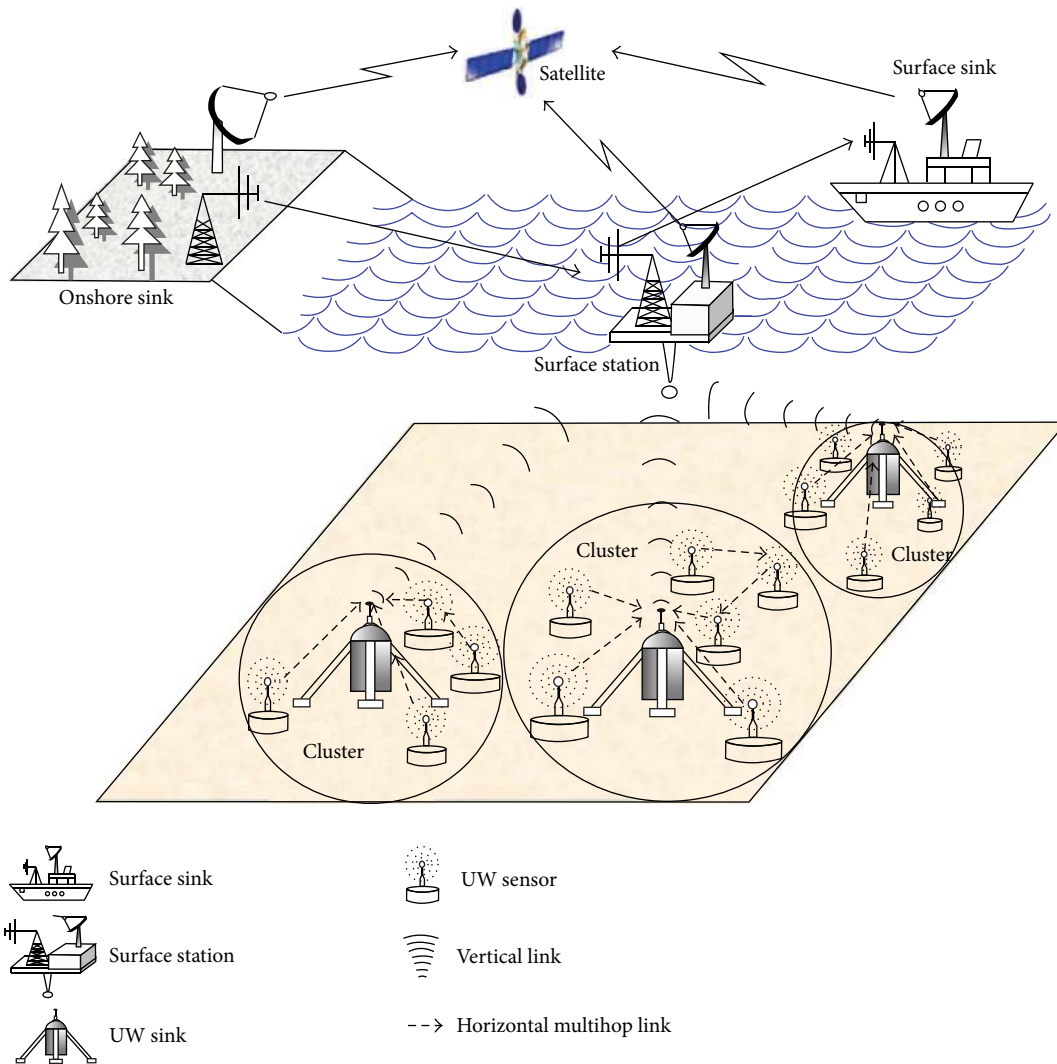
FIGURE 1: Example of an underwater network [1]. Different terrestrial communication links are used in the network, acoustic links between sensors underwater, and RF links between buoys and onshore stations.

battery. It is organized as follows. In Section 2 we discuss related work and existing MODEMs used for underwater sensing. In Section 3 we discuss the design of our integrated underwater sensor node. We discuss the functionality provided on the board, the software design of the MODEM, more specifically the modulation/demodulation of the FSK signal, and the ToA estimation approach. Moreover we discuss the software functionality of the host processor and the networking protocols used in the experiment setup. In Section 4 the results of the performance evaluation of the node are presented. We have measured the power consumption of the node in different operation mode, performed battery lifetime tests, and performed initial communication and ToA estimation tests.

2. Related Work

There are several commercial underwater acoustic MODEMs solutions available, for example, the Kongsberg Mini node [3]

and the MODEMs of Evologics [4]. The WHOI MicroModem [5] is a MODEM designed for navigation and communication with AUVs. These MODEMs implement functionality for underwater communication and in most cases an external CPU is required to perform networking and driving sensors.

The Kongsberg Mini node [3] uses Direct-Sequence Spread Spectrum (DSSS) and Quadrature Phase Shift Keying (QPSK) modulation schemes. The Evologics MODEMs use their proprietary sweep-spread carrier system [6]. The WHOI MicroModem is based on Frequency-Hopping Frequency Shift Keying (FH-FSK) and uses an optional floating point coprocessor to implement Phase Shift Keying (PSK) modulation for data rates from 300 to 5000 bps. These modulation schemes allow for higher symbol rates compared to FSK but require more computational power and therefore require more expensive and power consuming processors.

The underwater node described in [7] has features similar to our proposed design; it uses a two-processor architecture. The node uses a low-power controller CPU for system

management and a more powerful Digital Signal Processor (DSP) for the MODEM. The whole system can be placed in a low-power sleep mode when no acoustic communication is performed. In receive mode, the node requires more than 600 mW to operate, which is too high for our purposes.

Many of the existing platforms were not designed for low-power operation. Moreover, commercial platforms are closed designs and the MODEMs software cannot be adapted to perform experiments.

The MODEMs outlined above do not provide the network layer, the sensor integration, or the required data processing capabilities to deploy complete UASNs. For the network layer, a networking library could be used.

An example of an external networking library which can drive various MODEMs using a separate CPU is the Sunset framework [8]. This framework allows using the same code base for both simulation and deployment, easing the development of and the experimentation with the network layer. This flexibility comes at a cost however; the framework requires the Linux operating system, which runs on high-end power consuming CPU. For example, their recommended Gumstix [9] platform draws between 0.5 W and up to 1 W even when completely powered down. Simply running such a system in standby mode for a month would require a minimum battery capacity of 216 Ah at 3.3 V, which is far from practical.

Underwater sensor nodes that fit our (research) requirements are simply not available. We therefore designed our own low-power extensible nodes from the ground up.

In [10] we briefly described the design of our first generation low-cost underwater node, the SeaSTAR node, based on a single ARM Cortex-M3 processor. The MODEM implemented FSK modulation and ToA estimation. The node's power consumption was still quite high because a development board was used for the Cortex-M3. A 433 MHz surface radio interface was used for collecting measurements when the node was deployed at the surface. This proved to be very useful for performing experiments and our new design also incorporates a surface radio.

3. Design of the Underwater Sensor Node

The following section describes the hardware and software design of the Proteus II sensor node. First we will discuss the hardware components on the board and their functionality. The Proteus II sensor uses two processors, one for the MODEM functionality and one for the MAC and networking protocols. After discussing the hardware design, we discuss the software design of the HOST and the MODEM processor. We also discuss the physical modulation and packet format of the acoustic communication and the ToA estimation approach used for time synchronization and positioning.

The Proteus II sensor node (see Figure 2) is built around two low-cost low-power 32-bit ARM Cortex-M4 processors. The host CPU is responsible for the network protocols and processing the data of onboard and optional external sensors. The second CPU is dedicated to the acoustic MODEM functionality, which is implemented following
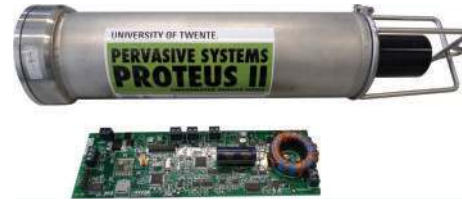

FIGURE 2: A picture of the Proteus board and enclosure. The board and battery (not shown) are placed in the enclosure to make an integrated underwater sensor node. Connectors on the lid of the enclosure allow external sensors to be connected. The lid is also fitted with an RF connector for the 433 MHz radio antenna. The node can be completely submerged; however no 433 MHz radio connection will be available when doing so. At the bottom end of the enclosure, the (black) acoustic transducer is visible and the transducer operates in the frequency range of 20–40 kHz.
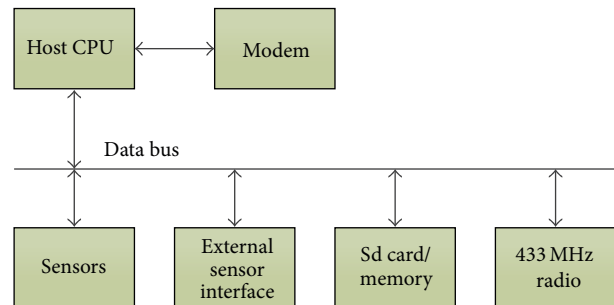

FIGURE 3: A block diagram showing the architecture of the Proteus II node. The host CPU can transmit information to other nodes via the MODEM. Various subsystems, such as the sensors, SD card, and the 433 MHz radio, are interfaced to the host processor via the internal data bus.

the software-defined radio paradigm [11]. The MODEM also provides ToA estimation for time synchronization and positioning.

The MODEM needs to perform CPU-intensive decoding of the acoustic signal, requiring the processor to run in maximum performance mode. The host CPU will mostly be dormant and run software timers to control the sensing and transmission of data. This processor can therefore be run in the very low-power run mode available on the CPU.

The onboard sensors, which consist of an accelerometer and a temperature sensor, can be used to measure basic environmental parameters such as surface wave characteristics, node orientation, and water temperature. The node supports additional internal or external sensors through its industry-standard digital serial interfaces (I2C, SPI, and RS232) and can interface analog peripherals through a 16-bit ADC. A 433 MHz radio provides an additional way of communication when the nodes are deployed on the water surface.

Figure 3 shows the architectural block diagram of the Proteus II node.

3.1. Hardware Architecture. Figure 4 shows the components available on the Proteus II board. The board can be powered
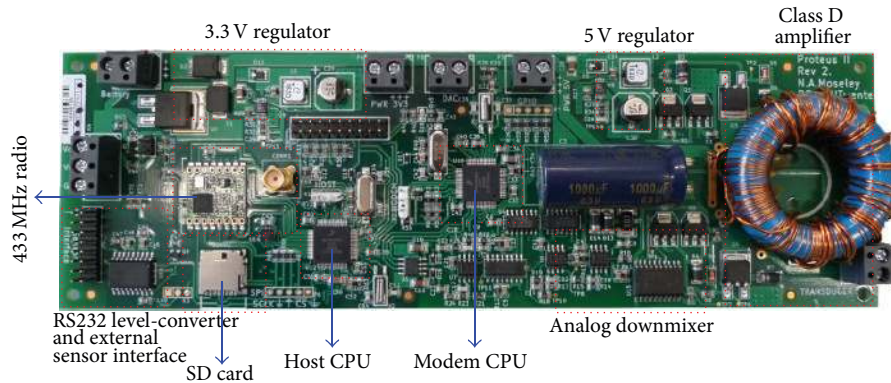
FIGURE 4: A photograph of the Proteus II PCB, showing the location of various subsystems, such as the CPUs, the receive downmixer, the amplifier, voltage regulators, SD card, and the 433 MHz radio.

from a battery or an external power in the range of 12 V–24 V. The input voltage is converted to 3.3 V and 5 V by two switching regulators. The regulators were selected for high energy efficiency even during the ultra low-power (sleep) operation. The 3.3 V regulator powers all the digital circuitry while the 5 V regulator powers the analog part of the MODEM. When the MODEM is not operating, the 5 V regulator is turned off to save power. To further save power, the host processor is placed in low-power mode when the node is sleeping.

In receive mode, the signal from the acoustic transducer is amplified, filtered, and downconverted to baseband, resulting in a quadrature signal. A 24-bit stereo sigma-delta is used to convert the received signal into the digital domain where it is further processed by the MODEM processor. The main benefit of using a sigma-delta converter is the much relaxed requirements on the antialiasing filters. This reduces the design complexity and saves board space. Both the downconverter center frequency and the sample rate of the ADC are under software control.

In transmit mode, an FSK signal is generated by toggling a general-purpose output pin through a timer on the MODEM processor. The output pin drives the class D amplifier and acoustic transducer. The timer, which is under software control, directly determines the frequency of the FSK signal.

The Proteus II node has several sensors and sensor interfaces on board. The onboard sensors are an accelerometer (ADXL362) and a temperature sensor. Additional analog and digital sensors are supported through the onboard 16-bit ADC and the serial interfaces (I2C, SPI, and RS232-level UART), respectively. For data logging, the node can be fitted with an SD card. Finally, the node features a 433 MHz radio for RF communication. This, however, only works when the node is deployed at the surface.

Figure 2 illustrates the Proteus II board and the designed enclosure. Rechargeable Lithium-Polymer (LiPo) battery packs of up to 5000 mAh@12 V fit into the enclosure to power the board.

With the integrated sensors, data processing, and communication capabilities the node is an integrated flexible sensor platform.

3.2. Software Architecture: Host Processor.

The host processor has many duties. It processes the data from the sensors, runs the network and routing layer software, drives the MODEM, and controls the 433 MHz radio. The 433 MHz radio is used for network debugging, configuring the sensors, and performing time synchronization when the sensor has been deployed at the water surface. In addition, the host processor is responsible for the power management of the node.

Because the host processor is an embedded processor, with limited memory (64 kb) available and no Memory Protection Unit (MPU), an operating system such as Linux will not run on this processor. It is our believe that small, energy optimized operating systems such as FreeRTOS [12] or TinyOS [13] are more suited for energy-efficient sensors.

Inspired by these small (real-time) operating systems, we have implemented the Proteus II operating system. This runs on both the host and the MODEM processor. The operating systems consist of a low-power time scheduling framework. Different periodic tasks can be scheduled and the operating systems control that when idling, the host processor goes into a low-power sleeping state with wakeup timers running. Moreover the operating system is able to handle events, such as incoming SPI or UART communication or a radio packet arriving from the 433 MHz radio, waking up the processor from low-power mode when such event occurs.

Using this operating system, the different processors can be placed in sleep mode. The MODEM processor and associated MODEM electronics are put into sleep mode when no communication is required. This is controlled by the host processor through the SPI interface. The host processor is able to run on a wakeup timer and shut down the MODEM. In this mode, the overall power consumption of the node is reduced to less than 8 mW.

In our test network we have implemented a time-division multiple access (TDMA) network protocol designed for performing experiments. Because we wanted to evaluate the Packet Delivery Rate (PDR) achievable by the node within our test environment, we used TDMA protocol to avoid collisions. The networking protocol uses both the RF radio and the underwater MODEM to perform communication.
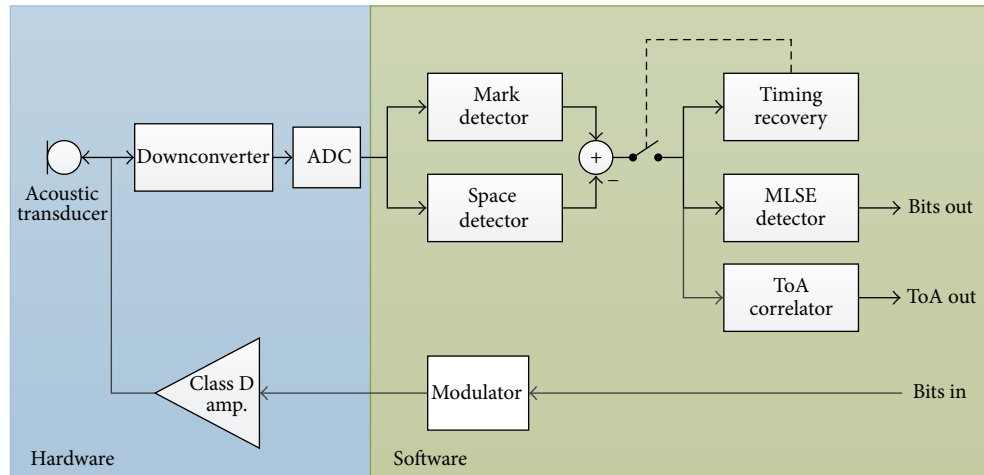
FIGURE 5: A block diagram of the hardware and software partitioning of the MODEM physical layer. In the receive path, the acoustic transducer signal is sampled and downconverted to baseband. Two tone detectors detect the energy at the Mark (1) and Space (0) frequencies. A maximum-likelihood sequence estimator (MSLE) performs the symbol detection and timing-recovery and then it outputs the received bits. In the transmit path, a modulator converts the incoming bits into symbols. The symbols are amplified and transmitted by the acoustic transducer.

Time is divided in 4-second timeslots and nodes in the network are assigned a static timeslot before deploying the network. The gateway uses the RF radio to poll the different sensors in their assigned timeslots. In the poll message to the sensor the following data is included.

(i) Command. It indicates what type of message should be transmitted by the sensor: a Time-of-Flight (ToF) beacon, a sensor measurement or nothing should be transmitted at all. During our tests we transmit 3 ToF measurements and every fourth packet is a measurement packet which includes measurements from the accelerometer and temperature sensor.

(ii) Baud Rate. It indicates what baud rate should be used for the acoustic transmission when sending the acoustic packet.

(iii) Timestamp. It indicates the current time of the gateway. A ToF beacon message includes this timestamp, such that when the packet arrives at the gateway, the round-trip time can be calculate.

The poll message indicates what type of packet should be transmitted by the sensor and what settings should be used. Upon a poll message, the sensor will reply with a statistics message using the RF radio.

(iv) Battery Power. It indicates the battery power voltage reading.

This setup allows us to configure the behaviour of the sensors and provides debugging information. It allows us to centrally control the network, experiment with baud rates, and perform ToF and PDR measurements. In future work we would like to extend this with configuration of the packet type (BCH or Reed-Solomon with configured error correction overhead) and with routing information (indicating which sensors should send to which sensor).

The described networking protocol is geared towards a test network, for experimenting with physical, MAC networking, routing, and positioning algorithms, and the intended usage of the network.

Because we use a TDMA approach, the sensor is able to sleep between the polling cycles, thereby reducing the power consumption of the sensor. In other projects the host processor can be reconfigured for supporting different sensors, performing networking algorithms, and positioning or implementing network security using data encryption and validation.

3.3. Software Architecture: MODEM Processor. Following the software-radio paradigm, the majority of the acoustic transmit and receive functions are implemented in software and run on the MODEM processor. Figure 5 shows the hardware/software partitioning of the physical layer.

The Proteus II uses FSK modulation with fixed frequencies around 21 kHz. FSK is chosen for its low computational complexity, its relatively good multipath resilience, and its constant envelope property, which greatly simplifies the power amplifier design. Communication speed of the MODEM is slow (10 to 200 baud), but the speed can be improved in the future by using multitone FSK.

In the receive mode, the quadrature baseband signal from the ADC is processed by two tone detectors, which operate at 8 ksps sampling rate. Their outputs are resampled at 16 times the baud rate to reduce the computational complexity of the following processing blocks. A maximum-likelihood sequence estimator is used to determine the most likely 4-symbol sequence of the most recently received data. Then bits are extracted and if necessary, the symbol timing is adjusted. The resulting bit-stream is fed into the packet decoder. See Figure 5 for an overview of the physical layer processing.

Sync	Preamble	Len	Payload	CRC

Figure 6: Proteus II packet format: the packet consists of sync bits for time synchronization, a preamble with low autocorrelation properties to allow for ToA estimation, and packet start detection. The payload of the packet is modulated using the Hamming error correcting approach. The CRC is used for validation of the packet data upon reception.

The packet format used by the Proteus II node is defined as follows. The packet consists of several sync bits (10101010), which allows the bit synchronization to be performed before the actual data of the packet data is received; see Figure 6. It is followed by a Barker sequence preamble [14]. The Barker code has low autocorrelation properties allowing accurate packet arrival detection and ToA estimation. Bit synchronization and ToA estimation are further improved by using data-whitening on the packet payload. Error protection is provided through a low-complexity Hamming code based Forward Error Correction (FEC) scheme and a Cyclic Redundancy Check (CRC).

After the sync bits and Barker code preamble, the length of the payload (including its CRC) is transmitted. The 16-bit CRC is added at the end of the packet to validate the packet upon reception. A CRC polynomial was selected which has good bit-error detecting properties [15]. Every 32-bit block of the payload and CRC is extended with a 6-bit parity block for storing the parity bits produced by the Hamming error correction code. This is shown in Figure 7.

Data-whitening is applied to the len + payload + CRC bits of the packet. This is done to reduce the autocorrelation properties of the packet payload, thereby improving the ToA estimation performance, and to increase the number of Mark to Space transitions required by the demodulator to perform bit synchronization.

3.4. ToA Estimation. The MODEM performs ToA for supporting ToF or Time-Difference-of-Arrival (TDoA) localization. The ToA can be determined by cross-correlating the transmitted signal and the received signal which is buffered by the MODEM. The sampling rate of the signal is dependent on the configured symbol rate of the MODEM. The energy-detector of the MODEM samples the signal at 16x the symbol rate. Therefore when the symbol rate of the MODEM is increased, the sampling rate is increased and the ToA estimation is expected to be more accurate.

Once a packet is correctly received, this can be determined by verifying the CRC, the fixed preamble, and a part of the decoded packet is used to perform the cross-correlation. The Barker sequence in the header and the whitened data of the packet improve the cross-correlation capabilities of the packet.

After cross-correlation is performed, the correlation shows the signal response of the acoustic channel. This response may contain several multipath arrivals of the signal. The strongest arrival may not necessarily be the first or line-of-sight arrival. Therefore the cross-correlation needs to be searched to estimate the first path arrival. In [16] several

Len		Payload		CRC

32-bits	32-bits	32-bits	32-bits

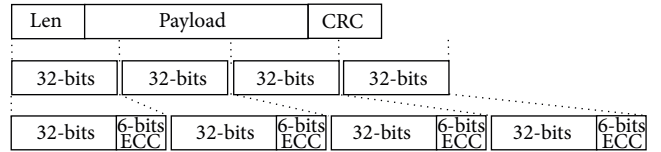| 32-bits | 6-bits ECC | 32-bits | 6-bits ECC | 32-bits | 6-bits ECC | 32-bits | 6-bits ECC |

Figure 7: Hamming based error correction is performed on subblocks of the payload + CRC. The number of error correction bits for the Hamming code is fixed and error correcting capabilities for the Hamming code cannot be configured.

techniques for estimating ToA of ultrawideband (UWB) signals have been proposed. These algorithms can also be used for estimating the arrival of an acoustic signal. We use the "search and subtract" algorithm [16], which works as follows:

(1) Find the strongest arrival in the cross-correlation.

(2) Set the first arrival threshold to 1/4 of the peak of the strongest arrival.

(3) Subtract the strongest arrival from the cross-correlation result.

(4) Search again for the next strongest arrival that is received before all other strongest arrivals and is stronger than 1/4 of the strongest arrival.

(5) Subtract this arrival and go back to step (4).

The process of finding an earlier arrival in the cross-correlation and attempting to determine the first arrival of the signal rather than the strongest arrival is repeated up to 4 times or is halted earlier if no strong arrival peak can be found. Repeating the algorithm (4 times) represents the maximum number of earlier multipath arrivals we expect to be able to detect.

4. Performance Evaluation

The following section describes the performance evaluation we have performed on the Proteus II node. We have evaluated the power consumption of the node in different state and we make an estimation whether one-month lifetime is achievable. We have performed a lifetime test with a high duty cycle and a low duty cycle; in the low duty cycle test we achieve a lifetime of one month. Next to the lifetime tests, we evaluate the PDR and ToA estimation of the node. Moreover we have evaluated the maximum communication range of the node.

4.1. Power Consumption. We have measured the power consumption of the node during acoustic transmission, acoustic reception, and idle and with the 433 MHz radio enabled. Results are shown in Table 1. Transmission means that an acoustic packet is being transmitted; reception indicates that the MODEM is listening for the reception of an acoustic packet. In the idle state the acoustic MODEM is turned off and the host processor is running a sleep timer to wake up the host periodically and radio that indicates the 433 MHz radio is turned on. As can be seen the acoustic MODEM transmits at about 8 W; during reception the MODEM uses 300 mW.

TABLE 1: Power consumption of the node in different states.

State	Power (W)	Description
Transmission	8 W	Active acoustic transmission
Reception	300 mW	Active packet reception
Idle	60 mW	Node sleeping on wakeup timer
Idle (fix)	8 mW	Node sleeping on wakeup timer after disabling the RS232 level-converter
Radio	200 mW	433 MHz radio active

TABLE 2: Estimation of the battery capacity required for a battery lifetime of one month.

State	Duty cycle	Power (mW)	Total (mWh)
Idle	1	8	5952
Transmission	12/3600	8000	19840
Battery capacity required (mWh)			25792
Battery capacity 5000 mAh at 12 V (mWh)			60000

Initially with the MODEM turned completely off and the host running on a wakeup timer and being in sleep mode the complete node consumed 60 mW. This was higher than what should have been achievable with the design. Later, we discovered that the RS232 level-converter was still in standby mode and after disabling the RS232 level-converter, the idle power consumption of the node dropped to 8 mW. Using this idle power consumption, one month of monitoring can be easily achieved using different communication patterns.

Table 2 gives an indication of the battery capacity required to run the node for one month ($31 \times 24 = 744$ hours). The idle power consumption of the node is always present (duty cycle is 1). For the transmission we assume a duty cycle of 12 seconds for every hour (3600 seconds). The total battery capacity required is about 25792 mWh, which indicates that a battery of 5000 mAh@12 V (60000 Wh) is more than sufficient, and leaves plenty of room for self-discharge of the battery. The calculation indicates that a battery of 5000 mAh is more than sufficient to run the node for one month. Such a battery fits in the enclosure; therefore we can say that we can achieve the required one-month operation; other configurations are also possible. In Section 4.3 we perform lifetime test with different transmission duty cycles.

4.2. Packet Delivery Rate. Figure 8 shows our Packet Delivery Rate (PDR) setup. Four nodes and one gateway were deployed in an artificial lake at the University of Twente campus. The acoustic communication is performed underwater while a radio connection to the gateway is used to collect the measurement data. Acoustic communication was performed in the 20–22 kHz band; the same frequencies were used for the different baud rates. The deployment of the node with the acoustic transducer underwater and the radio antenna above the water surface is shown in Figure 9. The test environment is shown in Figure 10; a lot of concrete is present in the form of walls, pillars, and the lake floor. The communication is therefore highly influenced by many multipaths.
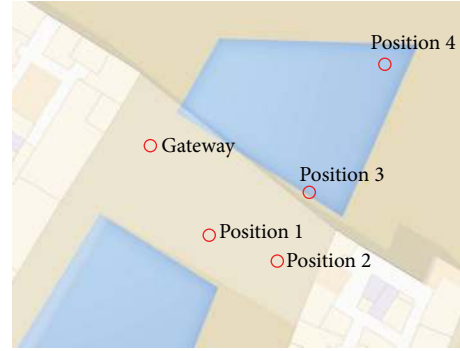

FIGURE 8: Illustration of the PDR setup deployment. The nodes were placed at different distances from the gateway and the water continues under the shown building.

FIGURE 9: A Proteus II node deployed in the water at the test setup. The acoustic part of the node is underwater and the radio antenna is above the water surface to allow a radio connection with the gateway.

All the nodes have a single-hop connecting to the gateway. Packets were transmitted for a period of 22 hours at different symbol rates (50, 100, and 150 baud). The PDR rate was calculated for this setup. Results shown in Figure 11 indicate that about 30–40% PDR is achieved in this environment. When the symbol rate is increased to 150 bps, the delivery rate of two nodes decreases to only a couple of percent. In this case, the multipath causes these nodes to be unable to communicate.

Figure 12 gives an indication of the pattern of packet loss. Shown in the graph are the received packet numbers of the gateway, with their respective receive time. The packet numbers cycle from 0 to 64 before wrapping back to 0 and ideally all should be received. What can be seen is that certain periods of time experience little packet loss, while other periods of time experience a significant packet loss. What causes this effect is still unclear to us; possibly this is the result of the node slightly moving in the water and experiencing different multipath effects for an extensive period.

4.3. Lifetime. In the same setup we have performed two lifetime tests. With a transmission interval of every minute, performing a transmission of 1 second every minute, the nodes were able to communicate for 5 days using a 2200 mAh@12 V

FIGURE 10: Picture of the deployment in the artificial lake at the University of Twente campus. The environment is a small lake between office buildings and consists of concrete walls and floor. The gateway node was placed near an office to allow a network connection to be made to campus network, allowing access to the testbed through internet. The nodes are kept floating with the radio antenna above the water surface, to allow a radio connection to the gateway for collecting measurement data, and are fixed with a sandbag on the floor. The acoustic transducer is deployed underwater and acoustic communication is performed through the water.

TABLE 3: Estimation of the battery capacity required for the lifetime test over a duration of 5 days.

State	Duty cycle	Power (mW)	Total (mWh)
Idle	1	60	7200
Transmission	60/3600	8000	16000
Battery capacity required (mWh)			23200
Battery capacity 2200 mAh at 12 V (mWh)			26400

FIGURE 12: Arrival of packets at the gateway over the period of 10 hours. The received packet numbers at the time they are received are shown; packet numbers cycle from 0 to 64. What can be seen is that certain periods of time show little packet loss, while other periods of time experience significant packet loss.

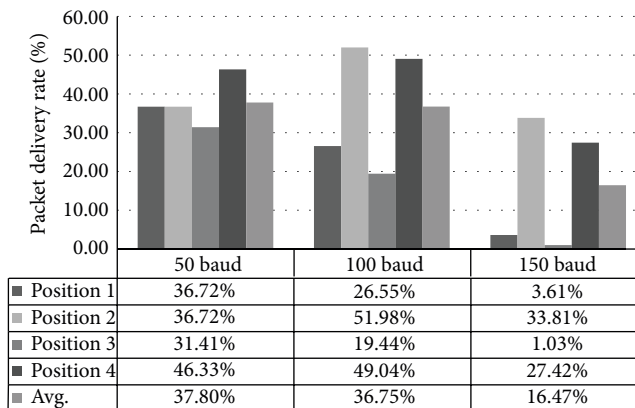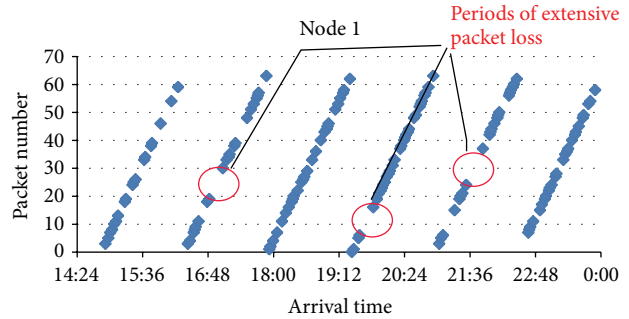|	50 baud	100 baud	150 baud
■ Position 1	36.72%	26.55%	3.61%
■ Position 2	36.72%	51.98%	33.81%
■ Position 3	31.41%	19.44%	1.03%
■ Position 4	46.33%	49.04%	27.42%
■ Avg.	37.80%	36.75%	16.47%

FIGURE 11: Results of the PDR test shown are the percentage of packets that were correctly delivered to the gateway over a period of 22 hours for different symbol rates.

rechargeable battery. In Table 3 we show the calculation of the required energy for these 5 days. For the transmission the duty cycle was 60 seconds for every hour (3600 seconds). The energy required was 23200 mWh; this is in line with the capacity of the battery we used (2200 mAh@12 V = 26400 mWh). The calculated consumed energy is in line with the battery capacity that we calculate.

In another test we have equipped the node with a 5000 mAh@12 V battery. With a low duty cycle, a transmission of 2 seconds every 10 minutes, we have achieved a one-month lifetime of the underwater node. This shows that our set goal, a lifetime of one month using a low duty cycle, is achievable.

4.4. Time of Arrival. In a different setup, ToA estimation was evaluated. Using two nodes we estimate the range between two nodes by using the radio and an acoustic transmission. The master node transmits a radio packet to the slave node to start a transmission. The slave node replies with an acoustic packet and the master node determines the round-trip time of the radio and acoustic transmission.

The distance between the two nodes was changed from the two nodes deployed completely next to each other (0-meter distance) to 10, 20, and 30 meters (setup shown in Figure 13). The distance of 0 meters is used to determine the processing delay of the measurement. The results can be seen in Figure 14.

Our results show that ToA of the MODEM is able to estimate the distance between the nodes in the experiment. As expected, when the symbol rate is increased the accuracy of ToA measurement and consequently distance measurements also increases. At the distance of 30 meters, the MODEM appears to lock on to a multipath arrival rather than the direct line-of-sight. A relative distance of about 40 meters is measured, rather than the expected 30 meters. This result is the same for all symbol rates.

ToA estimation based on modulated signals remains difficult and is affected by multipath. Even with the "search and substract" algorithm described in Section 3.4, the MODEM still locks on the multipath arrivals in some cases. ToA estimation based on wideband chirps is definitely more accurate. Experiments with a chirp signal in highly reflective environments were performed in [10, 17]. The performance of the ToA estimation can be increased using multitone FSK and therefore using a wider band for the signal.
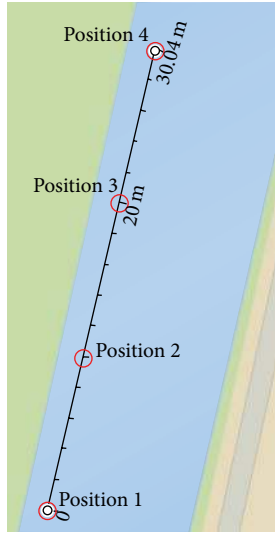
FIGURE 13: Illustration of the ToF setup deployment. The ToF was measured at four different positions ranging in distance from 0 m to 30 m. How well the distance between the different positions can be estimated using the ToA measurement was evaluated.

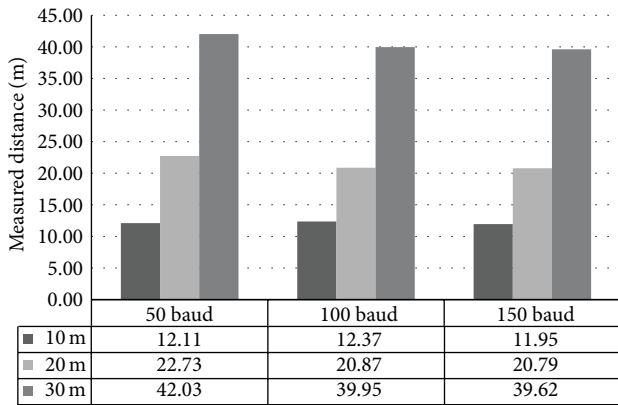|	50 baud	100 baud	150 baud
■ 10 m	12.11	12.37	11.95
■ 20 m	22.73	20.87	20.79
■ 30 m	42.03	39.95	39.62

FIGURE 14: Measured distance between Proteus II nodes at different symbol rates and at different distances.

In the same environment as the ToA setup, maximum communication range tests indicate that the node is able to communicate over a distance of 140 m. Because the matching of the transducer is very much frequency dependent, we believe the range can be improved by tuning the frequency of the communication. For our current setup the FSK were fixed in the band of 20–22 kHz.

5. Conclusion

For our future research we require an underwater sensor node which can be reconfigured at all networks layers, the physical layer up to the transport layer. Moreover, we require a node which can be extended with different sensors. The underwater node should be able to run on a rechargeable

battery for one month and, ideally, the node should also be of low cost.

Because existing commercial nodes do not allow reconfiguration of the physical layer protocols in the MODEM and existing research platforms were not available to us or did not meet our energy efficiency criteria, we have designed our own underwater sensor node called Proteus II.

This node has an energy-efficient power circuit and is designed for low-power operation; a lithium battery inside the enclosure powers the sensor node. The node has integrated sensors such as an accelerometer and temperature sensor and ADC allows connecting analog sensors. External sensors can be connected through connectors available on the enclosure.

A 433 MHz radio is available on the node to allow a radio connection for debugging and collecting measurement data when the node is deployed at the water surface. In our test a TDMA MAC protocol was implemented to prevent collisions. Sensors are polled by the gateway using the RF radio; this allows us to control, configure, and diagnose the network centrally.

The node consists of a low-power MODEM and host controller both based on a Cortex-M4 processor. The MODEM is designed as a software-defined MODEM and uses FSK modulation with Hamming error correction and ToA estimation for positioning and time synchronization. Bit decoding and bit synchronization are performed using maximum-likelihood decoding and ToA estimation is performed using cross-correlation with "search and substract" algorithm for estimation of first arrival and multipath arrivals. The host processor is used for performing networking, reading the onboard sensors, and controlling the 433 MHz radio.

Tests were performed to measure the power consumption and performance of the node. In low-power mode, the node draws about 8 mW. In a lifetime test we have achieved the desired one-month lifetime, using a 5000 mAh@12 V battery and a duty cycle of transmitting 2 seconds every 10 minutes.

Initial communication tests in our test environment show that the Proteus II is able to achieve a PDR of 30–40% in an environment which has concrete walls and floors and is therefore highly reflective. Moreover, communication was performed over the distance of 140 m.

ToA estimation was also evaluated and the fact that the MODEM is able to perform ToA estimation with an accuracy of about 1 to 2 meters was shown. The ToA based on the modulated FSK signal is still affected by multipath interference.

In the future, multitone FSK and perhaps PSK will be researched to see if an improvement can be made in communication speed and ToA by using a wider band signal. Multitone FSK likely has a positive effect on the delivery rate, because many multipaths are present within the environment. We would also like to evaluate the performance of our nodes in less reflective environments. The maximum communication range of 140 m can also be improved in the future by tuning the frequency to the matching of the transducer. In our experiments the used frequencies were fixed; by allowing configurable frequencies, the use of different frequencies can be evaluated.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[2] J. Rice, B. Creber, C. Fletcher et al., "Evolution of seaweb underwater acoustic networking," in *Proceedings of the MTS/IEEE Conference and Exhibition (OCEANS '00)*, vol. 3, pp. 2007–2017, IEEE, Providence, RI, USA, September 2000.

[3] T. S. Husøy, F. R. Knudsen, B. Gjelstad, and A. Furdal, "Product development at kongsberg maritime related to underwater sensor networks," in *Proceedings of the 7th ACM International Conference on Underwater Networks and Systems (WUWNet '12)*, pp. 16:1–16:5, ACM, New York, NY, USA, November 2012.

[4] Evologics, http://www.evologics.de/.

[5] 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," in *Proceedings of the MTS/IEEE OCEANS Conference*, vol. 2, pp. 1086–1092, IEEE, Washington, DC , USA, September 2005.

[6] R. Bannasch and K. Kebkal, "Method and devices for transmitting and receiving information," US Patent 6,985,749, 2006.

[7] Y. Yang, Z. Xiaomin, P. Bo, and F. Yujing, "Design of sensor nodes in underwater sensor networks," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 3978–3982, May 2009.

[8] C. Petrioli, R. Petroccia, and D. Spaccini, "SUNSET version 2.0: enhanced framework for simulation, emulation and real-life testing of underwater wireless sensor networks," in *Proceedings of the 8th ACM International Conference on Underwater Networks and Systems (WUWNet '13)*, pp. 43:1–43:8, ACM, Kaohsiung, Taiwan, November 2013.

[9] Gumstix, http://www.gumstix.com/.

[10] W. A. P. van Kleunen, N. A. Moseley, N. Meratnia, and P. J. M. Havinga, "Experiments with aLS-Coop-Loc cooperative combined localization and time-synchronization," in *Proceedings of the 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '14)*, pp. 86–91, IEEE Computer Society, Hammamet, Tunisia, May 2014.

[11] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, 1995.

[12] FreeRTOS, A free open source rtos for small embedded real time systems, http://www.freertos.org/.

[13] P. Levis, S. Madden, J. Polastre et al., "Tinyos: an operating system for sensor networks," in *Ambient Intelligence*, pp. 115–148, Springer, Berlin, Germany, 2005.

[14] S. W. Golomb and R. A. Scholtz, "Generalized barker sequences," *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 533–537, 1965.

[15] P. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 145–154, IEEE, June-July 2004.

[16] C. Falsi, D. Dardari, L. Mucchi, and M. Z. Win, "Time of arrival estimation for UWB localizers in realistic environments," *Eurasip Journal on Applied Signal Processing*, vol. 1, Article ID 032082, 2006.

[17] B. S. Borowski, *Application of channel estimation to underwater, acoustic communication [Ph.D. thesis]*, Stevens Institute of Technology, Hoboken, NJ, USA, 2011, AAI3467242.