The UNET-2 modem – An extensible tool for underwater networking research

Mandar Chitre, Iulian Topor and Teong-Beng Koay
Acoustic Research Laboratory, Tropical Marine Science Institute,
National University of Singapore, Singapore 119227.
{mandar, iulian, koay}@arl.nus.edu.sg

***Abstract*—Several decades of research in underwater communication and networking has resulted in novel and innovative solutions to combat challenges such as long delay spread, rapid channel variation, significant Doppler, high levels of non-Gaussian noise, limited bandwidth and long propagation delays. Many of the physical layer solutions can be tested by transmitting carefully designed signals, recording them after passing through the underwater channel, and then processing them offline using appropriate algorithms. However some solutions requiring online feedback to the transmitter cannot be tested without real-time processing capability in the field. Protocols and algorithms for underwater networking also require real-time communication capability for experimental testing. Although many modems are commercially available, they provide limited flexibility in physical layer signaling and sensing. They also provide limited control over the exact timing of transmission and reception, which can be critical for efficient implementation of some networking protocols with strict time constraints. To aid in our physical and higher layer research, we developed the UNET-2 software-defined modem with flexibility and extensibility as primary design objectives. We present the hardware and software architecture of the modem, focusing on the flexibility and adaptability that it provides researchers with. We describe the network stack that the modem uses, and show how it can also be used as a powerful tool for underwater network simulation. We illustrate the flexibility provided by the modem through a number of practical examples and experiments.**

I. Introduction

With an increasing interest in marine monitoring using robotic and autonomous sensor systems, the ability to communicate effectively underwater is rapidly becoming very important. Although underwater acoustic communications has been explored for several decades, the achievable throughput in an underwater network today is very modest as compared to terrestrial wireless networks. This is largely attributed to challenges such as long delay spread, rapid channel variation, significant Doppler, high levels of non-Gaussian noise, limited bandwidth and long propagation delays. In the past years, researchers have developed novel and innovative solutions to combat many of these challenges [1]. In fact, some of the challenges such as propagation delay offer new opportunities that may be exploited in scheduling transmissions [2]. Researchers' ability to test some of these new ideas and innovations is often limited by the modems available to them for field testing. Although several off-the-shelf modems are commercially available, they provide limited flexibility in the physical layer signaling and sensing. Most of the modems also provide very limited control over the exact timing of transmission and reception, which can be critical for efficient implementation of networking protocols with strict time constraints [2], [3], [4]. To aid in physical and higher layer research at the Acoustic Research Laboratory (ARL), we developed the UNET-2 software-defined modem (shown in Figs. 1 and 2) with flexibility and extensibility as primary design considerations. This modem is now well-tested and extensively being used for communication between sensor nodes, autonomous underwater vehicles (AUVs) and surface stations in several research projects.

In this paper, we present the hardware and software architecture of the modem, focusing on the flexibility and adaptability that it provides researchers with. We describe the network stack that the modem uses, and show how it can be used as a powerful tool for underwater network simulation. We also present some examples and experiments to demonstrate how the modem's flexibility can be applied in different channel conditions and application scenarios.

II. Modem Features

Designed primarily as a research tool, the UNET-2 modem provides a flexible platform for a variety of underwater networks. With substantial computing power packed into a small package, researchers are able to implement and deploy complex algorithms in the modem. The modem provides options for customization and extension at many levels (more on this in section III-D), allowing network protocols as well as physical layer algorithms to be implemented and tested easily.

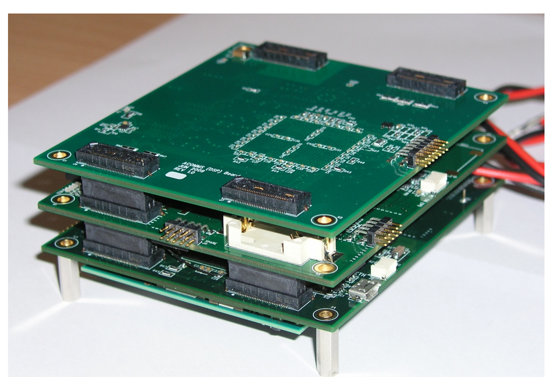

Fig. 1. The UNET-2 modem "dry-end" digital electronics ($10 \times 9 \times 5\frac{1}{2}$ cm).

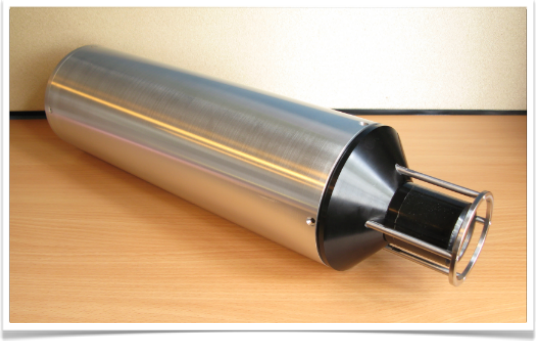

Fig. 2. The UNET-2 modem packaged as a stand-alone surface unit.

The firmware and software in the modem starts up with a configuration that provides robust communication performance, but can be reconfigured on the fly to meet application or protocol demands.

In the next few sections, we outline the flexibility provided by the modem in various aspects of communication and networking:

A. Carrier frequency, bandwidth & source level

The transmit and receive carrier frequencies of the modem can be changed dynamically, up to a maximum carrier frequency of 78 kHz. By independently and dynamically controlling these, we can compensate for mean Doppler at the analog front-end if desired. The digital filters and baseband sampling rate are dynamically adjusted to provide a bandwidth of up to $2f/3$ where f is the carrier frequency. The source level of each physical layer packet transmission can be dynamically controlled in steps of 1 dB.

B. Packet detection & logical channels

Each physical layer packet transmitted by the modem is prefixed by a packet detection preamble and an optional guard time. Up to 3 user-defined preambles may be configured in the modem. When reception is enabled, the modem uses a bank of sign-correlators [5] to monitor the water for all configured preambles and triggers a packet reception when any of the preambles is detected. The preamble can be used to dynamically determine the signaling and forward error correction (FEC) schemes to be used for the packet that follows. This allows multiple *logical channels* to be created. For example, we can configure a *control channel* to use an incoherent signaling scheme with a low-rate FEC for robust communication, and a *data channel* to use coherent modulation with a high-rate FEC for high-speed communication. Link tuning algorithms may dynamically adapt the parameters of the data channel to provide good performance in time-varying channels [6].

C. Collision detection

The preamble detection operates independently of the packet acquisition and processing. This enables the modem to notify the network stack of a collision if a preamble is detected while a packet is being acquired. Medium access control (MAC) protocols which require collision detection can therefore be supported on the modem. To avoid false collision alarms due to a multiple arrivals arising from the delay spread of the channel, the preamble detector has a user-settable blanking time immediately after a detection.

D. Modulation & FEC schemes

The modem is preconfigured with two modulation schemes. The *incoherent OFDM* modulation scheme uses multicarrier differential energy detection signaling to provide Doppler-resilient robust performance in doubly-spread channels. The *coherent OFDM* modulation scheme employs multicarrier phase-coherent differential signaling with a cyclic prefix to provide high data rate communication [7]. Modulation scheme parameters such as number of carriers, prefix length, suffix length, null carriers, peak-to-average power ratio (PAPR) reduction, signal constellation size (binary or quadrature phase-shift-keying, i.e., BPSK or QPSK), differential mode (time or frequency), packet length, etc are dynamically controllable by the user. The modulation schemes may be used with no FEC, a 1/3-rate convolution code, a 1/2-rate Golay code, or a 1/6-rate concatenated code. All codes are implemented with bit-interleaving to ensure uncorrelated errors. The convolution code is decoded using a 1-norm Viterbi decoder for robust performance in non-Gaussian noise [8]. The user may customize the modem to add new modulation and/or FEC schemes.

E. Ranging, synchronization & timestamping

The modem maintains a clock with a resolution of 1 μs. If the modem is equipped with an optional oven-controlled crystal oscillator (OCXO), this clock provides a low-drift timing reference for one-way travel time (OWTT) ranging and for network protocols that need accurate time synchronization.

The time of transmission or detection of each packet is noted and made available to the network stack. Physical layer packets may be queued for transmission at a predetermined time. This is useful for slotted protocols and necessary for protocols which require the transmission time of a packet to be encoded into the packet. The modem generally automatically controls the transmit/receive mode of the analog front-end for half-duplex communications. The switching between the modes can take several tens of milliseconds for some front-end designs. Timing sensitive protocols wishing to control the transmit/receive mode directly are empowered to do so. This allows extraneous timing information about the protocol or application behavior to be used to improve the protocol performance.

Using the timing services at the physical layer, a ranging protocol is implemented in the network stack. The clocks on the modems are usually not synchronized, although synchronization is possible using an external digital input. Initially two-way travel time measurement is used to estimate range and clock offset between a pair of modems. The clock offset is then communicated to the peer modem and stored for future use. Once this is done, the modems are considered to be *synchronized*. Subsequent range measurements between synchronized

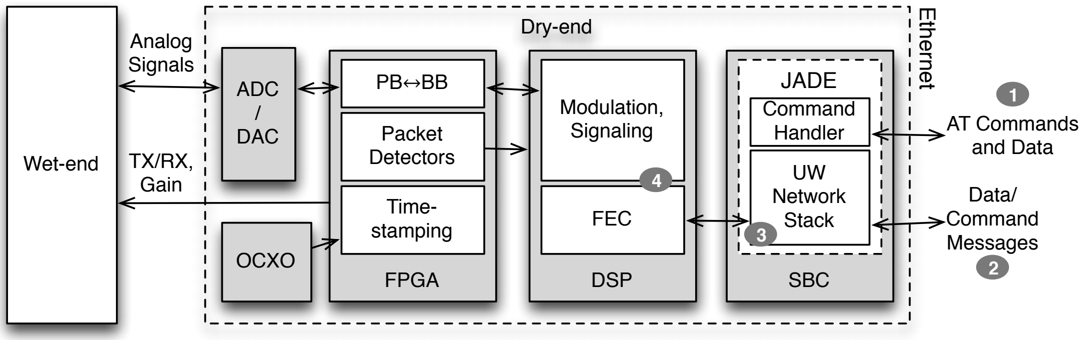

Fig. 3. The UNET-2 modem architecture. Customization/extension points are shows as numbered ovals as per section III-D.

modems can be achieved through OWTT measurements on timestamped packets. The ranging protocol is able to provide a range resolution of 0.1 m.

F. Diagnostic information

Being a research modem, it is important to be able to collect valuable diagnostic information on the environment and modem performance. A logging service in the modem allows terse or verbose logs to be collected on modem operations. These logs may be stored in volatile memory or in Flash memory on the modem and downloaded over Ethernet. The modem also provides online access to a plethora of diagnostic information including noise statistics, packet detector output, delay spread estimates, bit error rate (BER) estimates, baseband packet recordings and passband acoustic recordings. These information may potentially be used by the network stack for selection and configuration of appropriate protocols.

III. Architectural Overview

A. Hardware architecture

The UNET-2 modem comprises a *dry-end* (shown in Fig. 1) that interfaces with a *wet-end* (analog front-end). The dry-end contains a Toradex Collibri PXA270 single board computer (SBC), Analog Devices TigerSHARC TS201 digital signal processor (DSP), Altera Stratix-III field-programmable gate array (FPGA), 24-bit Σ-Δ analog-to-digital converter (ADC), 16-bit digital-to-analog converter (DAC) and an optional 10 MHz OCXO (± 0.05 ppm drift). The wet-end contains the preamplifier, power amplifier, analog matching circuitry and transducer. The dry-end can easily be interfaced with any of several off-the-shelf or custom wet-ends, depending on the desired frequency band and transmit power. The interface uses differential lines for analog input/output noise resilience. More than one wet-end may be used to provide multiple-input-multiple-output (MIMO) capability (using wet-ends with the same frequency band), or multi-band capability [9], [10] (using wet-ends with different frequency bands). In this paper, our focus is on the dry-end electronics, signal processing and networking software in the modem. An off-the-shelf 18–36 kHz wet-end capable of transmitting at a maximum source level of about 185 dB RMS re 1μPa @ 1m was used in all experiments presented in section V.

B. Functional blocks

An overview of the functional architecture of the modem is shown in Fig. 3. The FPGA is primarily responsible for packet detection, passband-baseband conversion and time-stamping. The DSP implements the physical layer signal processing and FEC coding. All higher layer functionality is implemented on the SBC running TinyCore Linux and the Java agent development (JADE) framework [11]. The optional OCXO provides a low-drift clock for accurate timing for OWTT ranging and network protocols that require accurate time synchronization. The 24-bit ADC provides 138 dB of dynamic range making automatic gain control (AGC) at the wet-end unnecessary in most circumstances. The modem can use a digital gain control if the wet-end supports it. The 18–36 kHz wet-end provides a digital gain control with 48 dB of dynamic range, but a fixed gain has been sufficient in all our experiments.

C. Underwater network stack

A network stack based on the underwater networking architecture (UNA) [12] was used in the UNET-1 modem, a predecessor of the UNET-2 modem. Although the stack provided a powerful platform that enabled network simulation code to be field tested using the UNET-1 modem without any need for porting [13], we felt that the UNA has a few shortcomings. Adding new protocol implementations required cross-compilation of the code. Due to lack of support for dynamic loading, only one implementation of a layer could be deployed in the modem at a time. The layered architecture put some unnecessary restrictions that made implementation of protocols requiring cross-layer information awkward. To address these shortcomings, we extended the layered architecture in UNA into an agent based architecture for the UNET-2 modem. The agents are similar to layers in conventional communication stacks, but provide richer interaction and information flow to achieve cross-layer (cross-agent) optimization.

We adopted a subset of the foundation for intelligent physical agents (FIPA) specifications [14] as a framework for the UNET-2 modem network stack, since it provides the necessary infrastructure and is well supported by a large community. JADE [11] is an implementation of the FIPA specifications

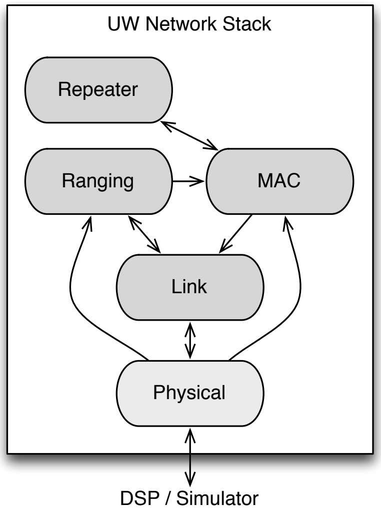

Fig. 4. Underwater network stack in the UNET-2 modem. The Physical agent is a proxy for the physical layer provided by the modem or simulator. The Link agent provides addressing and error checking. The MAC agent provides medium access control. The Ranging agent provides time synchronization and ranging capability. The Repeater agent can be enabled to configure a modem as a repeater node to extend communication range.

that runs on our SBC and manages the all the agents in the network stack. Using this framework, we now have the ability to dynamically load agents into the modem and interact with them in various ways outlined in section III-D. In Fig. 4, we summarize the default agents in the UNET-2 modem network stack. As the network stack matures, we expect the agents to become more sophisticated and the number of agents to grow.

Although the network stack was developed in conjunction with the UNET-2 modem, it is quite independent of the modem. The *Physical* agent in the modem provides an interface to the DSP and FPGA which implement the physical layer in the UNET-2 modem. To operate with a new modem, a one-time effort is required to develop the *Physical* agent for that modem. All other agents in the network stack are able to operate with any modem with minimal or no porting effort. Thus we have developed a general underwater network stack that may be used with any modem.

D. Customization and extension

Flexibility and extensibility was one of the primary objectives in developing the UNET-2 modem. Access to the modem is provided at four levels (depicted graphically in Fig. 3) depending on the required degree of flexibility:

1) For most applications, it is sufficient to access the modem over a TCP/IP socket interface using text commands loosely based on the Hayes AT command structure [15]. The command set has been extended to provide access to sophisticated functionality specific to the UNET-2 modem. The main advantage of this interface is that users and applications can directly interact with the modem in a human-readable form using a simple *telnet* connection. To customize the modem, a start up script with AT commands can be stored on the SBC.
2) A message-based application programming interface (API) is provided for applications that require more sophisticated control over the behavior of the modem. The messages comply to the agent communication language (ACL) defined in the FIPA specifications [14]. We have developed a simple Java API that allows Java applications to communicate with the modem using ACL. Third party ACL/FIPA implementations are available for other programming languages.
3) New behaviors (such as networking protocols) can be added to the modem as FIPA-compliant agents. The easiest way to implement an agent is to develop in Java using the JADE framework. The resulting compiled *jar* file is simply copied into the SBC over Ethernet. A simple Java extension API is also available to extend the AT command set to support the new behaviors added to the modem.
4) New modulation/signaling schemes and FEC schemes can be implemented in C on the DSP. A plug-in API is available to add such schemes to the modem. The implementation of such schemes requires cross-compilation of the C code for the TS201 DSP. The resulting binary is simply copied to the SBC over Ethernet.

IV. Real-time Simulation

It is often useful to simulate network protocols before testing them in the field. A unified simulation framework can be of great value in this effort [13], especially if it simulates the behavior of the modem accurately. Since only the *Physical* agent in the underwater network stack described in section III-C depends on the UNET-2 modem, we have developed a simulation version of the *Physical* agent that emulates the behavior of the UNET-2 physical layer (DSP, FPGA and wet-end) and the underwater channel. With the JADE framework that runs on any Java-enabled computing platform and our simulated *Physical* agent, we can deploy many virtual UNET-2 modems in a simulated environment on a single computer (or on different computers in a network). The virtual modems communicate with each other using TCP/IP multicast packets but provide appropriate propagation

Fig. 5. STARFISH AUV with the UNET-2 modem.

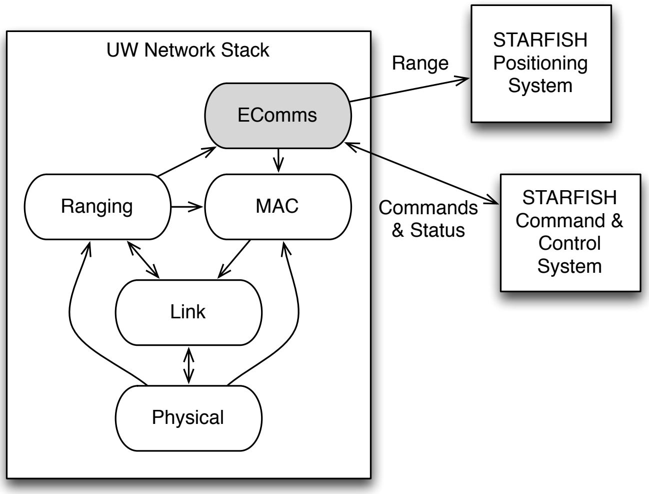

Fig. 6. STARFISH EComms agent deployed in the UNET-2 modem coordinates acoustic communication and ranging activities of the command & control system and positioning system.

delays based on the locations of the virtual modems. Packet loss resulting from collisions and BER is also accurately simulated. Once the simulations are successful, the agents are simply copied to the actual modem and tested in the field. No porting effort is required between simulation and field testing. We use this simulation tool extensively in our own underwater networking research.

V. Applications and Experiments

Next we present some application examples and experiments where the flexibility provided by the UNET-2 modem was exercised.

A. Dynamic frequency band adjustment

During an experiment in Singapore waters, we had two network nodes deployed at a depth of about 4 m. One of the node was moored, while the other slowly moved away. At short range the communication was very good, but as the range increased to more than 2 km, the nodes were unable to communicate. From the diagnostic information provided by the modem, the cause of the loss in communication was identified as very low signal-to-noise ratio (SNR) in the upper half of our frequency band (27–36 kHz). Although packets could be detected at the receiver, error-free communication was not possible. With a simple change in a few physical layer parameters (carrier frequency and the number of null carriers), we were able to move our frequency band to 18–30 kHz and communication was restored.

As shown in Fig. 5, the UNET-2 modem is now used by STARFISH autonomous underwater vehicles (AUVs) [16]. After integrating the modem on the AUV, the communication performance was found to be limited by the self-noise of the AUV. The primary contribution to the noise was from the thruster, which generated a strong tonal between 19–20 kHz depending on the thrust setting and some spectrally smooth noise with a peak around 20 kHz. Again, with a simple change in carrier frequency and the number of null carriers, we were able to move our frequency band to 24–36 kHz to obtain good communication performance.

B. Link tuning

The default *control channel* settings in the UNET-2 modem offer Doppler-resilient robust communication at a data rate of about 400 bps. Although this is sufficient for many applications (e.g. sending commands to AUV, receiving status information from AUV, ranging, etc), some applications (e.g. file/image transfer) require higher data rates. In an experiment in Singapore waters, we deployed two network nodes at a range of 200–400 m. The task was to manually tune the modulation and FEC parameters to obtain a high-rate communication link. By using the phase-coherent modulation scheme, we were able to tune the link to yield data rates of up to 9 kbps for reliable communication within a few minutes. This high-rate link was then be used to transmit a large amount of data rapidly.

The task of an automated link tuning algorithm is to tune the parameters of a link automatically while data is being communicated [6]. We have implemented a *Link Tuner* network stack agent in the UNET-2 modem that is able to use machine learning techniques to observe communication performance and adapt the *data channel* parameters in real-time. Preliminary tank testing shows that the parameters rapidly converge within a couple of minutes to yield error-free data rates of about 5 kbps. The agent is currently being field tested.

C. Cooperative positioning

When deployed in teams, the STARFISH AUVs are able to use range measurements between vehicles for position estimation and navigation [17], [18]. With the UNET-2 modems, the STARFISH AUVs are now able to obtain range information at regular intervals using OWTT measurements. As shown in Fig. 6, a STARFISH *EComms* (external communications) agent deployed on the UNET-2 modem coordinates the transmission of ranging beacons and status information, and routes range information and incoming commands to the STARFISH command and control system. Ranging information from AUVs with good navigation sensors enables AUVs with poor navigation sensors to accurately navigate and accomplish their missions.

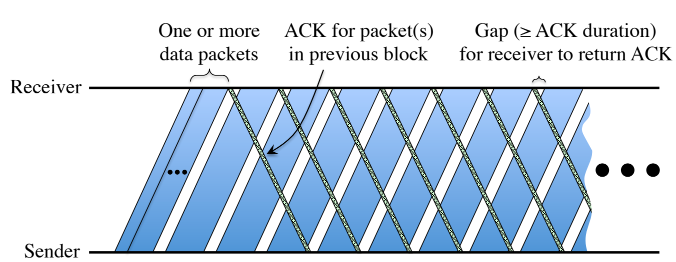

Fig. 7. An illustration of juggling-like stop-and-wait scheme for half-duplex modems in presence of long propagation delay (figure adapted from [20]).

Fig. 8. A UNET-PANDA node being prepared for deployment as a network node in a multihop networking experiment.

The UNET-2 modem was recently successfully integrated and tested with autonomous surface crafts (ASCs) from the Singapore-MIT alliance for research and technology (SMART) to demonstrate cooperative positioning between ASCs and AUVs.

D. *Juggling-like stop-and-wait protocol*

In [19], a new transmission scheme, known as the juggling-like stop-and-wait scheme, was proposed to improve the channel utilization of point-to-point data transfer between half-duplex acoustic modems. This scheme is studied in detail in [20] and depicted in Fig. 7. As seen in the figure, the timing requirements for transmission and reception are stringent; any shift in timing (too early or too late) is unacceptable due to the half-duplex nature of the modems [21]. This protocol has been implemented and tested on the UNET-2 modem. Accurate control of packet transmission timing was critical for the successful implementation of this protocol. Since a small tank does not provide the necessary propagation delays for such a protocol to work, the simulation tools were vitally important during the implementation. Accurate timing models in the simulator allowed us to develop and test the protocol in simulation prior to field testing.

E. *UNET-PANDA network nodes*

The UNET-2 modem has been integrated into a UNET-PANDA network node shown in Fig. 8. The UNET-PANDA is a rapidly deployable and self-recovering bottom-mounted system that can be used as a self-contained sensing and network node. It contains a UNET-2 modem, additional SBC, GPS and batteries. Once deployed, the electronics package of the UNET-PANDA is anchored and floats 1–2 m above the seabed. It can be later recovered via an acoustic command that triggers the self-recovery unit of the UNET-PANDA to surface. The unit remains attached to the electronics package and anchor via a line. All three components (anchor, electronics package, self-recovery unit) of the UNET-PANDA are then recovered from a surface vessel. The UNET-PANDA node has been used in several underwater networking experiments in Singapore [22].

VI. Conclusions

In this paper we described the key features of the UNET-2 software-defined modem, particularly focusing on the flexibility and adaptability that it provides researchers with. We described the hardware and software architecture and the network stack that the modem uses. We also illustrated the flexibility provided by the modem through some practical application examples and experiments. We believe that the UNET-2 modem provides a powerful and extensible tool for underwater network research, allowing a variety of novel protocols to be easily implemented and tested. The network stack in the UNET-2 modem is not strongly tied to the modem, and can be easily used with other modems and also as a powerful simulation tool. We have found that the ability of the network stack to simulate an underwater network in the lab with the exact implementation of network protocols to be tested in the field is invaluable.

Acknowledgments

We would like to thank several colleagues and collaborators who have been instrumental in making the UNET-2 modem a success. Alan Low and Ang Yen Khim from ST Electronics (Info-Comm Systems) helped develop and test the dry-end electronics for the modem. Manu Ignatius developed a substantial part of the DSP software. Many other colleagues such as Shankar Satish, William Tan, Rohit Bhatnagar and collaborators such as Sultan Imenov (SMART), Tawfiq Taher (SMART), Marek Doniec (MIT), Haojie Zhuang (I2R) and Ma Xiaoping (I2R) provided valuable feedback as users of the modem, and helped test the modems in a variety of application scenarios. Tan Soo Pieng and Mohan Panayamadam helped develop the UNET-PANDA nodes which were used in many of the tests. Profs. Barbastathis and Patrikalakis (MIT) kindly provided access to SMART ASCs for testing cooperative positioning.

References

[1] M. Chitre, S. Shahabudeen, and M. Stojanovic, “Underwater acoustic communications and networking: Recent advances and future challenges,” *The Spring 2008 MTS Journal, ”The State of Technology in 2008”*, vol. 42, no. 1, pp. 103–116, 2008.

[2] M. Chitre, M. Motani, and S. Shahabudeen, “A scheduling algorithm for wireless networks with large propagation delays,” in *IEEE OCEANS’10 Sydney*, Australia, May 2010.

[3] H.-H. Ng, W.-S. Soh, and M. Motani, “BiC-MAC: Bidirectional-concurrent MAC protocol with packet bursting for underwater acoustic networks,” in *OCEANS 2010*, sept. 2010, pp. 1 –7.

[4] K. Kredo, P. Djukic, and P. Mohapatra, “STUMP: exploiting position diversity in the staggered TDMA underwater MAC protocol,” in *INFOCOM 2009, IEEE*, 2009, pp. 2961–2965.

[5] M. A. Chitre, J. R. Potter, and S. H. Ong, “Optimal and near-optimal signal detection in snapping shrimp dominated ambient noise,” *Oceanic Engineering, IEEE Journal of*, vol. 31, no. 2, pp. 497–503, 2006.

[6] S. Shankar, M. Chitre, and M. Jayasuriya, “Data driven algorithms to tune physical layer parameters of an underwater communication link,” in *OCEANS’10 Sydney*, Australia, May 2010.

[7] M. Chitre, S. H. Ong, and J. Potter, “Performance of coded ofdm in very shallow water channels and snapping shrimp noise,” in *OCEANS, 2005 MTS/IEEE*, 2005, pp. 996–1001 Vol. 2.

[8] M. Chitre, J. R. Potter, and S. H. Ong, "Viterbi decoding of convolutional codes in symmetric α-stable noise," *Communications, IEEE Transactions on*, vol. 55, no. 12, 2007.

[9] L. Freitag, M. Grund, J. Partan, K. Ball, S. Singh, and P. Koski, "Multi-band acoustic modem for the communications and navigation aid auv," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1080–1085.

[10] S. Shahabudeen, M. Chitre, and M. Motani, "A multi-channel MAC protocol for AUV networks," in *Oceans 2007-Europe*. IEEE, 2007, pp. 1–6.

[11] (2012) Java Agent Development Framework (JADE). [Online]. Available: http://jade.tilab.com/

[12] M. Chitre, L. Freitag, E. Sozer, S. Shahabudeen, M. Stojanovic, and J. Potter, "An Architecture for Underwater Networks," in *OCEANS 2006 - Asia Pacific*, May 2006, pp. 1–5.

[13] S. Shahabudeen, M. A. Chitre, M. Motani, and Y. S. Low, "Unified Simulation and Implementation Software Framework for Underwater MAC Protocol Development," in *IEEE/MTS Oceans'09 Conference, Biloxi, US*, October 2009.

[14] (2012) The Foundation for Intelligent Physical Agents (FIPA). [Online]. Available: http://www.fipa.org/

[15] F. Durda IV. (1993) The AT command set reference. [Online]. Available: http://nemesis.lonestar.org/reference/telecom/modems/at/summary-at.html

[16] T. B. Koay, Y. T. Tan, Y. H. Eng, R. Gao, M. Chitre, J. L. Chew, N. Chandhavarkar, R. Khan, T. Taher, and J. Koh, "STARFISH - A Small Team of Autonomous Robotics Fish," in *3rd International Conference on Underwater System Technology: Theory and Applications 2010*, Cyberjaya, Malaysia, Nov 2010.

[17] R. Gao and M. Chitre, "Cooperative positioning using range-only measurements between two AUVs," in *IEEE OCEANS'10 Sydney*, Australia, May 2010.

[18] Y. T. Tan and M. Chitre, "Single beacon cooperative path planning using cross-entropy method," in *IEEE/MTS Oceans'11 Conference, Hawaii, USA*, Kona, Hawaii, USA, September 2011.

[19] M. Gao, W.-S. Soh, and M. Tao, "A transmission scheme for continuous ARQ protocols over underwater acoustic channels," in *Proceedings of IEEE ICC*, Dresden, Germany, June 2009.

[20] M. Chitre and W.-S. Soh, "Reliable point-to-point underwater acoustic data transfer: To juggle or not to juggle?" *IEEE/ACM Transactions on Networking (TON)*, 2011 (under review).

[21] R. Bhatnagar, "Implementation and testing of ARQ protocols for underwater networks," National University of Singapore, EE5003 Project Report, 2012.

[22] M. Xiaoping, A. Valera, and H. P. Tan, "Final report for robust multihop underwater network to support long-range sensing applications," Institute for Infocomm Research (I2R), Tech. Rep., 2011.