Project Deliverable

UNET II - D6

Research & Development in Underwater Acoustic Communication and Networking Technology

(Final Report)

Prepared by

Mandar Chitre

Version 1.1

December 1, 2011

Contents

1. Project overview

1.1 Background

This report forms deliverable D6 of the project "Research & Development in Underwater Acoustic Communication and Networking Technology," or "UNET II" for short. The deliverable provides final results from two sets of field trials:

1. Point-to-point communication performance between modems
2. Robust multi-hop network for long-range sensing

The point-to-point performance evaluation was conducted in collaboration with DSO National Laboratories, and the results are detailed in Chapter 2. The roust multi-hop underwater network for long-range sensing was developed in collaboration with I2R, and the field trials results for that network are detailed in the I2R report forming an enclosure to this report.

1.2 Project outcome

The primary goal of the UNET II project was to develop a modem capable of robust performance that would form a platform for future underwater communications & networking research. As the modem is likely to be used on autonomous underwater vehicles (AUVs), we wanted the modem to work reliably in presence of platform motion and consequent Doppler. This goal has been largely achieved. The UNET II modem has been able to demonstrate robust performance at ranges up to 2 km[1], speeds of up to 5 knots

[1] The modem is expected to communicate well at ranges of up to 3 km, but due to limitations in experimental setup, we were able to test the modem in the field at a maximum distance of 2 km.

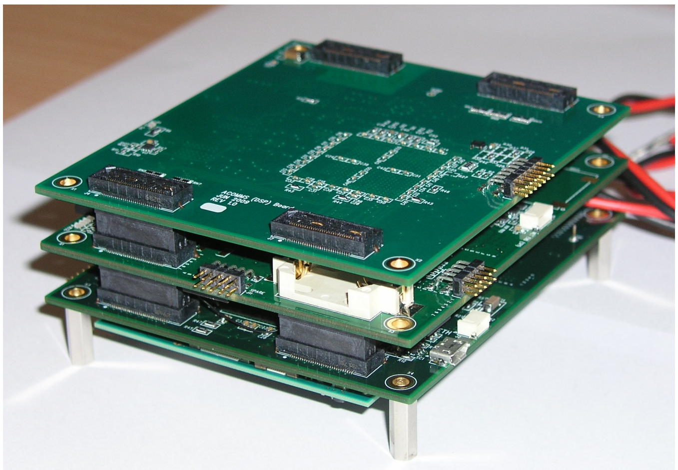


Figure 1: UNET II modem dry-end

and data rates of up to 9 kbps[2]. The modem uses an incoherent OFDM scheme for robust communications and a coherent OFDM scheme for high speed communications. It supports multiple physical layer schemes operating in parallel and supports dynamic switching between different software defined radios (SDR), potentially supporting other communication schemes such as QPSK/DFE, FH-BFSK, etc for compatibility with other modems. The default frequency band of operation is 18-36 kHz. This frequency band can easily be changed by simply replacing the wet-end electronics and transducer[3].

A Java network stack based on the JADE[4] framework has been developed for the modem. This stack supports pluggable network agents that can easily be developed and deployed on the modem. A basic medium access control (MAC) agent, link layer agent and a ranging agent has been developed for the UNET II modem. These agents together support point-to-point datagram transfer that is currently being used by the STARFISH AUV for command and status information exchange. They also support accurate time synchro-

[2]In the default robust mode, the modem sports a data rate of about 300-500 bps. The data rate can be increased by tuning link parameters.

[3]The modem is compatible with two wet-end systems. The primary wet-end system is an off-the-shelf solution from Evologics, and is available in different frequency band (7-14 kHz, 18-36 kHz, 48-78 kHz). The secondary solution is a custom solution using locally developed electronics and a transducer designed by SensorTech (18-36 kHz).

[4]JADE (Java Agent DEvelopment Framework) is a software framework originally developed for the telecommunications industry, that simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA specifications. More details on the framework can be found at http://jade.tilab.com/

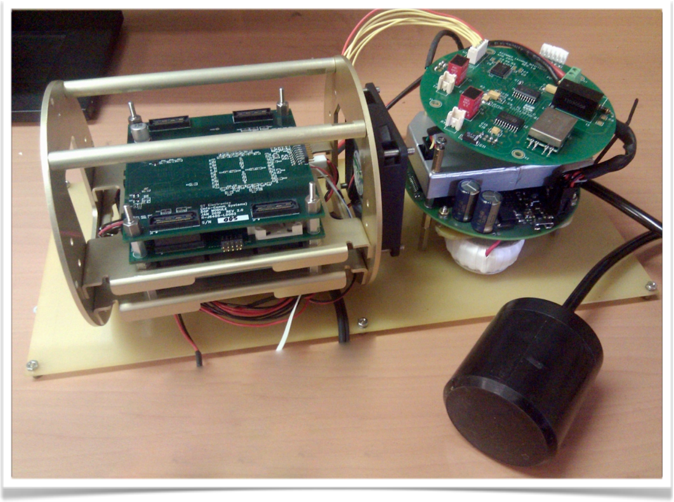


Figure 2: UNET II modem electronics (dry-end and off-the-shelf 18-36 kHz wet-end)

nization and one-way travel time (OTT) based ranging through the use of the modem's oven-controlled crystal oscillator (OCXO).

Finally, in collaboration with I2R, network protocols for long-range sensing were explored. These protocols were implemented using the UNET II modem and successfully field tested.

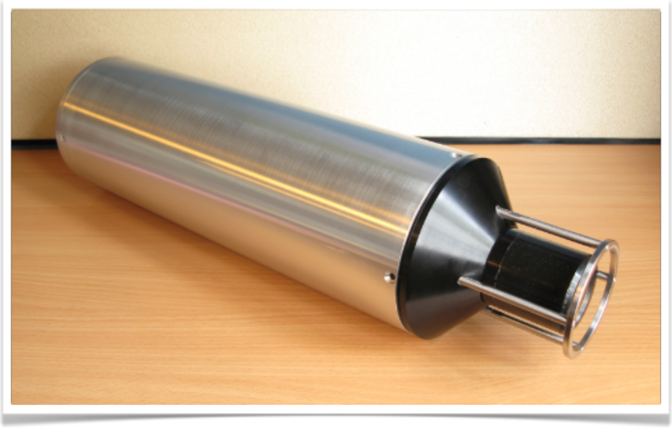


Figure 3: UNET II modem in a stand-along mechanical housing

1.3 Future work

The UNET II project lays the groundwork for exciting research in underwater communications and networking by providing a flexible and powerful platform that will help implementation and testing of novel ideas. In the next phase of the project (UNET III), we plan to use this platform to explore higher speed communications and automated link tuning. We also plan to develop a "underwater WiFi" network stack which will allow ad-hoc networks to be easily deployed using the UNET II modems.

To provide the flexibility and power needed for research, the UNET II modems were developed using high computational power and versatile components. As a consequence, the modem consumes more power than absolutely necessary and is larger and costlier than it could be. As commercial and defense exploitation of the technology is likely to demand low power and smaller modems in the future, the UNET III project will also undertake a feasibility study and design exercise to make the UNET II modem smaller and more energy efficient, and perhaps even cheaper.

Table 1: UNET II modem specification summary

Dry End		
Dimensions	10 × 9 × 5.5 cm	
Modulation	Incoherent OFDM	Robust mode
	Coherent OFDM	High-speed mode
Error correction	Rate 1/2 Golay Code	
	Rate 1/3 Convolution Code	
	Rate 1/6 Concatenated Code	
Packet detectors	3	Simultaneous PHY
Carrier frequency (Fc)	Programmable	Nominal: 27 kHz
Bandwidth	2/3 × Fc	Nominal: 18 kHz
ADC	24-bit at 4×Fc Sa/s	
DAC	16-bit at 16×Fc Sa/s	
Clock precision	1 μs	Over-controlled crystal
Network stack	Java / JADE / FIPA	User extensible
Wet End (18-36 kHz)		(off-the-shelf)
Dimensions	9.5 cm × 10.5 cm diameter	Electronics
	6 cm × 5.5 cm diameter	Transducer
Frequency band	18-36 kHz	Options: 7-14 kHz, 48-78 kHz
Source level	192 dB re μPa at 1m	Peak-to-peak, 50% duty cycle
Receiver gain	0-48 dB	Programmable
Performance		
Range	3 km	Tested up to 2 km
Data rate	300-500 bps	Robust mode
	up to 9 kbps	High-speed mode
Doppler robustness	10 knots (robust mode)	Tested to 5 knots
Range resolution	0.1 m	
Power consumption	6 W	Stand-by mode
	12 W	Active mode (nominal)
	20 W	TX mode (maximum)

2. Underwater communication performance

2.1 Background

DSO National Laboratories approached ARL to participate in a field experiment to test the performance of ARL's UNET II modem. ARL agreed to participate with two aims in mind. The primary aim was to test the robustness of communication at various ranges and platform speeds. The secondary aim was to collect data on data rate performance as a function of link parameters, with an aim to use the data in the development of link tuning algorithms in the future. The experiment was held from 1-3 November 2011 at Selat Pauh. The findings of the experiment are detailed below.

2.2 Experimental setup

The experiment was conducted over a period of 3 days at Selat Pauh anchorage using modems deployed from 2 boats. The receiver boat (RX) was tied to a navigation buoy at $1^o 12.9385'$ N, $103^o 44.4118'$ E. The transmitter boat (TX) was mobile and moved to various locations up to 2 km away from RX during the experiment. Both TX and RX boats deployed the modem at a nominal depth of about 5 m using a rope and weight arrangement as illustrated in Figure 4. During the long range test (2 km) and Doppler tests (TX moving), the TX boat deployed the modem on a rigid pole at a depth of approximately 4 m. The average water depth in the area was approximately 15 m.

During the trials, a large barge was parked about 50-100 m from RX, and works on the barge provided occasional loud noises. Several ships and boats

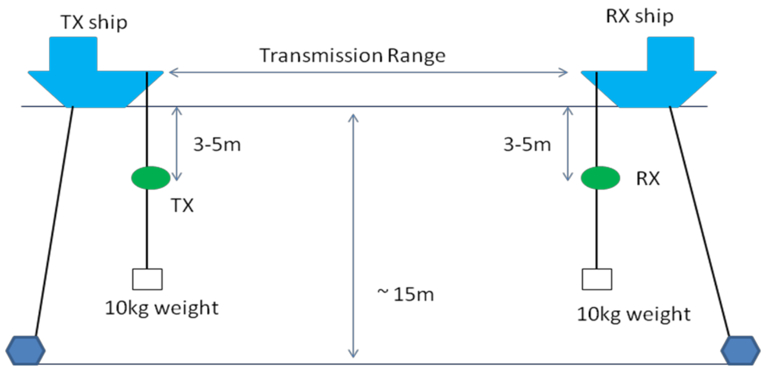


Figure 4: Planned trial setup (illustration by Terence Tan, DSO)

were anchored and transited within 0.5-1 km of RX, and on one occasion, a high speed noisy boat came right next to RX during an experiment to deliver supplies. The sea state was mostly calm with occasional drizzle or rain.

On day 1 and day 2, communication at several different ranges was tested. TX often drifted during the tests providing a low speed moving platform, while the motion of RX was constrained to a small area near the navigation buoy. During these tests, the robust low data rate communication scheme was primarily tested. However, occasionally the link was tuned manually to obtain data on high data rate communications. On day 3, communication was tested at higher speeds at various ranges using a pole setup on TX, while it moved around with its engines operating. For these tests, only the robust low data rate communication schemes were used. GPS loggers on RX and TX provided information on the position and speed of RX and TX.

2.3 Results & discussion

The data collected during the experiments was post processed and divided into a series of "tests". Each test consisted of a number of data packets transmitted from TX to RX in rapid succession. The data packets were decoded and compared against the transmitted data to check for errors. If the data was received without any errors, the packet was deemed to be successfully delivered. A packet delivery rate (PDR) was computed and tests with PDR $> 75\%$[1] were considered to be successful tests and tabulated below.

[1]For a typical packet size of 144 bits, PDR $> 75\%$ is equivalent to a BER of about 10^{-3} or better under the assumption of independent errors.

Test ID	Min Range (m)	Max Range (m)	TX Speed (kts)	Link Rate (bps)	Packets	Packet Length (bits)	PDR (%)
D1R100T2	146.8	148.5	0.2	1125	31	144	100
D1R100T7	165.1	168.4	0.2	2609	23	144	100
D1R100T8	214.7	230.7	0.4	374	87	144	100
D1R100T10	175.6	176.5	0.2	5883	43	240	95
D1R100T11	171.4	173.1	0.2	**8824**	42	720	83
D1R1200T1	1202.3	1206.3	0.2	281	76	144	86
D1R1200T5	1183.2	1186.8	0.4	281	22	144	100
D1R1200T7	1179.3	1180.7	0.2	374	25	144	100
D1R1200T11	1185.7	1186.6	0.2	1858	26	720	96
D1R400T2	440.1	441.3	0.0	1125	29	144	90
D1R400T5	423.4	425.5	0.6	2941	24	240	96
D1R400T6	421.9	422.8	0.2	5219	12	240	92
D1R400T8	432.2	436.5	0.4	3530	14	720	86
D1R750T4	754.8	755.0	0.0	374	15	144	100
D1R750T5	753.0	757.5	0.2	581	18	144	100
D1R750T7	753.9	756.0	0.4	2609	22	144	77
D2R2000T6	2030.7	**2033.9**	0.4	186	39	72	100
D2R2000T7	2013.7	**2021.8**	0.4	281	30	72	93
D2R600T1	621.4	626.2	0.6	374	18	144	100
D2R600T3	629.8	633.4	1.2	1125	21	144	95
D2R600T7	851.6	858.0	0.8	374	13	144	100
D2R600T12	1113.5	1127.3	1.0	375	23	288	100
D2R600T14	1181.2	1340.6	1.0	374	329	144	100
D2R600T15	1369.0	1374.7	1.0	374	13	144	100
D3RdT1	429.2	429.9	0.0	374	11	144	100
D3RdT2	428.9	430.7	0.0	374	26	144	100
D3RdT3	431.1	431.9	0.0	374	16	144	100
D3RdT4	430.2	430.9	0.0	374	14	144	100
D3RdT7	431.4	432.9	0.0	374	10	144	100
D3RdT8	373.4	464.7	**4.7**	374	13	144	100
D3RdT9	138.7	187.3	4.3	374	10	144	100
D3RdT10	491.6	652.6	4.3	374	17	144	100
D3RdT11	678.1	930.9	3.9	374	30	144	97
D3RdT15	597.4	1001.6	**4.5**	374	43	144	100
D3RdT16	1031.1	1314.6	**4.5**	374	30	144	80
D3RdT18	931.8	1167.0	4.3	374	38	144	100
D3RdT19	512.6	708.5	**4.7**	374	28	144	93

The tests with the highest data rate (8,824 bps), longest range (2,022-2,034 m) and highest platform speed (4.5-4.7 knots) are highlighted for quick reference. Ranges more than 2 km and speeds more than 4.7 knots could not be tested due to limitations of the experimental setup. The pole to mount the TX modem only allowed deployment of the modem at about 4 m depth and could handle speeds of no more than 4-5 knots. With the TX modem deployed at about 4 m depth and the RX modem deployed at about 6-7 m depth, the transmission range is limited to about 2 km due to Lloyd's mirror effect[2]. A longer range may be achieved by deploying the modems are larger depths. The theoretically computed range of the modems is about 3 km.

Although manual tuning of modem parameters is far from optimal and therefore unlikely to give the best data rates, plotting the achieved data rates against range provides a rudimentary idea of the performance envelope of the modem. An often quoted figure of merit for modems is the range-rate product. The achieved performance of the UNET II modems in this trial was observed to be approximated by range-rate product of about 2×10^6 bps-m as shown in Figure 5. With appropriate link tuning algorithms and modems deployed deeper at longer ranges, we expect to be able to improve the measured range-rate product further.

As can be seen from the last 8 rows of the performance table, the robust low data rate scheme in the modem is unaffected by platform motion and consequent Doppler. Higher data rate schemes were not tested during the motion tests due to limited time available to change modem parameters as TX is constantly moving.[3]

The UNET II modem is capable of measuring range using a time synchronization and ranging protocol built into it. To test the performance of the ranging, we compare the range measured by the modems with the range estimated from GPS logs in Figure 6. At 200 m and 1.2 km ranges, the acoustic and GPS estimates agree to within 1-2 m. However, for the 400 m and 750 m ranges, the difference was larger (10-15 m). Since the difference is within the expected accuracy of the GPS, we were unable to ascertain

[2]A rule of thumb for our frequency band of interest (18-36 kHz) is that the range cut-off is approximately equal to 16 times the square of the sum of depths of the transmitting and receiving modems. In mathematical terms, if d_{tx} and d_{rx} are the depths of the transmitting and receiving modems respectively, the maximum achievable range is approximately $16(d_{tx} + d_{rx})^2$.

[3]High data rate schemes in the modem are currently not Doppler compensated and we do not expect that they will perform well under Doppler. Doppler compensation for high data rate schemes has been tested in simulation, and will be implemented into the modem during the next phase (UNET III).

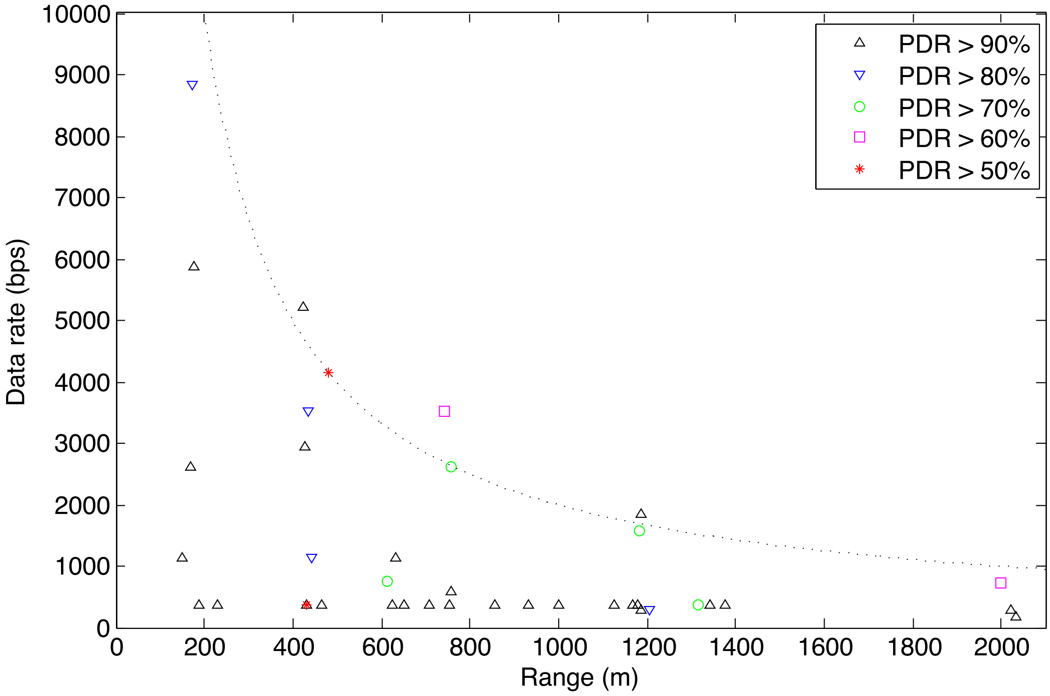


Figure 5: Range-rate plot for various tests with PDR > 50%

whether the acoustic range or the GPS range was closer to reality.

2.4 Summary

In summary, the UNET II modem demonstrated robust performance in Singapore waters. At low data rates of about 300-400 bps, it was able to communicate at all ranges up to over 2 km and speeds of up to 5 knots. Higher ranges and speeds were not tested due to limitations of the experimental setup. By changing the tunable scheme parameters of the modem, the data rate could be increased. Data rates of up to 9 kbps were achieved at short ranges, with the data rate reducing such that the range-rate product stayed constant at about 2×10^6 bps-m. The ranging and synchronization protocol in the modem was tested and shown to have performance within GPS accuracy.

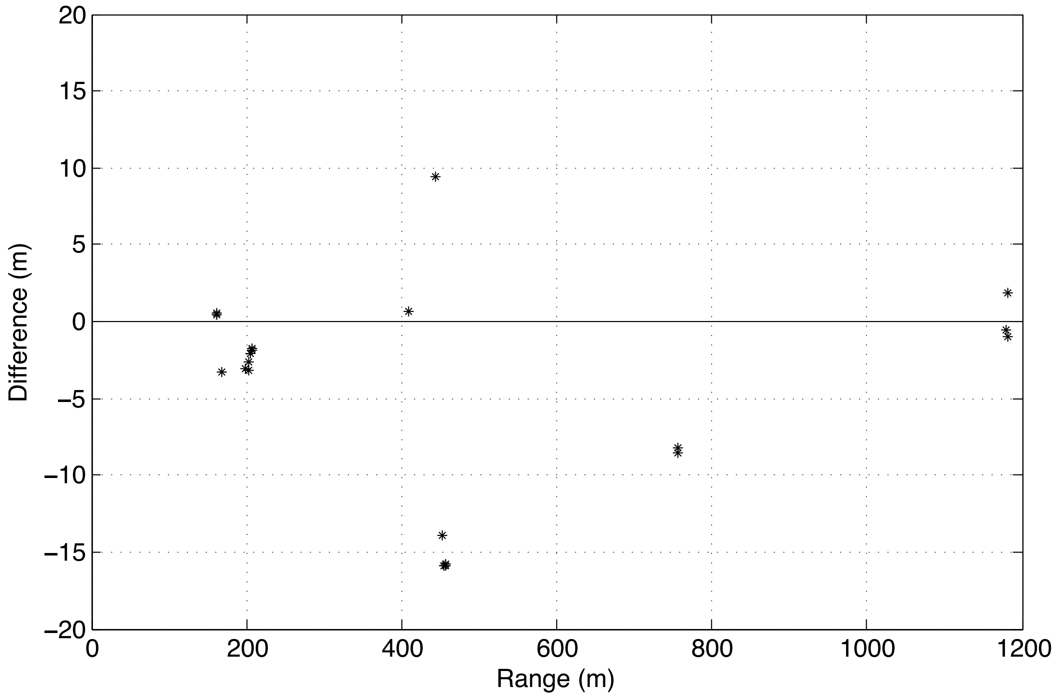


Figure 6: Difference between range estimates from modem and GPS

3. Underwater networking for long-range sensing

The following report from I2R is enclosed as an integral part of this deliverable:

(A) Ma Xiaoping, Alvin Valera and Tan Hwee Pink, “Final Report for Robust Multihop Underwater Network To Support Long-Range Sensing Applications,” November 2011.

FINAL REPORT FOR ROBUST MULTIHOP UNDERWATER NETWORK TO SUPPORT LONG-RANGE SENSING APPLICATIONS

Department Name	Networking Protocols
Authors(s)	Ma Xiaoping, Alvin Valera, and Tan Hwee Pink
Date	November 25, 2011
Version Number	1.0

DOCUMENT ACCEPTANCE

ACCEPTED:	Tan Hwee Pink		Date __ / __ / __
	Project Manager	*Signature*	
ACCEPTED:	Mandar Chitre		Date __ / __ / __
	Customer	*Signature*	

DOCUMENT REVISION

Revision number	*Date*	*Description of changes*
1.0	November 25, 2011	Initial document release

Contents

List of Figures

1 Introduction

1.1 Project Aim

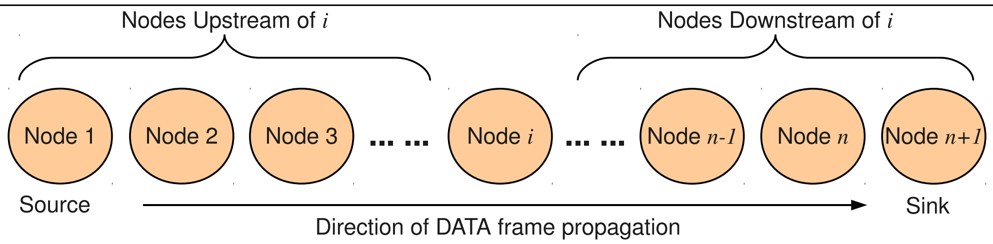
The aim of this project is to develop a robust multihop underwater network to support long-range sensing applications.

1.2 Technical Objective

In a typical underwater sensor network, sensor nodes are interconnected via wireless links to one or more underwater sinks that are responsible for reliably relaying data to a surface station. While direct communication between sensors and sink nodes (long range, high power) is simple, multihop (short-range, low power) communication using complex protocols may be necessary, for example, in military surveillance applications, where operational covertness and low power consumption are critical requirements to ensure that the network is able to operate for long durations without being detected and the need for replenishing the energy resource on the nodes. In addition, unlike other forms of wireless networks, acoustic communications is the most promising for underwater sensor networks as RF is severely attenuated in water and optical signals suffer from scattering.

To achieve long-range sensing, bearing in mind the high costs of underwater acoustic modems, this report considers an n+1 node underwater network that is deployed in a linear topology comprising n hops, where node n+1 is designated as the sink (data collection) node. This is illustrated in Figure 1.

Figure 1 Linear Network Topology (n-hops).



1.3 Report Content

This document presents a comprehensive account of all the accomplishments of the project titled Robust Multihop Underwater Network To Support Long-Range Sensing

Applications. The entire project spans for thirty-two months from 1^{st} April 2009 to 30^{th} November 2011 and is divided into three main tasks. To fulfil these three tasks, a network architecture is established and is presented in Section 2.

The three main tasks involved in this Project are marked as project milestones or deliverables:

1. **Data Link Layer (DLL)** – Existing Logical Link Control (LLC) and Medium Access Control (MAC) schemes designed for RF wireless channels need to be adapted to underwater acoustic channels. Taking into account the long and dynamic propagation delay, and harsh channel conditions, we designed and implemented (i) MAC schemes to coordinate channel access among nodes so as to minimize collisions due to channel contention to maximize energy efficiency; and (ii) Automatic Repeat reQuest (ARQ) scheme to provide reliable per-hop data delivery to maximize end-to-end throughput. Details of our design and implementation of the DLL are presented in the first two project deliverables [1] [2]. In this report, we will describe its salient features in Section 3.

2. **Multihop Routing/Data Delivery** – Existing multihop routing protocols are designed for high bandwidth radio frequency links, which are very different from underwater acoustic channels. We designed and implemented multihop data delivery schemes targeted at the lossy underwater acoustic channel with long propagation delay, and subject to temporal blackouts. Details of our design and implementation of multihop data delivery are presented in [2] and [3]. In this report, we present the key features in Section 4.

3. **Integration to modem platform, testing and demonstration** – Once the designed MAC, ARQ and data delivery schemes are validated using simulations, they were integrated onto the underwater acoustic modem developed by the National University of Singapore (NUS) Acoustic Research Laboratory (ARL). The system was field tested at various sites. There are two versions of the acoustic modems. The first version was used in the earlier field tests in this project until the second version, a deliverable from ARL for this project, was completed. The first version of the modems is referred to as the old modems in this report and the second version, the "new modems". Details of the integration with the "old modems" can be found in [2] and [3]. Here we present the integration and field tests for the new modem in Section 5.

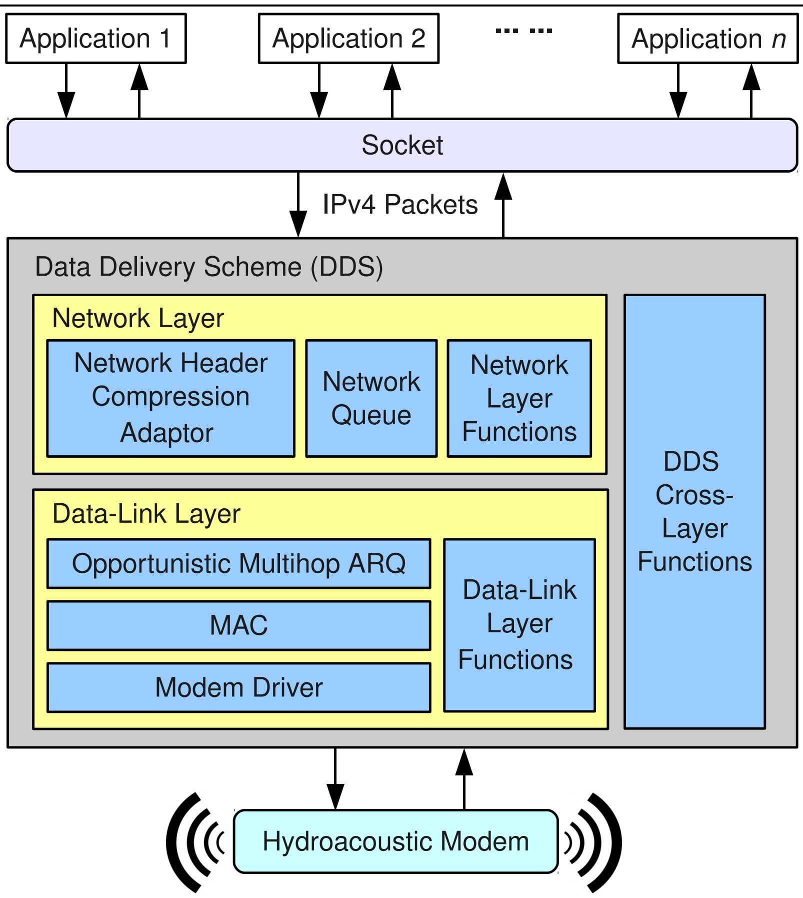
In addition, to demonstrate the utility and reliability of the proposed data delivery scheme in a real-life linearly arranged multihop underwater network, a sample application (File Transfer Application) is presented in Section 6.

Lastly, a summary of the project and recommendations for future work are presented in Section 7.

2 Overview of the Network Architecture

To achieve the project objectives, a network architecture that utilizes the ARL modems for networking, communications and configuration is established, as shown in Figure 2. The network architecture is essentially similar to the Open Systems Interconnection model (OSI model), and consists of an application layer, a network layer, a data link layer and a physical layer (transport layer is missing in the current design). In particular, to minimize the dependency on the system inside the modems and to make the design more robust to changes in the interface specifications, the architecture provides an abstraction (Lower MAC layer and Modem Driver) to wrap the functions provided by the modems. With this abstraction, higher level components are loosely coupled and are immune to changes associated with the modem. The interactions between two neighbouring layers are loosely based on [4] which presents a general framework for underwater network.

Figure 2 Network Architecture



2.1 Application Layer

The application layer is where user applications can be implemented. In the current design, user applications send and receive data through a socket interface which provides multiplexing and de-multiplexing to support multiple applications. The interface between user applications and the data delivery scheme is adapted from the design in [5], whereby instead of redesigning an entire networking stack, the data delivery scheme leverages on the widely-tested TCP/IP protocol stack, using IP packets as the standard exchange format with the upper layers.

Currently the socket interface only serves as a path for a user application and the data delivery scheme to communicate with each other. It does not serve as a transport layer for the whole underwater network. In the future, a transport layer that is specially designed for an underwater network should reinforce or replace the socket interface in the network architecture.

As the lack of a transport layer for the underwater network, user applications have to take care of missing packets, out-of-order packets as well as flow control. An example of user applications that relies on the data delivery scheme is presented in Section 6.

2.2 Data Delivery Scheme

The data delivery scheme (DDS), the focus of this project, spans over both the network layer and the data link layer (DLL) so as to enhance the efficiency of packet routing. It endeavours to make full use of the broadcast nature as well as spatial and temporal variance of the underwater acoustic channel to improve the overall network performance. The data delivery scheme is the focus of this project.

The DDS cross-layer functions enable the data link layer to access the compressed IP header of DATA frames and the network queue, which is most useful for the Opportunistic Multihop ARQ component in the DLL to manipulate the network queue whenever needed.

2.2.1 Network Layer

The network layer is responsible for multihop data delivery in the data delivery scheme. The Network Queue functional block is used to enqueue DATA frames, either from an application or an upstream node (relaying) if the data link layer is currently busy processing another frame.

In addition, the network layer has a functional block termed Network Layer Compression Adaptor that is used to compress the standard IPv4 headers and reduce address size so as to reduce the overhead as much as possible.

2.2.2 Data Link Layer

The data link layer is divided into several sub-components. The Opportunistic Multi-hop ARQ is a functional block in which the key intelligence of the data delivery scheme resides. It makes decisions to process, queue or discard overheard DATA & ACK frames through the DDS Cross-Layer functions. The medium access control (MAC) component actually consists of an upper MAC and a lower MAC sub-components. The upper MAC implements the medium access protocol while the lower MAC performs low-level device access through Modem Driver which is the module that directly control modems to carries out specific tasks.

2.3 Physical Layer

The most important hardware component in the network protocol stack is the underwater acoustics modem developed by the National University of Singapore (NUS) Acoustics Research Laboratory (ARL), which serves effectively as the physical layer in the network architecture. As mentioned in Section 1.3, there are two versions of the acoustic modems. Prior to the delivery of the modems from ARL for this project, an older version was used to test and verify the successful integration of the data delivery scheme onto the ARL modems. The modems were also used in several sea trials to evaluate the performance of the DDS.

At the beginning of the project, ARL provided a software interface, called the Framework API (FAPI), that could be used to perform low-level operations on the old modems including transmission, reception, and configuration. As the project progressed and the new modems were ready, ARL provided the lower-level operations (AT-Commands) for us to control the new modems. As a result, FAPI has now been combined into the Modem Driver functional block within the Data-Link layer which uses AT-Commands to carry out specific tasks. All functional blocks of the data delivery scheme except for the Modem Driver functional block are independent of the type of hydroacoustic modem used. The Modem Driver module can thus be easily replaced, allowing the data delivery scheme to be ported onto different modems without considerable effort.

3 Data Link Layer

The details of the design of the data link layer including simulation results that are obtained from Qualnet Simulator [6] are presented in Progress Report #1 [1] as well as in Progress Report #2 [2]. In this section, a brief discussion of the data link layer will be presented. In addition, a time synchronization scheme that is newly incorporated into the TDMA-related MAC protocols will be presented in detail. The Opportunistic Multihop ARQ functional block plays a key role in the whole data delivery scheme and will be discussed in Section 4.

3.1 Upper-MAC Protocols

The upper MAC sublayer plays the key role of controlling medium access, detecting and avoiding collisions in our underwater multihop network, processing frames from upper layer and passing them down to Lower MAC according to implemented medium access rules.

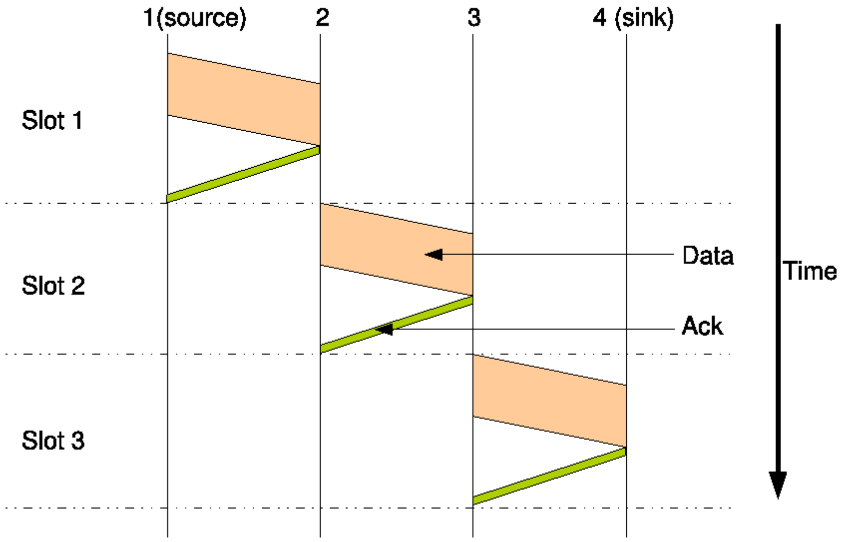
With significant signal propagation delays, amplified by the limited bandwidth and transmission rates, classical channel sensing schemes in conventional networks are not suitable for underwater acoustic networks. A network deemed by such sensing schemes to be currently free from traffic might in actual fact have a signal (or even signals) already "in flight", resulting in frequent collisions. The situation is made more challenging when we take into account that underwater transmissions attenuates quickly and are at times (and unpredictably) uni-directional (channel asymmetry).

Time Division Multiple Access MAC

Such considerations led us to consider a **Time-Division-Multiple-Access (TDMA)** based scheme that enforces strict medium access for each network node based on its allocated time slice, ensuring that the transmission medium is collision-free at all times. A basic description of the TDMA MAC framework for a network with N nodes is as follows:

- Time is divided into slices called *frames*.
- A *frame* is further divided into N slices called *slots*, with a *slot* assigned to each of the N nodes.
- Each node can only transmit data during their respective *slots*. During *slots* other than their own, nodes typically switch to RX mode. This ensures that no collision can happen in the network, thus saving the need for carrier sensing.

Figure 3 Basic Stop and Wait TDMA MAC



- Clock synchronization is mandatory and plays a key role in the effectiveness of TDMA MAC. In our design, the clock synchronization is achieved by a **Piggyback-on-ACKs** time synchronization scheme.
- An arbitrary amount of *guard time* can be padded at the front and back of individual slot times to counter the effects of clock-drift resulting from the imperfection of the network time synchronization scheme. The more accurate the time synchronization scheme is, the less the amount of guard time needed.

Acknowledgement Schemes for TDMA MAC

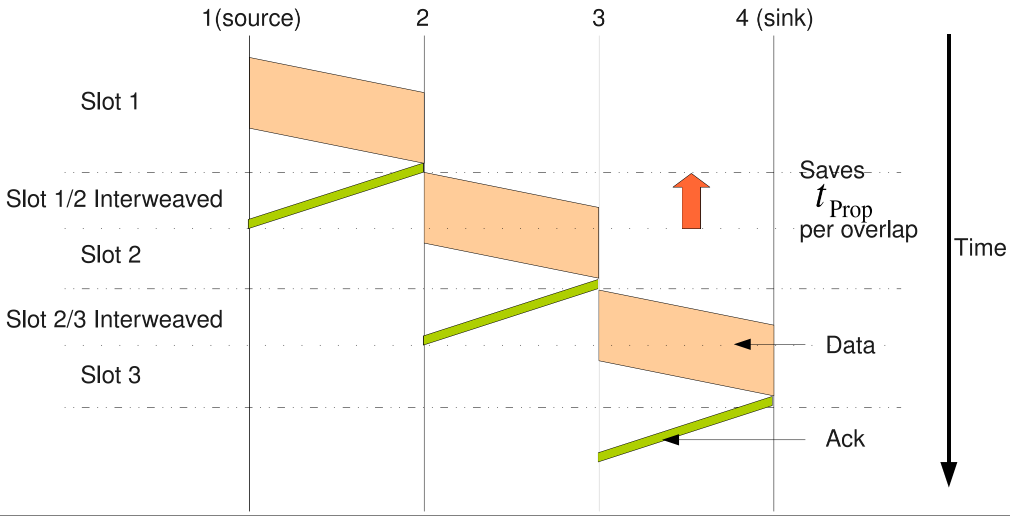
Due to the inherently adverse channel conditions of the underwater medium, the probability of successively receiving more than one error-free packet is low. Therefore, the application of cumulative acknowledgements which is commonly used in high-speed networks is not suitable. We thus propose the use of a Stop and Wait (S&W) acknowledgement scheme to mitigate the effects of the adverse underwater channel.

The use of explicit acknowledgements, together with ARQ (retransmission scheme) implemented at the Opportunistic Multihop ARQ functional block, provides reliable data transfer for the DLL. The two proposed S&W TDMA MAC schemes for this project: **Basic S&W TDMA MAC** and the **Interweaved S&W TDMA MAC** will be explored as follows.

Basic Stop & Wait TDMA MAC

Figure 3 is a timing sequence diagram showing the delivery of a single DATA packet over a 3-hop linearly arranged network using Basic S&W TDMA MAC.

Figure 4 Interweaved Stop and Wait TDMA MAC.



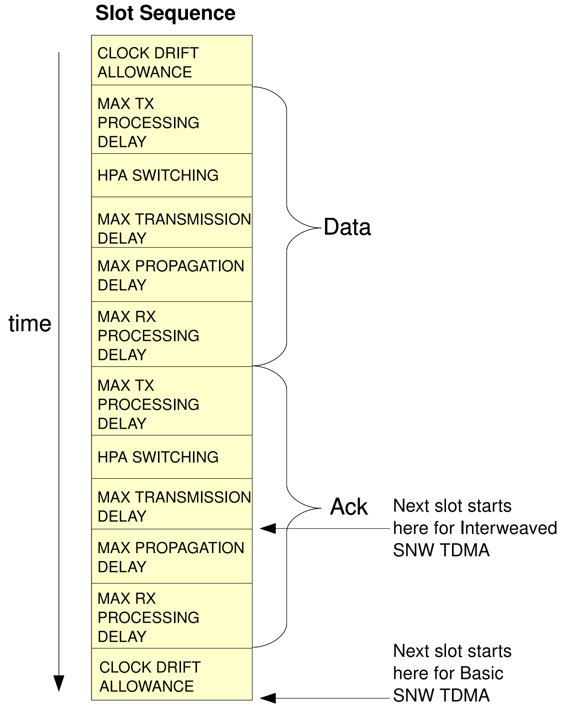
The Basic S&W TDMA MAC protocol has areas for improvements in terms of making better use of resources. Careful analysis of the timing sequence diagram for methodologies to improve performance in terms of throughput and end-to-end delay shows that certain optimization can be done by "interweaving" *slots* without introducing collisions. This is explained in the following section.

Interweaved Stop & Wait TDMA MAC

The modified version of the Basic S&W TDMA MAC scheme is named the Interweaved S&W TDMA MAC due to the slight overlapping, or interweaving, of neighboring nodes' *slots* as illustrated in the timing sequence diagram in Figure 4. Figure 4 shows how a single DATA packet is delivered over a 3-hop linearly arranged network. Some key points relating to Interweaved S&W TDMA MAC:

- Figure 4 illustrates that the *slot* times can be interweaved safely without issues of collisions.
- After acknowledgement has been transmitted for a successfully received DATA packet, a node immediately switches to TX mode and transmits the DATA packet to the next node of the multihop network. This is the key difference between the Interweaved S&W TDMA MAC and the Basic S&W TDMA MAC. In the Basic S&W TDMA MAC, a node waits for the acknowledgement packet to propagate and be received by the previous node before commencing transmission.
- The acknowledgement of a DATA packet is only expected to be received in the next *slot*, after the sender's *slot* has ended.

Figure 5 Slot Time Breakdown.



- As compared to the Basic S&W TDMA MAC, each individual *slot* duration is reduced by t_{Prop} due to the interweaving of the *slots*. The savings in the total *frame* duration grows linearly with the size of the network, with an overall reduction of $N * t_{Prop}$ for a N-hop network.

TDMA *Slot* Time

The calculation of *slot* duration is pertinent to the effectivness and efficiency of a TDMA based scheme. Excessive *slot* duration is detrimental to throughput and end-to-end delay, whereas an underestimated *slot* duration might cause a TDMA scheme to fail completely due to multiple access collisions.

Figure 5 shows the breakdown of *slot* time calculation for both Basic S&W TDMA and Interweaved S&W TDMA MAC.

3.2 Time Synchronization Scheme in Upper MAC Layer

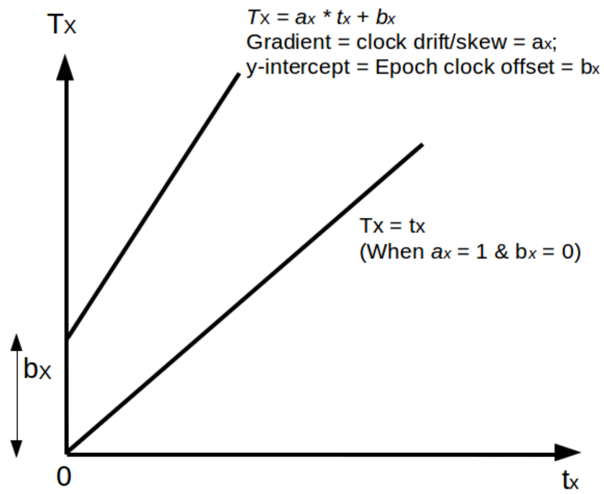
Clock synchronization is mandatory and plays a key role in the effectiveness of TDMA-related MAC protocols. During simulations, time synchronization among nodes is easily taken care of by the simulator. During a field test, however, the clocks of the nodes may drift as time passes, which introduces an unaccountable factor that affects the results of tests that are carried out at different times. One way to minimize the harmful effects is to increase the guard time in each slot, which necessarily decreases the performance of data link layer in terms of throughput.

In order to achieve fairer comparison among tests during a sea trial without introducing additional traffic, a simple time synchronization scheme that uses timestamps piggybacked on ACKs is designed and implemented on the Upper MAC layer as an optional feature for the TDMA-related protocols. Existing time synchronization schemes such as TSHL [7] and MU-Sync [8] require two sequential phases to synchronize a network which may take a long time to complete in a real-world deployment. Our scheme is derived from careful mathematical analysis and only requires a single phase which consists of two concurrent steps. It can quickly synchronize the whole network[1] and maintain the accuracy. The scheme has the following properties:

- When the time synchronization scheme is activated, it will allow every node to be time-synchronized to the sink node. In the 4-node (3-hop) experiments, it means that Node 1, Node 2 and Node 3 will eventually be time-synchronized to Node 4.
- There are two concurrent steps in the time synchronization process: Firstly, Node X estimates the clock difference (current clock offset) between its local time and Node X+1's local time. Secondly, Node X assumes Node X+1 knows the current clock offset between its local time and the sink node's time and expects Node X+1 will tell Node X about this current clock offset. As long as every node 'thinks' in the same way, all of them will eventually be synchronized to the sink node. All nodes perform these two steps at the same time, so the scheme is scalable.
- Every node keeps two version of clocks: 1. System time (local time), which is never changed and facilitates the process of network time synchronization; 2. Synchronized time, based on which the data delivery scheme arranges all its timers.
- Only the active nodes will be time-synchronized.
- The only assumption in this scheme is delay symmetry, which means that the time taken from the instant Node X sends a data packet to Node X+1 to the instant Node X+1 receives the packet equals to the time taken from the instant Node X+1 sends an ACK packet to Node X to the instant Node X receives the packet. This assumption makes the scheme very simple but is the main source of the estimation error.
- For a linear 4-node network, the clocks in the whole network can be time-synchronized and maintained at an accuracy of about 200 ms in the real-world implementation, as shown in lab tests.

[1] With the current implementation, the fastest it can achieve time synchronization in the linear 4-node network is 17 TDMA frames.

Figure 6 Clock skew and Epoch clock offset

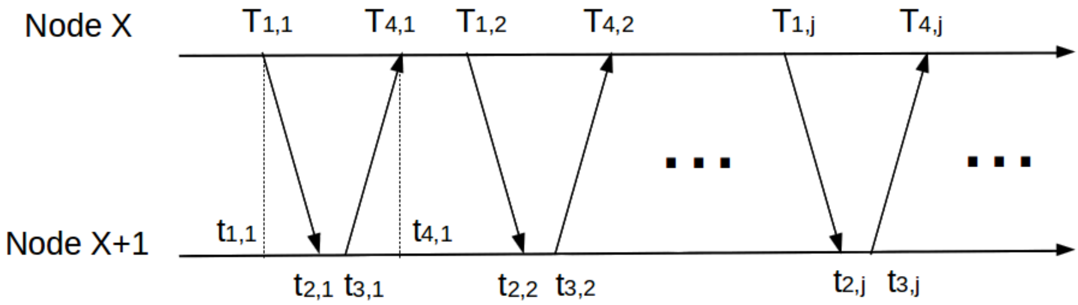


Notations

There are some notations that will be used in the analysis, which are listed as follows:

- a_X: the clock drift (or clock skew) of Node X relative to Node X+1's clock.
- b_X: the clock offset of Node X relative to Node X+1's clock at time $= 0$ (Epoch). When using C programming to get the current time, the return value is the time elapsed since 00:00:00 GMT, Jan. 1, 1970, which is effectively the Epoch in this case. The basic assumption is that $T_X = a_X t_X + b_X$ (Figure 6), where T_X is Node X's local time, and t_X is the local time of Node X+1 when Node X's clock is right at T_X.
- $b_{X,X+1}$: the current clock offset of Node X relative to Node X+1.
- $b_{X+1,S}$: the current clock offset of Node X+1 relative to the sink node.
- $\hat{z}$: the estimated value of the true parameter z. The population parameter z is a constant, but $\hat{z}$ is a variable that is estimated from a sample, and $\hat{z}$ follows a certain distribution (assume normal distribution unless otherwise stated). For example, $\hat{a}_X$ means the estimated value of a_X, the clock skew of Node X relative to Node X+1.

Figure 7 Timestamps made during packet exchanges between two neighbouring nodes



3.2.1 Time Synchronization between Two Neighbouring Nodes

Figure 7 shows how two neighbouring nodes (Node X and Node X+1) exchange packets. When Node X sends a generic DATA frame to Node X+1, Node X marks its local time of sending this frame as $T_{1,j}$ (Right at this moment, the local time of Node X+1 is $t_{1,j}$). When Node X+1 receives this frame, it records its local time as $t_{2,j}$ and when it is ready to send the ACK frame, it records its local time as $t_{3,j}$ and puts both $t_{2,j}$ and $t_{3,j}$ (as well as its estimate about the current clock offset between its local time and sink node's time) into the ACK frame and sends it to Node X. Node X then stores the local time $T_{4,j}$ when it receives the ACK frame (Right at this moment, the local time of Node X+1 is $t_{4,j}$). Node X now has one data point j: ($T_{1,j}$, $t_{2,j}$, $t_{3,j}$, $T_{4,j}$). As Node X sends more data to Node X+1 and receives the corresponding ACKs, Node X will accumulate more data points. When the number of data points reaches a certain value (15 during our field tests), Node X will estimate the current clock offset between its local time and Node X+1's local time, through estimating its the clock drift (or clock skew) and the EPOCH clock offset relative to Node X+1's clock.

Focusing on just one data point j ($T_{1,j}$, $t_{2,j}$, $t_{3,j}$, $T_{4,j}$), the relations among these four timestamps can be derived from the following four equations:

$$T_{1,j} = a_X t_{1,j} + b_X \tag{1}$$

$$T_{4,j} = a_X t_{4,j} + b_X \tag{2}$$

$$t_{1,j} = t_{2,j} - \Delta_{X,j} \tag{3}$$

$$t_{4,j} = t_{3,j} + \Delta_{X+1,j} \tag{4}$$

where $\Delta_{X,j}$ is the total delay (measured in Node X+1's time scale) between the instant when the timestamp $T_{1,j}$ is made in Node X and the instant when the timestamp

$t_{2,j}$ is made in Node X+1, and $\Delta_{X+1,j}$ is the total delay (again, measured in Node X+1's time scale) between the instant when the timestamp $t_{3,j}$ is made in Node X+1 and the instant when the timestamp $T_{4,j}$ is made in Node X

Substitute (3) to (1) and (4) to (2), we have

$$T_{1,j} = a_X t_{2,j} - a_X \Delta_{X,j} + b_X \tag{5}$$

and

$$T_{4,j} = a_X t_{3,j} + a_X \Delta_{X+1,i} + b_X \tag{6}$$

(5) + (6) will be

$$T_{1,j} + T_{4,j} = a_X(t_{2,j} + t_{3,j}) + 2b_X + a_X(\Delta_{X,j} - \Delta_{X+1,j})$$

or

$$T_{1+4,j} = a_X t_{2+3,j} + 2b_X + a_X(\Delta_{X,j} - \Delta_{X+1,j}) \tag{7}$$

Equation 7 forms a simple linear regression. Assuming that the error term $(\Delta_{X,j} - \Delta_{X+1,j})$ follows a normal distribution with zero mean, has finite variance and is uncorrelated with the regressor $t_{2+3,j}$[2], a_X and b_X can be estimated by ordinary least squares (OLS) with data points ($T_{1+4,j}$, $t_{2+3,j}$), as shown in Equation 8 and Equation 9. Whether the assumption on the error term $(\Delta_{X,j} - \Delta_{X+1,j})$ is valid will be discussed.

$$\hat{a}_X = \frac{\sum(t_{2+3,j}T_{1+4,j}) - \frac{1}{n}\sum t_{2+3,j}\sum T_{1+4,j}}{\sum t^2_{2+3,j} - \frac{1}{n}(\sum t_{2+3,j})^2} \tag{8}$$

$$\hat{b}_X = \frac{1}{2}\left(\frac{\sum T_{1+4,j}}{n} - \hat{a}_X \frac{\sum t_{2+3,j}}{n}\right) \tag{9}$$

where n is the total number of data points collected by Node X.

Once Node X obtains $\hat{a}_X$ and $\hat{b}_X$, it can estimate its current clock offset relative to Node X+1's clock by Equation 10:

$$\hat{b}_{X,X+1} = T_X - (T_X - \hat{b}_X)/\hat{a}_X \tag{10}$$

[2]To make the discussion easier, a stricter assumption that $\Delta_{X,j} - \Delta_{X+1,j} = 0$ is adopted.

3.2.2 Time Synchronization in the Whole Network

As Node X+1 piggybacks its estimated current clock offset relative to the sink node's clock ($\hat{b}_{X+1,S}$) on every ACK that it sends to Node X, Node X now can synchronize itself to the sink node by

$$\hat{S}(T_X) = T_X - (\hat{b}_{X,X+1} + \hat{b}_{X+1,s}) = T_X - \hat{b}_{X,s} \tag{11}$$

But how does Node X+1 estimate $\hat{b}_{X+1,S}$? It does exactly the same thing as Node X does, i.e. it only estimates its current clock offset relative to Node X+2's clock and expects Node X+2 to piggyback its estimated current clock offset relative to the sink node's clock ($\hat{b}_{X+2,S}$) on every ACK. Therefore, the value of $\hat{b}_{X+1,S}$ Node X+1 piggybacks on its ACKs to Node X is actually $\hat{b}_{X+1,X+2} + \hat{b}_{X+2,S}$.[3] Even the node that is closest to the sink node does the same thing. It estimates its current clock offset relative to the sink node's clock, and the sink node simply sets its own 'estimated' current clock offset relative to its own clock as 0 and puts it in the ACKs.

3.2.3 Is the Assumption $\Delta_{X,j} - \Delta_{X+1,j} = 0$ Valid?

Based on [9] and [10], $\Delta_{X,j}$ can be seen as the summation of access time in Node X ($A_{X,j}$), interrupt handling time in Node X ($I_{X,t,j}$), encoding time in node x ($E_{X,j}$), transmission time in Node X ($T_{X,j}$), propagation time from Node X to Node X+1 ($D_{X,j}$), reception time in Node X+1 ($R_{X+1,j}$), byte alignment time in Node X+1 ($B_{X+1,j}$), interrupt handling time in Node X+1 ($I_{X+1,r,j}$), decoding time in Node X+1 ($C_{X+1,j}$), and any other delay that is not included ($\delta_{X,j}$), i.e.

$$\begin{aligned}\Delta_{X,j} &= A_{X,j} + I_{X,t,j} + E_{X,j} + T_{X,j} + D_{X,j} \\ &\quad + R_{X+1,j} + B_{X+1,j} + I_{X+1,r,j} + C_{X+1,j} + \delta_{X,j}\end{aligned} \tag{12}$$

A similar expression for $\Delta_{X+1,i}$ can be obtained:

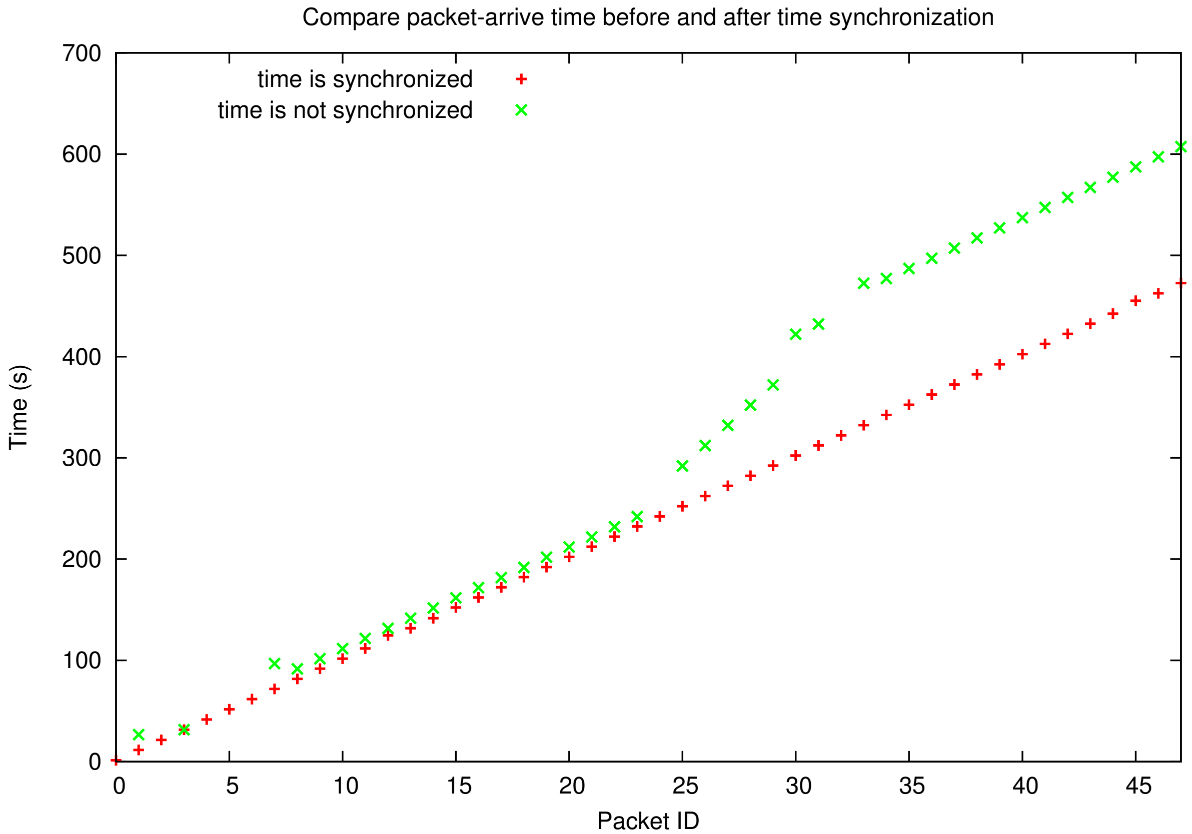
$$\begin{aligned}\Delta_{X+1,j} &= A_{X+1,j} + I_{X+1,t,j} + E_{X+1,j} + T_{X+1,j} + D_{X+1,j} \\ &\quad + R_{X,j} + B_{X,j} + I_{X,r,j} + C_{X,j} + \delta_{X+1,j}\end{aligned} \tag{13}$$

One way to achieve $\Delta_{X,j} - \Delta_{X+1,j} = 0$ is to make the corresponding pairs in equation (12) and (13) equal 0, which implies the need for high symmetry in the various processing time and delays in two nodes during one particular round of packet exchange. However, this is quite unlikely. For example, the transmission time depends on bandwidth and packet length. If Node X and Node X+1 have different bandwidth, the term

[3] The principle is exactly the same as how a recursive function works in programming.

$(T_{X,j} - T_{X+1,j})$ will not be 0. Moreover, the propagation delays $D_{X,j}$ and $D_{X+1,j}$ are not the same if nodes are moving during a particular round of message exchanges. As the problem of multipath in the underwater communication is common, a DATA packet and the corresponding ACK packet may not take the same routes, causing the values of $D_{X,j}$ and $D_{X+1,j}$ to further deviate. Therefore, more often than not, $\Delta_{X,j}$ does not equal $\Delta_{X+1,j}$, and in order to improve the accuracy of this time synchronization scheme in the future, the most crucial work is to make the assumption as valid as possible.

Figure 8 Plot of Packet arrival time vs packet ID before and after time-sync (BIDO Int. S&W TDMA)



3.2.4 Implementation and Performance Evaluation

From the perspective of the Upper MAC Layer, all our TDMA-related protocols acknowledge a packet as soon as they receive the packet. As all the timestamps used in the time synchronization scheme are created on the Upper MAC Layer, $t_{2,j}$ and $t_{3,j}$ are regarded as the same. Therefore, Node X+1 only has to piggyback $t_{2,j}$ (6 bytes) and its estimated current clock offset relative to the sink node's clock (5 bytes) on the ACKs, which occupy the previously paddled space in the ACKs and thus do not introduce any extra overhead.

Figure 8 displays the results obtained during a sea trial near St. John's Island on 3rd Oct 2011. There are four nodes and they are arranged roughly in a straight line (more details about the experimental setup will be presented in Section 5.4.1). The transmission power of the nodes is adjusted to be very high such that channel changes will unlikely affect the performance of the tests and the most significant factor that affects the result is whether nodes are time-synchronized. As the results show, the test with time synchronization achieved higher packet delivery rate and lower overall delay to complete sending 47 packets.

3.2.5 Recommendations for Future Development of the Time Synchronization Scheme

- To enhance the accuracy, it is necessary to generate the timestamps on the physical layer, rather than on the Upper MAC layer, to better capture the times of sending and receiving a packet. In this way, the assumption of delay symmetry is more valid.
- Currently only the downstream traffic contributes to the time synchronization of the network. Upstream traffic could potentially be used to enhance the accuracy of the scheme and shorten the time taken to synchronize the whole network.
- With the current scheme, the more upstream a node is, the larger the variances of the estimates will be. For a linear topology, it may be better for a node in the middle rather than at the end of the line to be the reference node.

3.3 Lower Medium Access Control

The operation of the lower MAC is closely tied with the modem operation. The state transition diagram (Figure 9) shows the operation.

The following table (Table 1) enumerates the different states and elaborates on the allowed state transitions:

Figure 9 Lower MAC state transition diagram.

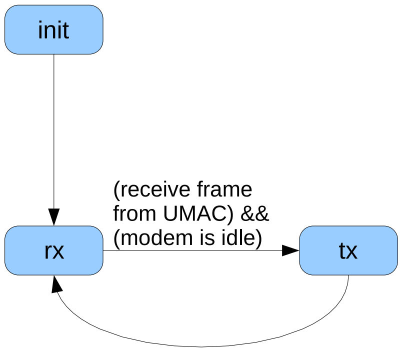


Table 1: Lower MAC States.

State	Name	Details
init	Initialize Mode	Initialize the lower MAC system parameters/configuration settings and perform hardware checks. Performed only once at start-up. After all necessary tasks are completed, state will transition to *rx*. To ensure that transition is complied by hardware, all receive/transmit capabilities are disabled at this state. Frames received from upper MAC are also rejected at this stage.
rx	Receive Mode	Set the modem to receive mode. If modem fails to switch to receive mode after MAX_SETRX_RETRY attempts, the lower MAC returns a system failure error. If a frame from upper MAC is received and the modem is ready to transmit or idle, state transitions to *tx* to transmit frame. If modem is not ready to transmit or busy receiving a frame, the lower MAC remains at this state.
tx	Transmit Mode	Wait to completely transmit a frame. At this state, the modem will reject frames received from upper MAC. At the end of this state, either the frame is completely transmitted by the modem, or a timeout (TX_WAIT_TIMEO) occurred. In any case, the state transitions back to *rx*.

3.4 Modem Driver

From the perspective of the data link layer, the Modem Driver functional block serves as the lowest sub-layer that interacts with the modem directly. With this functional block, the whole data delivery scheme is largely independent of what type of hydro-acoustic modems used, as the functional block alone will be able to handle any change associated with the modems.

Among other minor functions, the Modem Driver functional block provides three main services :

- **Set Parameter Value:** Whenever LMAC requests to set a parameter in the modem to a particular value, the Modem Driver sends an command to the modem to do the task. It then notifies LMAC whether the action succeeds or fails.
- **Send a Packet:** Whenever LMAC requests to send a packet, the Modem Driver sends an command to the modem to complete the task. It then notifies LMAC whether the action succeeds or fails.
- **Receive a Packet:** Modem Driver keeps a separate thread that always accepts anything sent from the modem. As soon as the Modem Driver identifies that it is a proper packet based on the format of the packet (which can still be corrupted inside), it simply passes the packet to the LMAC layer. The processing of the packet is the responsibility of LMAC, rather than of Modem Driver.

4 Data Delivery Scheme

In this section, a brief discussion of the data delivery scheme will be presented. The details of the design of the data delivery scheme including simulation results are presented in Progress Report #2 [2] and some results from the field tests at the marina at the Republic of Singapore Yacht Club with three old modems are also presented in Progress Report #3 [3].

4.1 Opportunistic ARQ with Bidirectional Overhearing

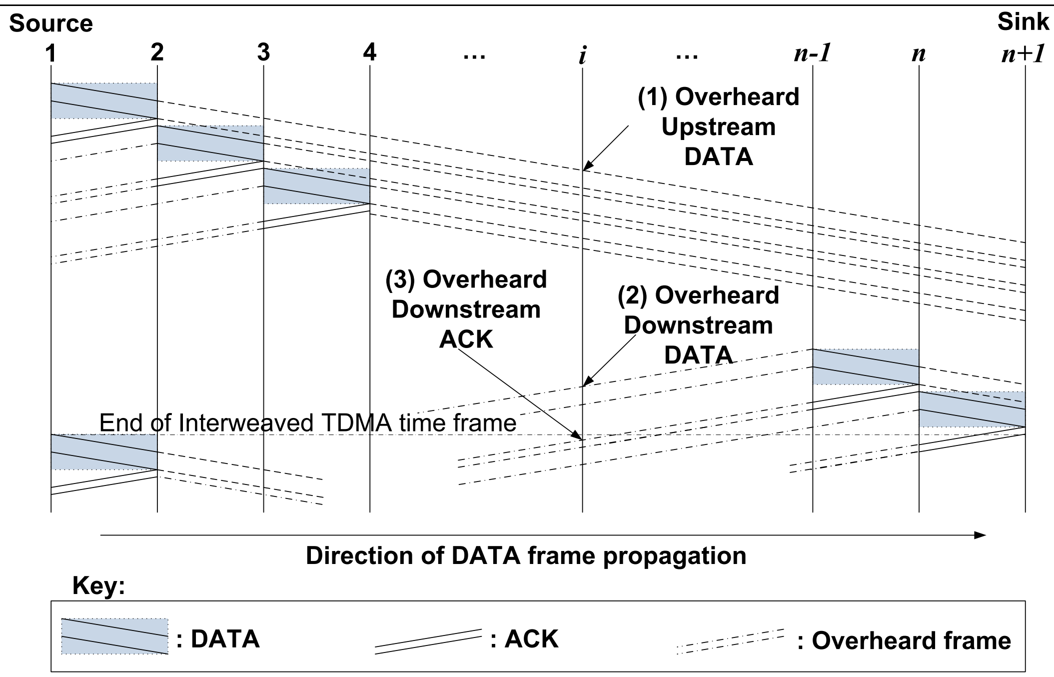
The data delivery scheme proposed in [2], termed the Opportunistic ARQ with Bidirectional Overhearing endeavors to make full use of the broadcast nature and spatial and temporal variance of the underwater acoustic channel to improve overall network performance.

The conventional strictly hop-by-hop propagation of DATA frames through a linearly arranged network is extended with the elements of opportunistic routing and implicit acknowledgement. DATA frames are no longer restricted to following a fixed route to the sink, and there are now more avenues of terminating unnecessary transmissions and re-transmissions. This is done by leveraging on overheard DATA and ACK frames from both directions.

Consider a linearly arranged underwater acoustic network consisting of n+1 nodes as illustrated in Fig. 1. A node i using a strictly hop-by-hop stop-and-wait (S&W) ARQ transmits a DATA frame p to the immediate downstream node i+1. Node i then proceeds to wait for an acknowledgement (ACK) frame from node i+1, necessarily re-transmitting frame p when no explicit ACK was received upon timeout expiry.

In addition to such hop-by-hop, non-opportunistic exchange of DATA and ACK frames, the proposed data delivery scheme enhances network performance by using three types of opportunistically overheard frames. These overheard frames are illustrated in Fig. 10, a sample timing diagram of a network running the proposed data delivery scheme on Interweaved TDMA MAC.

Figure 10 Timing Diagram of Opportunistic Stop and Wait ARQ with Bidirectional Overhearing on Interweaved TDMA MAC. The 3 types of opportunistic packets used by the proposed DDS are annotated: (1) Upstream DATA, (2) Downstream DATA, (3) Downstream ACK.



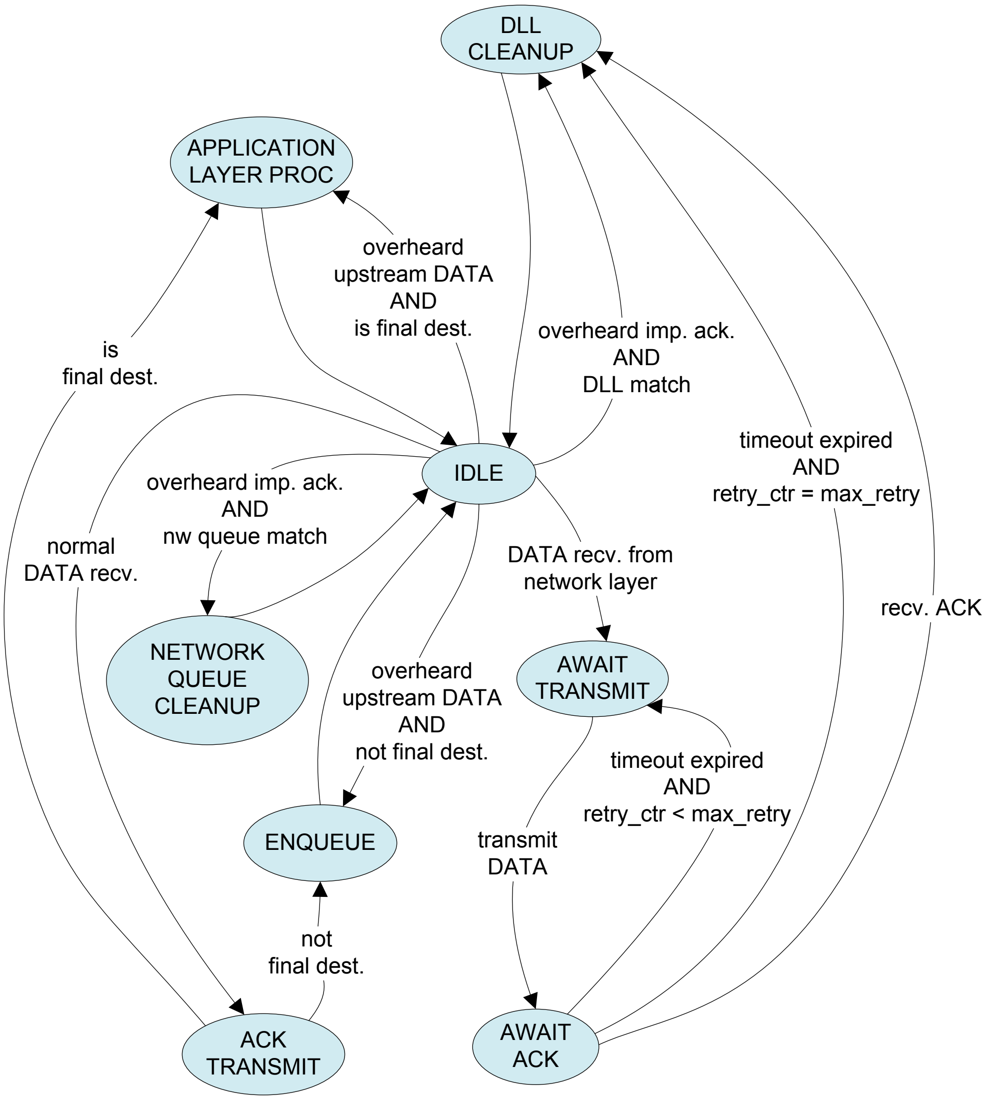
The overhearing of DATA or ACK frames, from any node in the network (both upstream and downstream: bidirectional), are used for either speeding up DATA delivery to the sink or as implicit acknowledgement for a previous DATA transmission. This increases the robustness of a data delivery scheme by reducing dependency on each link for data transmissions. The single points of failure in strictly hop-by-hop transmissions schemes are thus eliminated.

The following section describes the protocol states of the proposed data delivery scheme.

4.2 Protocol State Transitions

Fig. 11 shows the protocol's state transition diagram for a node i of an underwater network as illustrated in Fig. 1. The state diagram is generic, applicable to source, relay and sink nodes. The key protocol transitions are as follows:

Figure 11 Protocol state diagram of the *opportunistic* ARQ with *bidirectional overhearing* data delivery scheme.



- When node i receives a DATA frame from the network layer, it is first buffered in the AWAIT TRANSMIT state. This state is controlled by the medium access control (MAC) protocol. Upon transmission, node i transits to AWAIT ACK and waits for an explicit acknowledgement until timeout expiry, retransmitting when no ACK was received up to *max_retry* times. Finally, i transitions to DLL CLEANUP whereby the data link layer (DLL) buffer is cleared.

- Upon normal receipt of DATA frame p from node i-1, node i will immediately switch to ACK TRANSMIT, sending an explicit ACK back to i-1. If the final destination of p is i, the frame will be passed to the application layer for processing at the APPLICATION LAYER PROC state, else p will be enqueued at the network layer.

- When node i *overhears* a DATA frame p and i is the frame's final destination, p will be passed to the application for processing at the APPLICATION LAYER PROC state, else p will be enqueued at the network layer for relay. No explicit ACK is ever sent for *overheard* DATA frames.

- On *overhearing* an implicit acknowledgement (either downstream DATA or ACK), node i will proceed to check for any matches in the DLL or network layer queue. A match will result in node i transiting to DLL CLEAN UP or NETWORK QUEUE CLEANUP respectively for the removal of the matching DATA frame.

It is important to note that the state transition diagram contains the key intelligence of the proposed data delivery scheme, which essentially are the rules on handling different types of overheard frames. The state transition diagram is, however, independent of the medium access control (MAC) protocol used by the nodes in the underwater network, and should function (albeit with varying performance) over different MAC protocols.

5 Integration and Field Testing with ARL Hydroacoustic Modems

After successful simulation and implementation of the data link layer (DDL) and data delivery scheme (DDS), the next steps that have been conducted are to iteratively perform (i) laboratory tests; (ii) field tests in "controlled" environment and (iii) field tests in open sea to fulfil the following objectives:

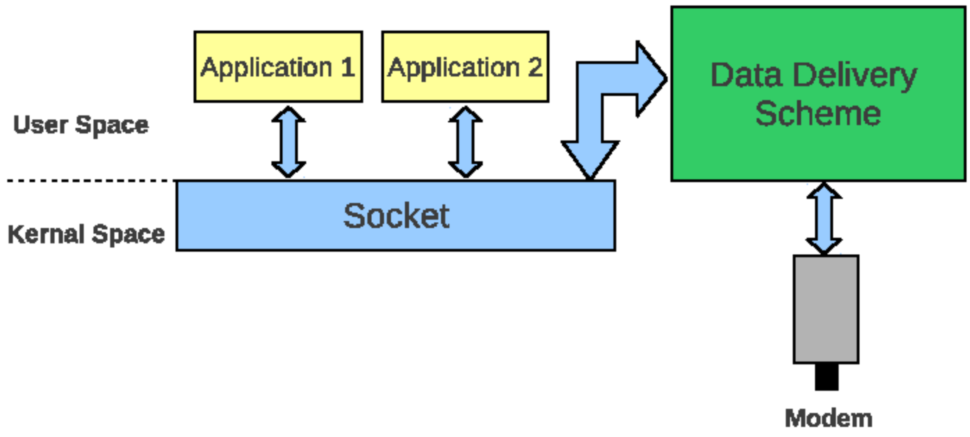
- To validate the integration efforts of the networking stack, in particular the proposed DLL and DDS, on the underwater acoustic modems;

- To test and validate the DDL and DDS design on the underwater acoustic modems over a linear multihop setup (DATA packet delivery over more than a single-hop).

5.1 Implementation Architecture

The network architecture shown in Figure 2 is essentially similar to the TCP/IP stack. For the integration process carried out in this project, an approach called "pure user implementation architecture" is employed to integrate the proposed data delivery scheme

Figure 12 Pure user implementation architecture.



with the hydroacoustic modems, in which the entire stack is implemented as a monolithic user-space application (Figure 12).

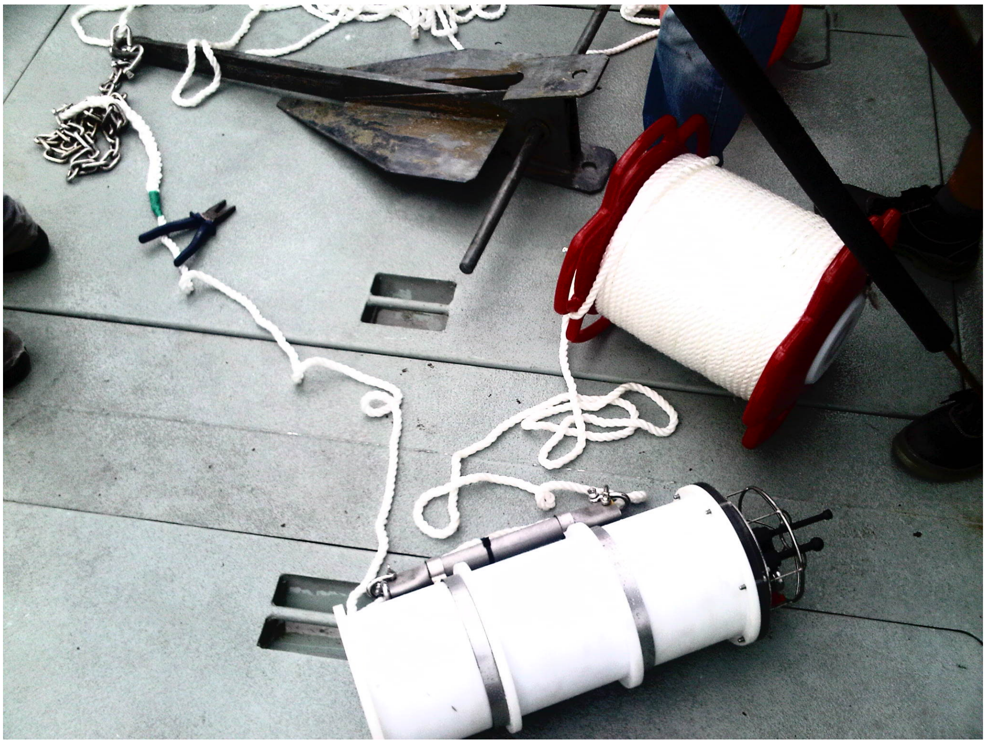
The approach of "pure user implementation" is contrast to another approach called "split kernel-user implementation", in which the transport and network layers are implemented inside the operating system kernel while the data link layer is implemented as a user-space application. The approach of "pure user implementation" is preferred because of its higher flexibility in its implementation. A more detailed discussion of both approaches can be found in Deliverable 2 [2].

5.2 ARL Hydroacoustic Modems

The ARL modems are developed for point-to-point communication for medium range links in warm shallow waters. The communication technique employs OFDM to divide the available bandwidth into multiple sub-carriers, each of which makes use of differential QPSK. By using block interleaved FEC coding across all sub-carriers and multiple symbols, the modems are robust against both frequency-selective fading and transient events such as snaps made by snapping shrimp which is the main source of noise in Singapore waters. [11]

The experiments carried out for this report employ the new modems which are delivered for this project from ARL, while those carried out for Deliverable 1 to Deliverable 3 did the old modems. The hardware used by the new modems has more computational capacity and therefore is able to support a larger signal buffer, stronger FEC codes etc. Although the new modems have a different interface, due to the high modularity in the

Figure 13 The PANDA modem, the surface buoy with rope, and the anchor



design of the network architecture, the only functional block that has to be modified in order to work with the new modems is the Modem Driver module.

The number of modems that are used in the field tests from Sep 2011 to Oct 2011 is four. Two of them are specially designed for Pop-up Ambient Noise Data Acquisition (PANDA) (Figure 13) [12], which have internal batteries. The other two are surface modems (Figure 14) which have to be powered by external batteries. Due to these characteristics and the fact that only one boat is available, the two PANDAs are deployed as relay nodes (i.e. Node 2 and Node 3) during sea trials. The PANDAs automatically run the data delivery scheme when they are powered up, and the two surface modems are deployed as end nodes (Node 1 and Node 4), which can be directly accessed through netbooks.

Controlling the Surface Modems

The data delivery scheme runs inside a netbook, which connects to a surface modem through an Ethernet cable. Modem Driver, the lowest layer in the data delivery scheme, uses a socket interface to interact with the modem. Whenever configuration of parameters in the data delivery scheme is required, the program is terminated and then the configuration file is modified manually and finally the program is re-run again. The

Figure 14 The surface modem

whole process is very straightforward.

Controlling the PANDA Modems

As the PANDAs are anchored and floating near to the sea bed and thus cannot directly be accessed during the sea trials, remote control has to be made possible in order to perform configuration of parameters whenever needed. This is achieved by a shell script. The script runs the data delivery scheme (which is cross-compiled by arm-linux-gcc) inside the PANDA. The data delivery scheme controls the modem through a socket interface, which is exactly the same as in the case of surface modems. The shell script also runs a small application that connects to the data delivery scheme through another socket interface (see Figure 2). Whenever there are packets destined for this node, the data delivery scheme passes the packets to this application which will check the content in every packet. When the content in the packet is to request a change in the value of a parameter, the shell script terminates the data delivery scheme, changes the configuration file accordingly, and restarts the data delivery scheme. When the change is successful, the shell script sends an acknowledge packet via the data delivery scheme to two end nodes.

The parameters that can be remotely be changed include MAC Protocol, number of maximum retransmissions, transmission attenuation level, MAC address, among others.

Prior to conducting sea trials, the shell script is tested many times in the lab as well as at the marina at Republic of Singapore Yacht Club (RSYC) to ensure it works well, as any flaw may mar the whole sea trial.

5.3 Laboratory Testing and Field Testing at RSYC

Adopting the practice of incremental testing, tests in a water tank (about 3m x 3m) at NUS-ARL are performed to ensure that both hardware and software are in working condition prior to outdoor testing.

Prior to conducting sea trials in Sep/Oct, tests were carried out on 14^{th} September 2011 at the marina of the Republic of Singapore Yacht Club (RSYC). The channel at RSYC is usually quite stable, though the heavy marine traffic may affect the performance of the modems. However, restrictions in the setup locations at the marina limit the inter-nodal distance to be about 30 meters. With such a short inter-nodal distance, the Packet Delivery Ratios from all tests on that day were close to 100%, which are not representative of the real-world scenario. Nonetheless, the tests at RSYC show that both the software and hardware are working well.

All of the subsequent analysis is based on the results harvested from the sea trials, which are closer to the real-world scenario.

5.4 Sea Trials

5.4.1 Sea Trial Setup

As mentioned in Section 5.2, the topology for the sea trials comprises four modems arranged approximately linearly as shown in Figure 15. Node 1 and Node 4 are surface modems, which can be directly controlled through netbooks. Node 2 and Node 3 are housed in the PANDAs, which are controlled by Node 1 and Node 4 remotely. Due to the restriction in the approved area to do experiments, the end-to-end distance is only about 340 meters and the internode distance is about 100m to 130m. As shown in Table 2, the depth of immersion is about 10 to 13 meters under the surface of the water.

Node 1 to Node 4 are set up one by one. Node 1 is set up at the jetty. After the setup, one person stays at the jetty to control Node 1. Node 2 is tied to a rope. The other end of the rope is anchored to the sea bottom and the modem is floating roughly vertically. When Node 2 is ready, Node 1 sends some packets to Node 2 and checks whether it receives ACKs from Node 2 to make sure Node 2 works. The same procedures are carried out with Node 3. The boat finally stays beside Node 4 which can be accessed directly. Before commencing the tests, the four nodes are time-synchronized using the implemented Piggyback-on-ACKs time synchronization scheme as described in Section

3.2. During the tests, the time synchronization scheme is always in active mode, which synchronizes the whole network continuously. Throughout all tests, the MAC-layer packet size (for both DATA packet and ACK packet) is 27 bytes.

Figure 15 Locations of the nodes

Node ID	Latitude	Longitude	Depth under the water surface
1	01 13.021 North	103 51.115 East	9 m
2	01 13.029 North	103 51.169 East	12 m
3	01 13.020 North	103 51.228 East	13 m
4	01 13.005 North	103 51.297 East	13 m

Table 2: GPS Coordinates of the locations of the nodes

5.4.2 Experimental Testplan

Table 3 lays out the test plan for the sea trials. The results from carrying out this test plan will facilitate the comparison between the two MAC protocols (Basic S&W TDMA and Int. TDMA), the evaluation of retransmissions on the performance, and the comparison between the proposed bidirectional overhearing opportunistic routing scheme and the static routing scheme.

Test Plan			
No.	**Scheme**	**Packets**	**Max. Retrans**
1	BIDO Int. TDMA	100	2
2	BIDO Int. TDMA	100	0
3	Int. TDMA	100	2
4	Int. TDMA	100	0
5	Basic S&W TDMA	100	2
6	Basic S&W TDMA	100	0
7	BIDO Basic S&W TDMA	100	2
8	BIDO Basic S&W TDMA	100	0

Table 3: Testplan Overview

5.4.3 Experimental Results

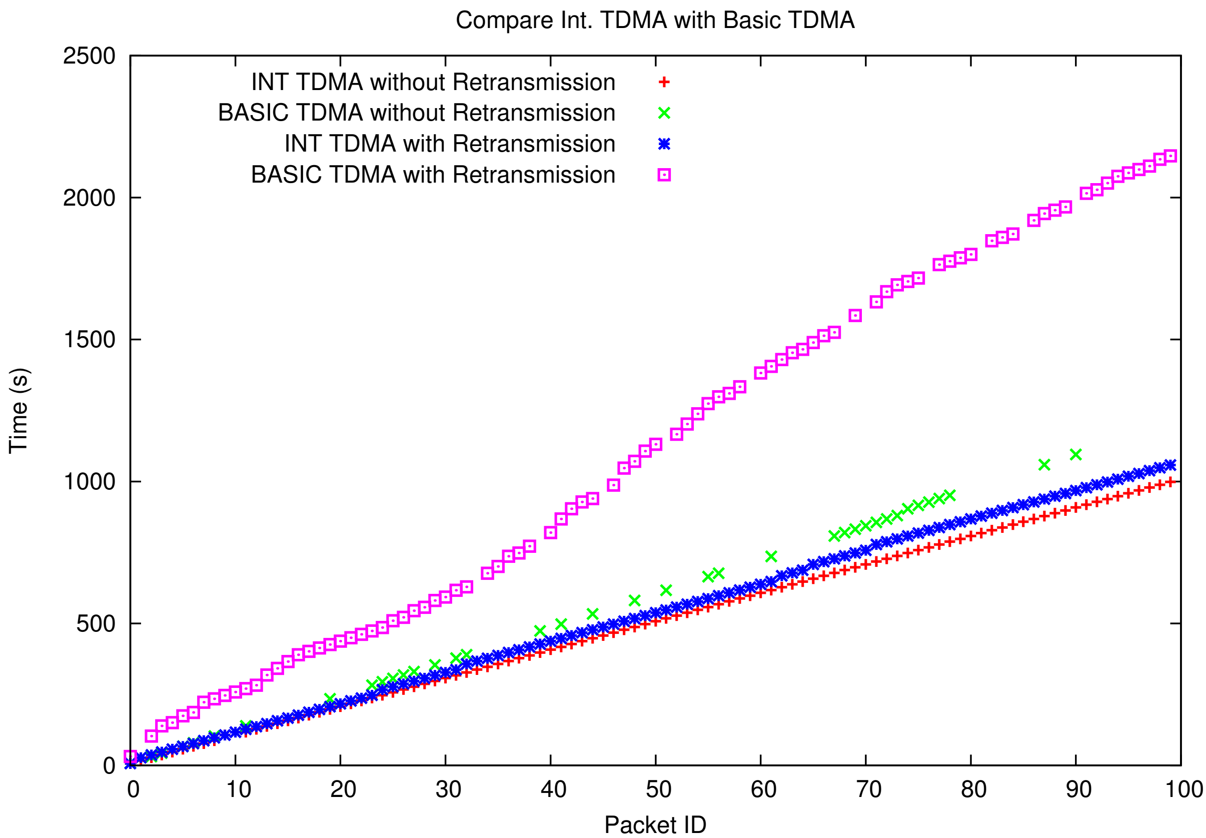
Table 4 summarizes the results obtained from the sea trials at the sea area around St John Island on 19^{th} Sep 2011 and 03^{rd} Oct 2011. Together with the plots of packet-received time vs packet ID, the analysis of the results will be done as follows.

Experimental Results						
No.	**Scheme**	**Max. Retrans**	**PDR**	**Throughput** (bps)	**Network Energy Consumption per Pkt**	**Date**
1	BIDO Int. TDMA	2	1.00	21.4	14.5	03^{rd} Oct 11
2	BIDO Int. TDMA	0	0.92	19.9	12.9	19^{th} Sep 11
3	Int. TDMA	2	1.00	21.4	14.4	03^{rd} Oct 11
	Int. TDMA	2	1.00	20.4	14.2	19^{th} Sep 11
4	Int. TDMA	0	0.99	21.4	14.3	19^{th} Sep 11
5	Basic S&W TDMA	2	0.88	8.98	15.9	19^{th} Sep 11
6	Basic S&W TDMA	0	0.38	6.85	22.0	19^{th} Sep 11
7	BIDO Basic S&W TDMA	2	-	-	-	-
8	BIDO Basic S&W TDMA	0	0.96	17.4	12.5	19^{th} Sep 11

Table 4: Experimental Results obtained from Sea Trials

Comparing Int. TDMA and Basic S&W TDMA

Figure 16 Plot of Packet-arrival time vs packet ID across tests with Int. S&W TDMA and Basic S&W TDMA



- **Packet Delivery Ratio**:
  From Table 4, it can be seen that Int. TDMA is much better than Basic S&W TDMA in terms of Packet Delivery Ratio (PDR) when retransmissions are not allowed (99% vs 38%). When up to two retransmissions are allowed for every packet,, the gap between the PDR of Int. TDMA (100%) and the PDR of Basic S&W TDMA (88%) narrows down.

  As explained in Progress Report #1 [1], their PDR performances are expected to be similar when other factors stay the same. Therefore, the reason that Int. TDMA has a better PDR than Basic S&W TDMA is more likely that the channel condition is much better when the test with Int. TDMA is carried out.

- **Throughput**:
  As the nodes' slots overlap slightly with Int. TDMA, the frame duration is shorter than Basic S&W TDMA with the same set of parameters. Therefore, even with similar PDRs, the throughput of Int. TDMA is usually better than that of Basic S&W TDMA as the overall delay of sending 100 packets is normally shorter for Int. TDMA. This can be seen from the comparison between the gradient of the graph

for "Int. TDMA without Retransmission" and that for "Basic S&W TDMA without Retransmission" in Figure 20, with the latter being steeper than the former.

With shorter delays and higher PDR, the throughput achieved by Int. TDMA is much better than that achieved by the basic version. Based on throughput alone, Int. TDMA is a better choice as a MAC protocol compared to Basic S&W TDMA.

- **Network Energy Consumption per Successfully Delivered Packet**:
  Before delving into analysis of network energy consumption between the different schemes, we should first understand how the values in Table 4 are obtained. As per Progress Report #2 [2], a simple energy model is adopted. Each time a node transmits a DATA or ACK frame, the amount of energy units consumed is equal to the length of the frame. The receipt of any frame, on the other hand, consumes the number of units equal to half the frame length. In this case, since the DATA frame and ACK frame share the same length, an even simpler energy model is adopted:

  – The transmission of a DATA or ACK frame consumes 1 unit of energy.
  – The receipt or overhearing of an uncorrupted DATA or ACK frame consumes 0.5 unit of energy[4].

  Channel condition has two opposite effects on this metric. When the channel is good, the number of successfully delivered packets increases, which tends to lower the network energy consumption per successfully delivered packet. However, there may be a lot of overhearing of uncorrupted DATA or ACK packets that would lead to further processing when the channel is good, which consumes energy and tends to increase the value of the metric.

  Table 4 shows that regardless of the number of retransmissions allowed, Int. TDMA results in a lower network energy consumption per successfully delivered packet. This means that the positive effect brought about by a higher PDR triumphs over the negative effect brought about by a higher rate of overhearing on the metric in these tests.

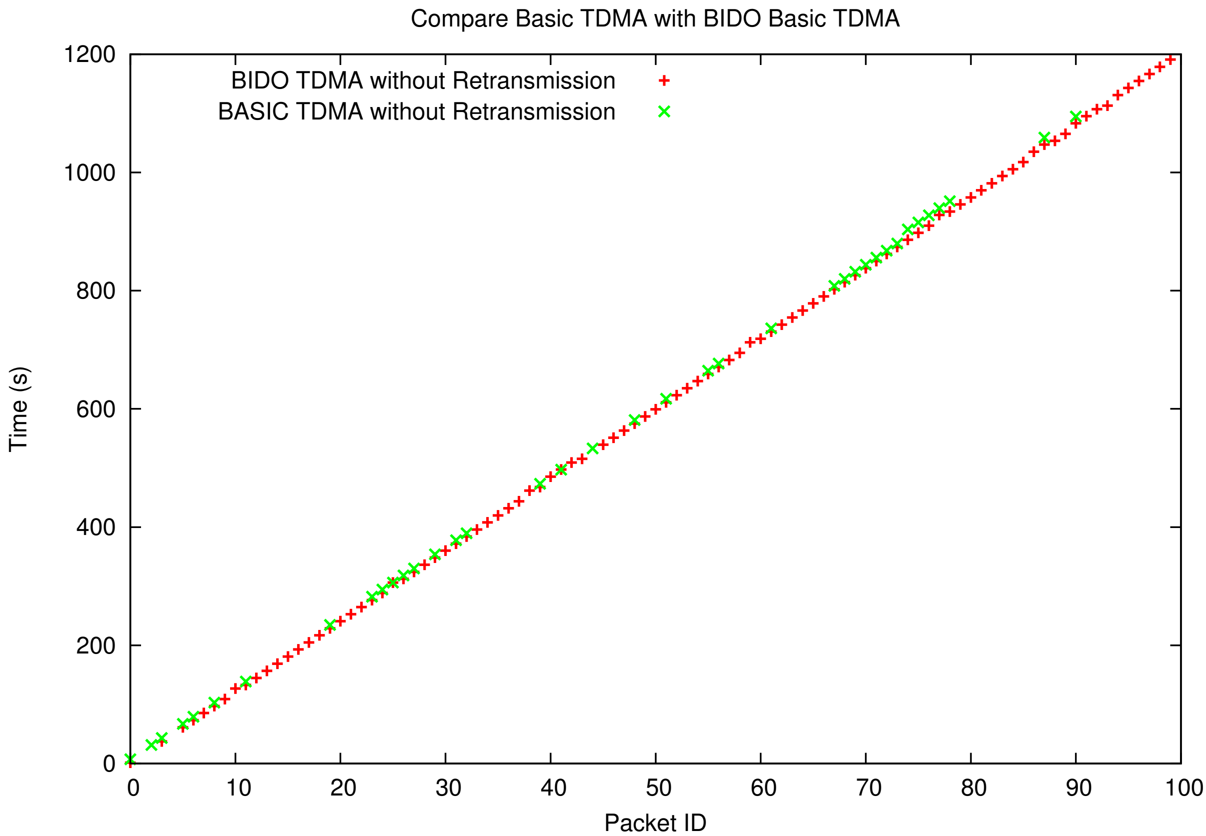
- **Effect of Retransmissions**:
  Retransmissions increase PDR, but not necessarily throughput. When the channel is good, retransmissions may decrease throughput instead. For example, the test with Int. TDMA with Retransmissions has a throughput of 20.4 bps, while the test with Int. TDMA without Retransmissions has a throughput of 21.4 bps, based on the results obtained from the sea trial on $19^{th} Sep 2011$. One reason is that, when the channel is good, ACKs may still be lost and some unnecessary retransmissions happen as a result. When PDR is really bad for some reasons, retransmissions can

[4]The receipt or overhearing of a corrupted DATA or ACK frame consumes energy, too. For simplicity here we assume the energy consumption in this case is 0, as corrupted packets are dropped by the data delivery scheme as early as possible and require no further processing.

increase both PDR and throughput. For example, the test with Basic S&W TDMA with Retransmissions has better PDR and throughput (PDR = 88%, and Throughput = 8.98 bps) than the test with Basic S&W TDMA without Retransmissions (PDR = 38%, and Throughput = 6.85 bps). There is a trade-off between PDR and the total delay in a test. To obtain the optimal number of retransmissions (as well as the optimal values of other parameters such as transmission power) given a situation is one of potential future studies.

Comparing Static Routing with Opportunistic Routing (Basic S&W TDMA)

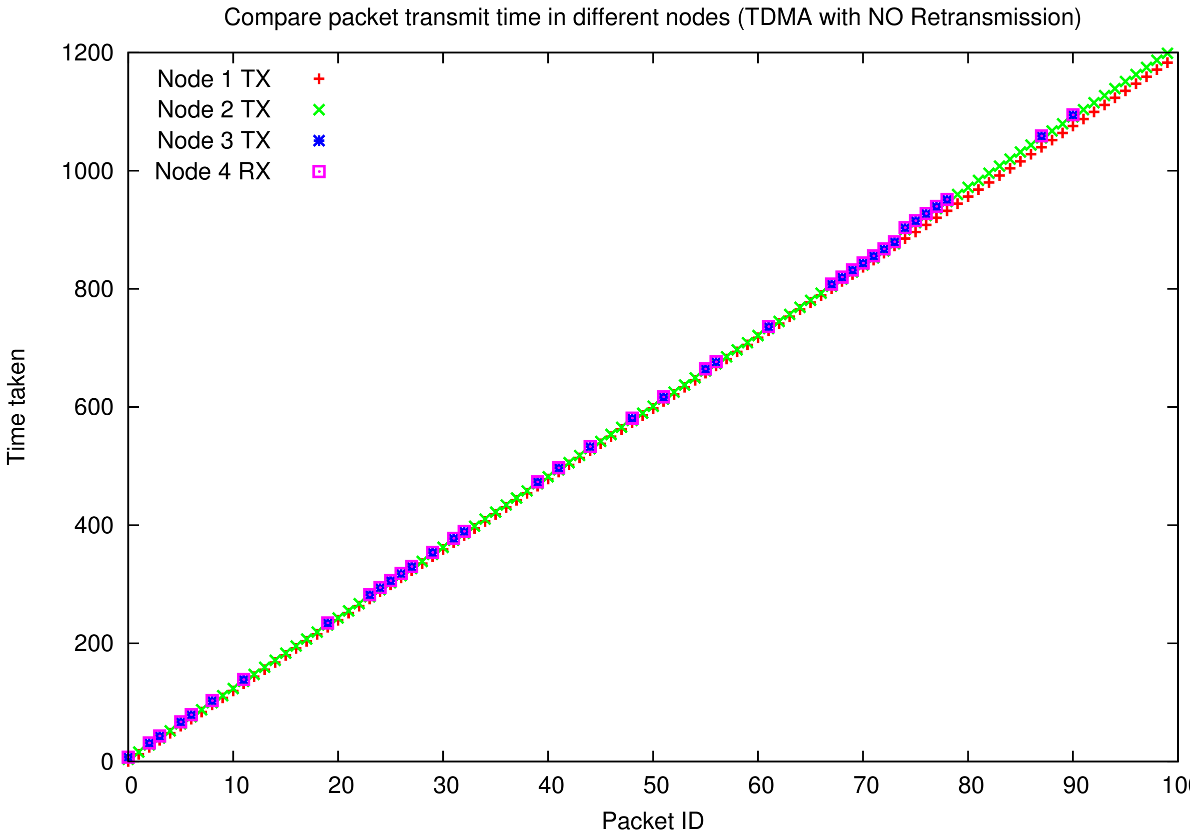
Figure 17 Plot of Packet-arrival time vs packet ID across tests with Basic S&W TDMA and BIDO Basic S&W TDMA



- **Packet Delivery Ratio**:

Table 4 and Figure 17 show that BIDO Basic S&W TDMA performs better than Basic S&W TDMA when retransmissions are not allowed. The reason is that when some of the links deteriorate, BIDO TDMA allows some packets to skip over the

Figure 18 Plot of Packet-transmit time vs packet ID for the Test with Basic S&W TDMA with no retransmission.

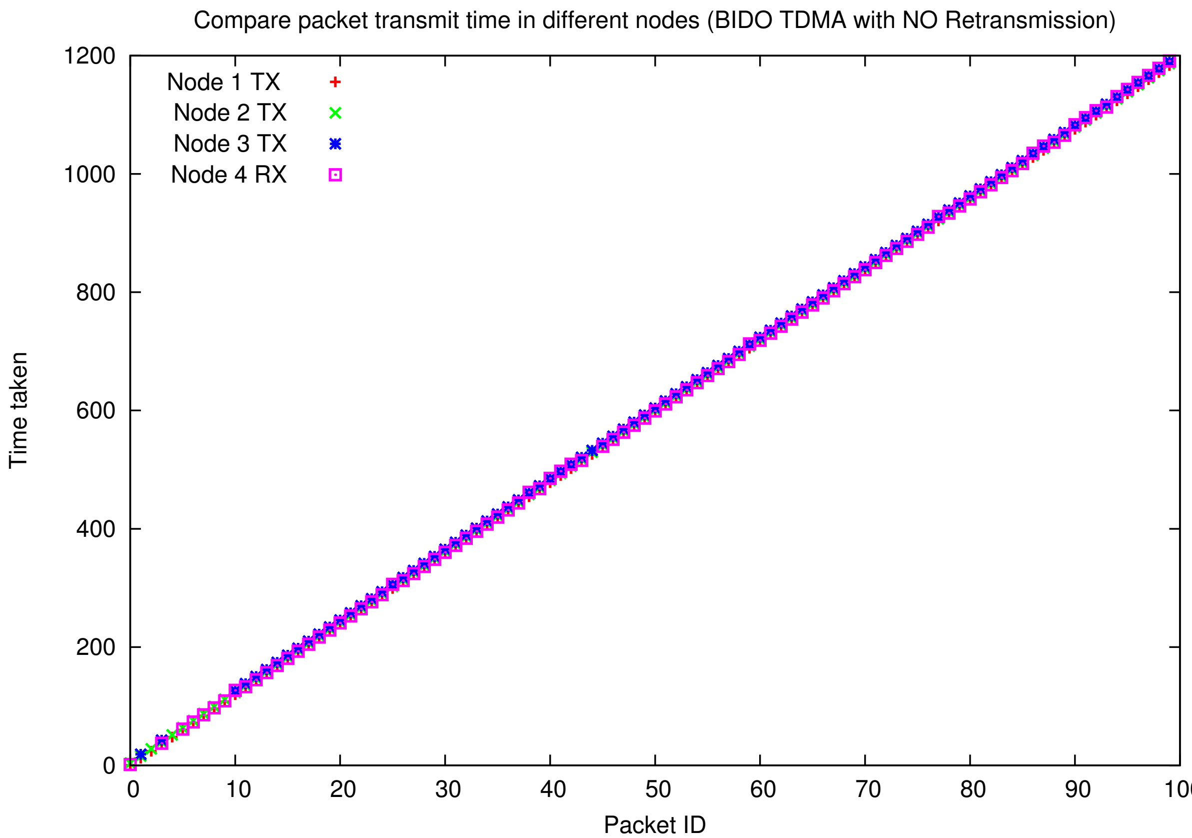


bad links, while Basic TDMA does not. This argument is supported by Figure 18 and Figure 19 which exhibit the packet-transmitted time in Node 1 to Node 3 and the packet-received time in Node 4 for tests with Basic S&W TDMA and BIDO Basic S&W TDMA. Figure 18 shows that the worst link lies in the link between Node 2 and Node 3, which is the bottleneck in this test and results in a low PDR. On the other hand, Figure 19 shows that Node 4 is able to overhear from Node 1 directly most of the time and readily passes the packets to upper layers for processing.

- **Throughput**:
  As Basic S&W TDMA has the same frame length whether or not the opportunistic overhearing feature is activated, Basic S&W TDMA and BIDO Basic S&W TDMA have similar total delays in these two tests. However, with a higher PDR, BIDO Basic S&W TDMA enjoys a higher throughput than Basic S&W TDMA.

- **Network Energy Consumption per Successfully Delivered Packet**:
  The Network Energy Consumption per Successfully Delivered Packet for BIDO Basic S&W TDMA is actually the lowest among all tests, while that for Basic S&W

Figure 19 Plot of Packet-transmit time vs packet ID for the Test with BIDO Basic S&W TDMA with no retransmission.



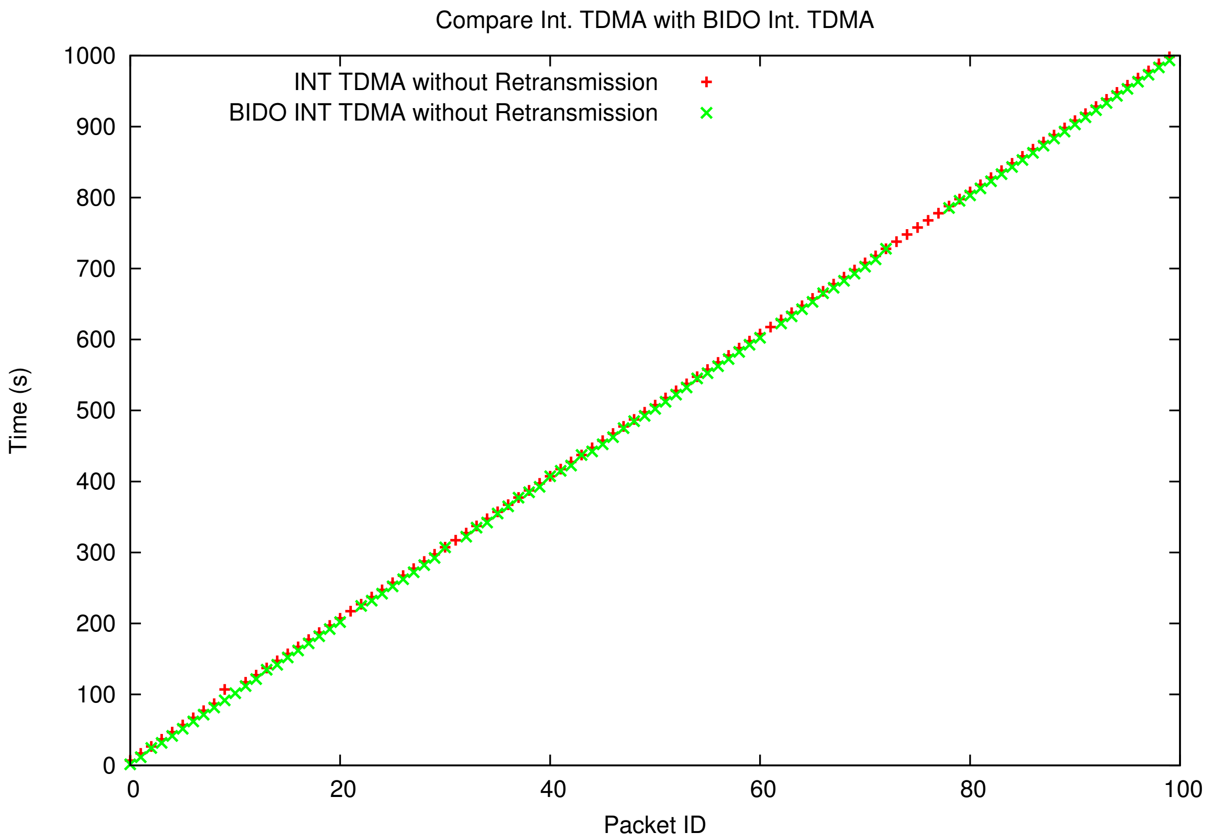
TDMA is the highest among all tests. The reason is that BIDO Basic S&W TDMA has a much better PDR than Basic S&W TDMA in this case.

Comparing Static Routing with Opportunistic Routing (Int. S&W TDMA)

Based on the results obtained from the sea trial performed on $3^{rd} Oct 2011$, for the case in which retransmissions are allowed, the performances between Int. TDMA and BIDO Int. TDMA are almost identical in term of PDR, Throughput and Energy Consumption per Successfully Delivered Packet. One observation about the energy consumption is that due to the overhearing feature that can reduce unnecessary transmissions and retransmissions, BIDO Int. TDMA actually results in fewer total number of packet transmissions, which means BIDO Int. TDMA spends less energy in transmitting packets. However, the Energy Consumption per Successfully Delivered Packet is still slightly higher for BIDO Int. TDMA because it has a higher rate of overhearing during the test.

In the following, we will focus on tests performed on $19^{th} Sep 2011$, when no retrans-

Figure 20 Plot of Packet-arrival time vs packet ID across tests with Int. S&W TDMA and BIDO Int. TDMA



missions are allowed.

- **Packet Delivery Ratio**:
  When there is no retransmission, Int. TDMA (PDR = 99%) performs slightly better than BIDO Int. TDMA (PDR = 92%) , although BIDO Int. TDMA is supposed to be at least as good as Int. TDMA. It is likely because of the difference in channel conditions between the two tests. During the test for BIDO Int. TDMA with no retransmission, five of the total eight lost packets are consecutive (Packet 73 to Packet 77), which indicates a deterioration in the channel during that period.

- **Throughput**:
  Int. TDMA enjoys a slightly higher throughput than BIDO Int. TDMA, because Int. TDMA has a higher PDR but a similar total delay.

- **Energy Consumption per Successfully Delivered Packet**:
  In spite of a lower PDR, BIDO Int. TDMA consumes less energy per successfully delivered packet than Int. TDMA. The reason is that compared to the lower PDR,

the test with BIDO Int. TDMA has a even lower rate of overhearing and a smaller number of total packet transmissions than the test with Int. TDMA.

5.4.4 Effects of Number of Hops and Transmission Power on the Performance

Besides the general study on the Data Link Layer and Data Delivery Scheme, the effects of the number of hops and the transmission power on the performance of the data delivery scheme given a fixed end-to-end distance are also studied empirically. The motivation for conducting this set of tests is to harvest preliminary and useful results from this sea trial that can facilitate our future study in optimizing the data delivery scheme for the real-world underwater communication.

Table 5 lays out the test plan for this purpose, and all the tests in this plan are carried out during the sea trial on 3^{rd} Oct 2011. Throughout all these tests, the MAC protocol used is Int. S&W TDMA, and simple static routing is selected as the data delivery scheme.

Test Plan 2: Fixed end-to-end Distance					
No.	**Scheme**	**Packets**	**Max. Retrans**	**Num of Nodes**	**Attenuation Level (dB)**
1	Int. TDMA	100	2	4	Low
2	Int. TDMA	100	2	4	Medium
3	Int. TDMA	100	2	4	High
4	Int. TDMA	100	2	3	Low
5	Int. TDMA	100	2	3	Medium
6	Int. TDMA	100	2	3	High
7	Int. TDMA	100	2	2	Low
8	Int. TDMA	100	2	2	Medium
9	Int. TDMA	100	2	2	High

Table 5: Test Plan for Studying the Effects of Num of hops and TX Power on Performance

The transmission power can be adjusted by changing the transmission attenuation level. The higher the value of the attenuation level, the lower the transmission power. For the two end nodes, the attenuation levels are changed manually. For the two relay nodes, the attenuation levels are changed remotely (see Section 5.2 for more details). The exact values of high, medium and low attenuation levels are decided at the beginning of the sea trial by performing some preliminary tests with just 20 packets. A value of 9 dB is chosen as the medium attenuation level because during the preliminary tests the PDR of the 2-node network is not perfect at this attenuation level (Though during the formal test, the PDR unexpectedly becomes 100%). As an increase (decrease) of 3 dB in attenuation level roughly halves (doubles) the transmission power, values of 6 dB and 12 dB are chosen as the low and the high attenuation levels.

To 'remove' one of the relay nodes, all that has to be done is to remotely change the MAC Address of the relay node to some number that is larger than 4. It is because with static routing, Node X will not participate in the communication among Node 1 to Node X-1. Therefore, to reduce the number of nodes from four to three, the original Node 3 will (remotely) be changed to, for example, Node 6, and the original Node 4 will (manually) be changed to Node 3, and then packets will be sent from Node 1 to Node 3 during a test. Moreover, to reduce the number of nodes from three to two, the original Node 2 can be changed to Node 5, and the current Node 3 will be changed to Node 2 and packets will be sent from Node 1 to Node 2. In this way the end-to-end distance is fixed, and the relay nodes do not have to be physically removed when they are not needed. On the other hand, when they are needed again, they can be brought back easily.

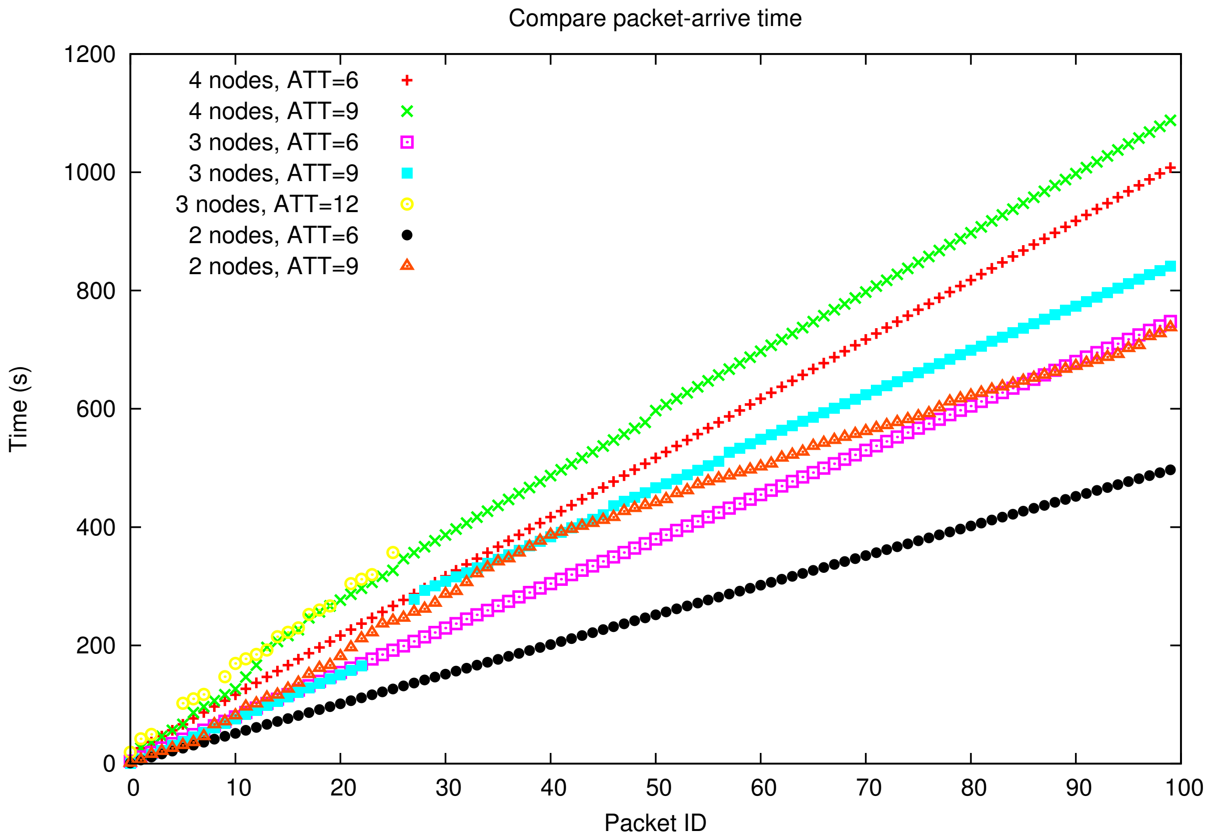
Table 6 summarizes the results for all the tests and Figure 21 displays the times that the receiver receives the packets under different circumstances. Note that Test 6 is deliberately terminated after the intended receiver fails to receive some 30 packets beyond Packet 25 and it is assumed that all packets beyond Packet 25 are lost. Test 3 and Test 9 are also deliberately terminated as none of the first 50 packets is received successfully by the intended receiver and it is assumed that all 100 packets are lost.

Experimental Results 2					
No.	**Scheme**	**Num of Nodes**	**Attenuation Level (dB)**	**PDR**	**Throughput (bps)**
1	Int. TDMA	4	6	1.00	21.4
2	Int. TDMA	4	9	1.00	18.1
3	Int. TDMA	4	12	0	0
4	Int. TDMA	3	6	1.00	28.9
5	Int. TDMA	3	9	0.96	24.6
6	Int. TDMA	3	12	0.21	>0.00
7	Int. TDMA	2	6	1.00	43.4
8	Int. TDMA	2	9	1.00	29.3
9	Int. TDMA	2	12	0	0

Table 6: Experimental Results for Studying the Effects of Num of hops and TX Power on Performance

- **Transmission Power Level**:
  For a particular network, when the transmission power decreases (i.e. attenuation level increases), the packet delivery ratio tends to decrease. For example, when the attenuation level increases from 6 dB to 9 dB and then to 12 dB for the 3-node network, the PDR decreases from 100% to 96% and then to 21%. For the

Figure 21 Plot of Packet-received time vs packet ID, when num. of hops and attenuation level change



2-node network and the 4-node network, although the PDR stays at 100% when the attenuation level reaches 9 dB, the throughput decreases distinctly, as more retransmissions occur and the total duration of the tests increases.

This shows that a high transmission power (relative to a given internodal distance) is important to ensure a good performance in terms of both PDR and throughput. However, higher transmission power may increase the network energy consumption. As energy is a very scarce resource for underwater network, it is important to determine the optimal transmission power level for a given network.

- **Number of Hops**:
  As the cost of an underwater modem is high, it is important to determine the minimum number of nodes needed for a given end-to-end distance (and other factors such as transmission power). As the end-to-end distance is relatively small in this case, when the attenuation level is 6 dB and 9 dB, the performance in terms of throughput increases when the number of nodes decreases, despite the PDRs are all close to 100% for different networks. The reason is that when the number of nodes increases, the number of slots in a frame increases, which in turn increases

the duration of a frame when the duration of a slot stays the same.

When the attenuation level is 12 dB, the PDR and thus the throughput drop to 0 for the 2-node network. The result is slightly better for the 3-node network, which shows more hops are necessary to help relay the packets from the sender to the receiver in this case. However, the results drop to 0 again when the number of nodes increases from three to four, although there are more nodes to help pass down the packets. The reason is that one of the links (Node 2 to Node 3) completely breaks down probably due to the change in the channel condition, which prevents any packets to be relayed by Node 3 and then received by Node 4 when static routing is used.

Thus, from these tests alone, it cannot be concluded that when the transmission power is not high enough to support a 2-node network for a given end-to-end distance, more nodes will be able to increase the PDR and throughput. However, in the case that direct communication is impossible between two end nodes, the existence of relay nodes are logically important to assist the transfer of data between two end nodes.

6 File Transfer Application

In Progress Report #3 [3], a sample sensing application is presented, which demonstrates the utility of the data delivery scheme in a real-life linearly arranged multihop underwater network. The network consists of a single sink node and multiple sensing nodes that, at the same time act as relays as well. At each sensing node, seawater temperature reading and global positioning system (GPS) coordinates are collected via a Vernier Go!Temp USB Temperature Sensor and GlobalSat BU-353 USB GPS Navigation Receiver respectively. The sensory data are then formatted accordingly and transmitted along the underwater acoustic network to a sink node, where the data is then displayed graphically on a mapping software.

In this section, we demonstrate the reliability of the proposed DDS in a real-life linearly arranged multihop underwater network for the case of a File Transfer Application.

6.1 File-Transfer Application Implementation

The file transfer program is implemented on the Application Layer, and uses a socket to connect to the data delivery program to transmit and receive data. The interface between an application program and the data delivery program is shown in the network structure in Figure 2. The application allows any node to be a sender or a receiver, but for simplicity, the subsequent details about the implementation assume Node 1 is the sender and Node 4 the receiver.

At the beginning of the file transfer, Node 1 calculates the number of packets the file has to be segmented based on the file size and the packet size. Node 1 then transmits the first packet to inform Node 4 the packet IDs of the first packet and the last packet that will carry the data. After sending all the packets, the program in Node 1 will wait for Node 4 to inform Node 1 whether there is any missing packets. If so, Node 1 will retransmit those packets.

After receiving the first packet from Node 1, Node 4 reserves the space for the packets and sets an alarm to estimate the expected remaining duration of the file transfer process based on the packets it has received and then uses the information to set another alarm to check the missing packets. When the second alarm expires, it is roughly the time that Node 4 would receive the last packet, and Node 4 will check its missing packets and request Node 1 to retransmit those packets. Moreover, Node 4 also estimates when all the missing packets will arrive and set a third alarm to check any further missing packets. This process goes on until Node 4 receives all the packets, after which they are grouped into a file.

6.2 Experimental Setup and Results

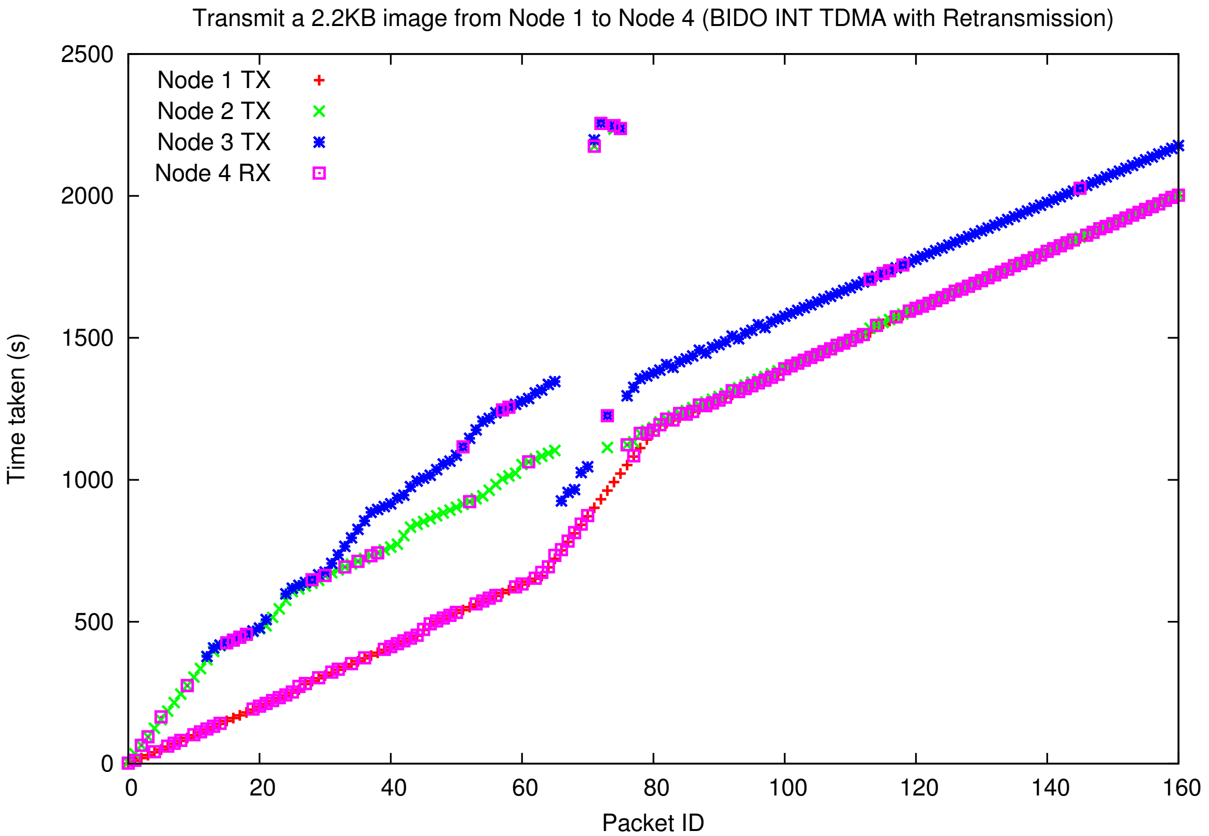
Figure 22 A 2.2KB image of the young Albert Einstein is sent over the 4-node network

The file-transfer demonstration is done on 19^{th} Sep 2011, at the sea area around St. John's Island[5]. In this demo, Node 1 is the sender, Node 4 is the receiver, and Node 2 and Node 3 act as the relay nodes. Section 5.4.1 provides more details regarding the experimental setup.

During this demonstration, the MAC Protocol used is Interweaved TDMA, with up to two retransmissions allowed. The data delivery scheme with opportunistic ARQ with Bidirectional Overhearing is also employed. Figure 22 shows the 2.2KB image (jpeg)

[5]Another file-transfer demonstration with a shorter internodal distance (30m) and smaller file size (1KB) was also performed on 14*th* Sep 2011 at the RSYC marina. This was to ensure that the application works well before moving on to the sea trial.

Figure 23 File Transfer: BIDO Int. TDMA, 2 retransmission, network time sync is achieved during the process



that is sent over the 4-node network, and Figure 23 shows the the time when Node 1, Node 2 and Node 3 transmit a particular packet and the time when Node 4 receives a particular packet, from which it can be seen that Node 4 is able to overhear and directly pick up Node 1's packets most of the time. Among those packets that Node 4 is unable to overhear directly from Node 1, they can be either overheard from Node 2 or received in a 'normal' way from Node 3. Among the 161 packets Node 1 sends to Node 4, four packets are lost and Node 1 retransmits those packets after Node 4 requests for them, and Node 4 receives these packets successfully. After the file transfer is complete, the image is opened and checked to ensure the transfer of the image is successful.

The overall duration of the file transfer process is 2175 seconds, PDR is 0.975 for the original 161 packets, MAC-layer throughput is 15.6 bps, and application-layer goodput is about 8 bps.

7 Conclusion

7.1 Summary

This report presents the achievements that fulfil the three tasks this project aims to achieve. The first task is to design and implement a data link layer for an underwater network, the second task is to develop a multi-hop routing scheme, and the last task is to integrate the whole data delivery scheme with the ARL modems and verify the integration through field tests. The achievements are summarized as follows:

- **Network Architecture**: To fulfil the three tasks, a Network Architecture is designed and implemented. Inside the architecture, the data delivery scheme that spans over the network layer and data link layer is the focus of this project. The interface between user applications and the data delivery scheme is also clearly defined, which allow us to create sample applications to demonstrate the how the underwater network works. One sample application implemented is File Transfer Application, which shows how a file with a size in the KB-range is reliably transferred from one node to another.

- **Data Link Layer**: A data link layer that mainly relies on TDMA-related protocols (Int. S&W TDMA and Basic S&W TDMA) is implemented so as to complete Task 1. The data link layer consists of several sublayers, including upper MAC layer, lower MAC layer and Modem Driver. Upper MAC layer is where the MAC protocols reside in. As time synchronization is a necessary component for TDMA-related protocols, a time synchronization scheme that relies on timestamps piggybacking on ACK packets is incorporated into the upper MAC layer. Lower MAC layer and Modem Driver are responsible for controlling the hydroacoustic modems. They hide the complexity of controlling the modems and allow upper MAC layer to easily transmit and receive packets through the modems.

- **Data Delivery Scheme**: To overcome the limitations of static routing, a cross-layered multihop data delivery scheme that exploits the full benefits of opportunistic ARQ through bidirectional overhearing is implemented. The data delivery scheme relies on overheard upstream DATA packets to fast-forward the packets to the sink node, and on overheard downstream DATA packets and ACK packets to eliminate unnecessary transmission. This is to fulfil Task 2.

- **Integration of DDS to ARL Modem Platform and Field Testing**:

  The network architecture is implemented using the "pure user implementation architecture" to fulfil Task 3. Prior to the actual field tests, the data delivery scheme is tested in the water tank at NUS-ARL and at RSYC in accordance with the practice of incremental testing. Sea Trials are carried out to validate the integration efforts in a real-world setting. The schemes were evaluated in terms of packet delivery ratio, goodput, packet delay and network energy consumption.

The results show that as a MAC protocol Int. S&W TDMA performs better than Basic S&W TDMA. The results also show that the cross-layer data delivery scheme that makes use of opportunistic overhearing improves the performance of static routing with Basic S&W TDMA. However, whether this feature can always improve the performance of static routing has to be further verified as some results show that static routing with Int. S&W TDMA performs slightly better than the opportunistic version during the sea trials.

Besides, the effects of transmission power and number of hops on the performance of a network with fixed end-to-end distance are also briefly studied. The results show that higher transmission power unambiguously improves the performance of the network in terms of PDR and throughput, although a very high transmission power may cause the network to consume too much energy and shorten the lifespan of the network if continuous power supply to the nodes is not achievable. More relay nodes are logically sound to improve the network performance when direct communication between two end nodes is impossible, but this statement has to be further verified in future sea trials as the results obtained from the sea trial on 3^{rd} Oct 2011 are not conclusive.

7.2 Recommendations

Despite the successful completion of this project with achievements that fulfil the project aims, there are still numerous improvements that can potentially enhance the performance of data delivery in underwater acoustic networks.

- **Application Layer**: More sample applications can be developed to demonstrate the usefulness and reliability of the data delivery scheme.
- **Transport Layer**: Because the data rate that an underwater acoustic network is able to support is much lower compared to a terrestrial wireless network, and the round trip time (RTT) between two nodes in the underwater network is unlikely to be stable even in the short run, the widely-used transport layer protocol such as TCP that heavily relies on estimated RTT to perform cumulative ACKs may not be suitable for the underwater network. However, as TCP has many useful and widely-tested features, A TCP-like underwater transport layer can be designed and optimized for the underwater environment.
- **Network Layer**: Currently, whether it is static routing or opportunistic routing, the main route along which packets travel is largely fixed. While this is adequate for a linear topology with a fixed number of nodes, the current data delivery

scheme will be unable to handle more complex topologies where nodes may join or leave the network. Therefore, it is important to allow nodes in a network to dynamically coordinate among themselves to find out a route for packet routing between any two nodes.

- **Data Link Layer**: In the case that TDMA-related MAC protocols are used, spatial reuse can be considered to allow TDMA to be more scalable. The time synchronization scheme used by TDMA-related protocols can be improved to better synchronize clocks of the nodes in a network. In addition, Data with piggybacked ACK frames should be fully implemented to enhance the efficiency of communication in certain situation.
- **Physical Layer**: During the sea trials carried out for this report, the duration of a slot is about 2.5 seconds when Int. S&W TDMA is selected as the MAC protocol, which makes the TDMA frame for the 4-node network be about 10 seconds. The relatively large duration is mainly to accommodate various hardware delays such as HPA switching delay. On the contrary, the propagation delay for a sound wave to travel along a 110-meter distance in the water is only 73 ms. Despite it being much larger than the radio propagation delay for the same distance, so far it is a minor concern in determining the duration of a slot. Therefore, in order to improve the performance in terms of throughput, it is important to decrease the hardware delays so that a shorter slot duration (and thus a shorter TDMA-frame duration) is sufficient for the operations.
- **Optimization of Parameters**: Thus far the parameters of the data delivery scheme set for the field tests tend to be conservative, as our priority is to verify the successful integration of the data delivery scheme to the ARL modems. For example, the size of a DATA frame is only 27 Bytes during the sea trials carried out in Sep 2011 and Oct 2011. On the other hand, for simplicity the size of an ACK frame is set to be the same as the size of a DATA frame, which is larger than what an ACK frame needs. To allow data to be more efficiently transferred in the underwater acoustic network, parameters such as packet size, transmission power, number of minimum nodes needed given a fixed area etc. have to be optimized.
- **Extensive Field Tests**: Field tests with more nodes and a larger area should be carried out to accept or reject some hypothesis.

References

[1] H. Tan, H. Zhuang, A. Valera, and Z. Bai, "Robust multihop underwater network to support long-range sensing applications," 2009, Institute for Infocomm Research, Progress Report #1.

[2] H. Zhuang, A. Valera, Z. Bai, and H. Tan, "Robust multihop underwater network to support long-range sensing applications," March 2010, Institute for Infocomm Research, Progress Report #2.

[3] ——, "Robust multihop underwater network to support long-range sensing applications," September 2010, Institute for Infocomm Research, Progress Report #3.

[4] M. Chitre, L. Freitag, E. Sozer, S. Shahabudeen, M. Stojanovic, and J. Potter, "An architecture for underwater networks," *OCEANS 2006 - Asia Pacific*, pp. 1–5, May 2006.

[5] A. Valera, P. W. Q. Lee, H. P. Tan, H. Liang, and W. K. G. Seah, "Implementation and evaluation of multihop arq for reliable communications in underwater acoustic networks," *Proc. of OCEANS*, May 2009.

[6] "Qualnet 4.5, programmer's guide," *Scalable Network Technologies Inc*, http://www.scalable-networks.com.

[7] A. A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," *Proc. INFOCOMM 2006*, pp. 1–12, April 2006.

[8] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "Mu-sync: A time synchronization protocol for underwater mobile networks," *WuWNeT 2008*, September 2008.

[9] H. Kopetz and W. Schwabl, "Global time in distributed real-time systems," Technische Universitat Wien, Tech. Rep. 15/89, 1989.

[10] M. Horauer, U. Schmid, K. Schossmaier, R. Holler, and N. Kero, "Psynutc - evaluation of a high precision time synchronization prototype system for ethernet lans," *Proceedings of 34th Annual Precise Time and Time Interval Meeting (PTTI)*, pp. 263–279, December 2002.

[11] "Underwater acoustic communications and networking," http://arl.nus.edu.sg/twiki/bin/view/ARL/UNET.

[12] "Pop-up ambient noise data acquisition," http://arl.nus.edu.sg/twiki/bin/view/ARL/PAND