COMMUNICATION ON LIMITED-MOBILITY UNDERWATER SENSOR NETWORKS

by

Nicholas Y. Yuen

A Thesis Submitted to the

Office of Research and Graduate Studies

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

School of Engineering and Computer Science
Engineering Science

University of the Pacific
Stockton, California

2013

COMMUNICATION ON LIMITED-MOBILITY UNDERWATER SENSOR NETWORKS

by

Nicholas Y. Yuen

APPROVED BY:

Thesis Advisor: Elizabeth Basha, Ph.D.

Committee Member: Ken Hughes, Ph.D.

Committee Member: Carrick Detweiler, Ph.D.

Department Chairperson: Jennifer Ross, Ph.D.

Interim Dean of Graduate Studies: Bhaskara R. Jasti, Ph.D.

ACKNOWLEDGMENTS

This thesis would not have been possible without the help my advisor Dr. Elizabeth Basha. She helped to motivate and guide me throughout the entire process. I have learned a great deal throughout this journey and I owe most of it to her.

I would also like to thank the remainder of my committee Dr. Carrick Detweiler and Dr. Ken Hughes. They pointed out important issues that I needed to address and helped to refine my thesis.

I would also like to thank the National Science Foundation for supporting the research (CSR #1217400 and CSR #1217428) as well as the School of Engineering and Computer Science and the University of the Pacific Graduate Office.

My family has always been there for me. They supported and encouraged me the entire way. I owe many thanks to them as well.

Finally, this thesis was fueled and supported by copious amounts of coffee. Without it I would have never finished.

Communication on Limited-Mobility Underwater Sensor Networks

Abstract

by Nicholas Y. Yuen

University of the Pacific
2013

More than 70% of Earth's surface is covered by water. Earth's underwater world holds many exciting forms of life and undiscovered possibilities. It is sometimes referred to as "The Unexplored Frontier." We still do not fully understand the entirety of what happens in this mysterious world. The field of underwater sensor networks is a means of monitoring these environments. However, underwater sensor networks are still fraught with challenges; one of the main challenges being communication. In this thesis we look to improve communication in underwater sensor networks.

We expand a simulation environment that models node to node communication in an underwater sensor network that utilizes AquaNodes. We address issues with the first iteration of the environment, expand it to include packet-loss for acoustic communication, and make the addition of three dimensional topologies. We found that acoustic packet-loss had a larger impact on the energy consumption of the communication algorithms with more acoustic communication and three dimensional topologies do not affect the communication

algorithms.

In addition to expanding the simulation environment we also explore using UAVs as a means of extracting data out of underwater sensor network. We conduct field experiments to characterize radio communication, develop an energy model to understand the energy limitations of an UAV, and develop overall policies for using an UAV with an underwater sensor network that utilizes AquaNodes. We learned that node to node radio communication range on the surface of the water had shorter ranges than on land. We also learned that node to UAV communication range was dependant on the altitude of the UAV. Overall, we found that using an UAV as a data mule was a viable method of extracting data out of certain underwater sensor network configurations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Underwater wireless sensor networks have applications ranging from climate monitoring, studying marine life, pollution control, prediction of natural disasters, search and survey missions, and potentially many other unexplored uses. However, one of the main challenges that underwater wireless sensor networks face is communication.

There are four main types of communication used in underwater sensor networks, acoustic, radio, satellite, and optical communication. Acoustic communication is the only long range underwater option of these four communication mediums. However, it suffers from a low data-rate, high energy consumption, and many complex problems. Radio communication is typically the dominant communication medium outside of water. It provides long distance communication with high data-rates. However, it can not be used underwater because water absorbs and disperses the vast majority of radio frequencies. Satellite is another long range surface communication method. However, scalability issues arrise as satellite communication is expensive. Optical communication and satellite communication are not explored in this thesis; we chose to primarily focus on radio communication and acoustic communication.

In this thesis we describe a simulation environment that models node to node communication in an underwater sensor network that utilizes AquaNodes. AquaNodes are sensor nodes that were developed by Carrick Detweiler. They have the unique capability of autonomous depth adjustment [4]–[6], [14], [23]. They also have radio, acoustic, and optical communications. We take advantage of the depth adjustment feature of the AquaNode in order to have the AquaNodes and UAV communicate with each other via radio communication. We also do experiments and simulations to analyze the effectiveness and limitations of using an UAV (Unmanned Aerial Vehicle) as a data mule in an underwater sensor network that also utilizes AquaNodes. Being able to simulate communication in underwater sensor network is important because an underwater sensor node can easily cost $1000 and is expensive to deploy. Simulation enables quick and reproducible results over a wide array of different network configurations for basically nothing. However, if the simulation environments are too theoretical then the results can be inaccurate. In order to avoid this issue we use experimentally measured communication range and energy results to improve the accuracy of our simulations.

1.1 Problem Overview

In this thesis the overarching problem that we explore is improving communication in underwater sensor networks. Under this, there are two main problems that we explore.

The first is simulating node to node communication in underwater sensor networks that utilize AquaNodes. In order to simulate node to node communication we modify and extend an existing simulation environment created by Michael O'Rourke [15]. It simulated different communication routing algorithms in underwater sensor networks that use AquaNodes. However, it was limited to two dimensional, linear network topologies and no packet-loss. In this thesis we extend the functionality of this simulator to support three dimensional network topologies and acoustic packet-loss. While extending the functionality of the simulation environment we came across and fixed a number of issues. These issues and fixes are also covered in this thesis.

The second is exploring using an UAV as a data mule for underwater sensor networks that also use AquaNodes. We chose to explore this network configuration because it was something that has not been done before and we wanted to discover the benefits and limitations of such a system. Typically, using an UAV as a data mule in an underwater sensor network is impossible because the node is submerged and unable to use radio communication. However, in our case UAVs become a viable option because AquaNodes have the ability to control their depth and rise to the surface. The idea is that the UAV will start at a base station, fly out to the network, traverse the network, use radio communication to extract data from each node in the network, and fly back the base station. In this thesis we conduct two field experiments to experimentally determine radio communication ranges. The first experiment determines the ranges and packet-loss values between two surfaced sensor nodes. The second is to determine optimal radio communication ranges and altitudes between a surfaced sensor node and an UAV. We also analyze the energy characteristics of the UAV and come up with overall policies.

The idea is to eventually combine both sections of work in this thesis into one cohesive and robust simulation environment. The combined simulation environment will have the ability to simulate the behaviour of AquaNode underwater sensor networks with different data extraction methods including multi-hop and UAV data muling. This will enable us to model and compare network lifetime in different network configurations.

1.2 Thesis Contributions

This thesis makes several contributions to underwater sensor network communication and aerial robotics. A simulation environment that models node to node communication in underwater sensor networks is further developed. A novel idea of using an UAV as a data mule in an underwater sensor network is explored. The contributions are:

- Correcting power constant calculation, distance calculation, and energy update errors in the node to node underwater sensor network simulation environment
- Addition of acoustic packet-loss to simulation environment
- Addition of three dimensional network topologies to simulation environment
- Experimentally measured radio communication ranges between two surfaced water sensor nodes
- Experimentally measured radio communication ranges between UAV and surfaced water sensor node
- UAV energy model, based on measured values, to simulate UAV power consumption
- Overall usage policies for using an UAV for data muling in an underwater sensor network

The hope is that the contributions of this thesis will help towards improving communication in underwater sensor networks.

1.3 Thesis Outline

The rest of this thesis is as follows. In Chapter 2 we start with related work. Chapter 3 goes into detail about the fixes and additions to the simulation environment for node to node communication in underwater sensor networks. In Chapter 4 we discuss using an UAV as a data mule in an underwater sensor network and the experiments and simulations that we did. Finally, in Chapter 5 we conclude and discuss future work.

Chapter 2: Related Work

In this thesis we further develop a simulation environment that models node to node communication in underwater sensor networks. We then explore using an UAV as a data mule for an underwater sensor network. The underwater sensor nodes that we model are AquaNodes. We utilize the AquaNodes depth adjustment, acoustic communication, and radio communication capabilities. The remainder of this section will discuss the prior work in underwater sensor network communication, data muling, and the AquaNode platform.

2.1 Underwater Sensor Network Communication

There are four main types of communication used in underwater sensor networks, acoustic, radio, satellite, and optical communication. Acoustic communication is the only long range underwater option of these four communication mediums. However, it suffers from a low data-rate (in the order of 1000 bits per second) [8], high energy consumption (0.1136 joules per bit for an AquaNode) [6], and many complex problems. Radio communication is typically the dominant communication medium outside of water. It provides long distance communication with high data-rates. However, it can not be used underwater because water absorbs and disperses the vast majority of radio frequencies [8]. Satellite is another long range surface communication method. However, scalability issues arise as it is important to keep the cost of each indivual node down and satellite communication is expensive [3]. Optical communication can be used underwater at distances up to 3 meters at variable speeds up to 4 megabits per second (for an AquaNode) [6]. Optical communication and satellite communication are not explored in this thesis; we chose to primarily focus on acoustic communication and radio communication.

The remainder of this section will discuss challenges in underwater acoustic communication, prior work in surface radio communication, and prior work in communication path planning.

2.1.1 Underwater acoustic communication.

The main form of communication today in underwater sensor networks is acoustic. It typically offers bit rates of up to a few kilobits per second at communication ranges up to a few kilometers [8]. In our system, acoustic communication is used for underwater node to node communication. There are many challenges that come with acoustic communication in underwater sensor networks. These challenges make efficient underwater acoustic communication difficult and in turn lead to high packet-loss rates; this is what drove us to model acoustic packet-loss in the simulation environment. The challenges for underwater acoustic communication are described in [2] and include:

- **Path Loss:** Path loss has two sub-categories of challenges, attenuation and geometric spreading. Attenuation is mainly caused by absorption due to the conversion from acoustic energy to heat. It can also be caused by scattering, reverberation, refraction, and dispersion. Geometric spreading is the spreading of sound energy as a result of the expansion of the wave-fronts. The problem worsens with the increase of distance the wave propagates but is frequency independent.

- **Noise:** Noise also has two sub-categories, man-made noise and ambient noise. Man-made noise mostly comes from machinery or shipping activity. Ambient noise comes from the movement of water, seismic activity, and biological activity.

- **Multi-Path:** Multi-path propagation can cause severe acoustic communication signal degradation as it causes Inter-Symbol Interference (ISI). However, in our case the AquaNode's communication protocol attempts to avoid this by designating each node of the network a time slot in which it is allowed to communicate acoustically.

- **High Delay and Delay Variance:** Underwater acoustic communication has a large propagation delay, around .67 s/km. This is five orders of magnitude slower than above-water radio communication. Even more troubling than the large propagation delay is the high variability of the delay. This makes estimating the round trip time difficult. The round trip time is a key measure for many common communication protocols.

- **Doppler Spread:** The Doppler frequency spread in underwater acoustic communication can be significant enough to cause degradation in the communication performance.

Most of the listed challenges of underwater acoustic communication are caused by the chemical-physical properties, water temperature, salinity, and density [2]. Our simulation model is based on the acoustic communication performance of the AquaNode [6]. Experimental results for AquaNodes show that success rate of acoustic communication is approximately 56% at typical communication range [6].

2.1.2 Surface radio communication. Radio communication is an ineffective means of underwater communication. However, the AquaNodes that we model are capable of rising to the surface and using radio communication above water. Radio communication offers higher data rates (KHz to tens of MHz), further ranges (meters to hundreds of kilometers), and uses less power (0.00016 joules per bit for an AquaNode) [6], [8], [9], [12] than acoustic communication. The idea for our system is that radio communication is only used when transmitting large amounts of data between nodes and acoustic is utilized for everything else including path planning.

2.1.3 Path planning. Our system utilizes path planning to determine which nodes will rise for a multi-hop radio communication. We examine different path finding algorithms to find the most energy efficient ones. We use greedy and optimal approaches adapted from [16], [21]. Stojmenovic [16] examined a wide variety of routing protocols and found that greedy algorithms have performance that rivals an optimal shortest path algorithm for dense graphs, but low delivery rates for sparse graphs. Also, algorithms that guarantee delivery may have high communication overhead for sparse graphs. Tan [21] studies the problem of shortest-path geographic routing in static sensor networks and develops an algorithm based on the construction of a reduced visibility graph to find near optimal paths. Given the effectiveness of greedy approaches on land, we choose to explore their effectiveness in underwater situations.

2.2 Data Muling

Prior research has not explored the use of UAVs as data mules with underwater sensor networks, to the best of our knowledge. Researchers have used UAVs as data mules with land based sensor networks and AUVs (Autonomous Underwater Vehicles) as data mules with underwater sensor networks; we summarize those works.

2.2.1 UAVs and land sensor networks. A preliminary design for land based UAV data muling is presented in [22]. They use a modified Boomerang 60 model aircraft as the UAV and Fleck 3 sensor nodes. They perform three major experiments: the first involved testing the radio communication range between ground and UAV, the second was designed to demonstrate the data muling capability of the UAV, and the third involved integrating the Fleck 3 node with the UAV autopilot. They conclude that UAVs are feasible data mules for wireless sensor networks. The idea of using UAVs as data mules for land based sensor networks is also seen in [10], [11], [17]–[20]; these are geared more towards UAV routing and control algorithms. Our work looks to utilize a quadcopter instead of an airplane as a data mule and is used in an underwater sensor network instead of a land based sensor network.

2.2.2 AUVs and water sensor networks. An underwater sensor network platform is presented in [24]. The platform consists of static sensor nodes called AquaFlecks and AUVs used for data muling and network maintenance called Amour and Starbug. The network uses optical and acoustic communication for networking. The Amour AUV uses optical communication to extract data from the AquaFlecks. What they found was that AUV data muling provided a power efficient, effective means of harvesting data in underwater sensor networks. AUV data muling is heavily explored in [7]; they focused primarily on novel AUV and optical communication design. They successfully deploy a fully autonomous underwater data muling system in the field. Underwater data muling is discussed in [13] and they formulate a path planning algorithm for the data mule. They found that in small, sparse networks the algorithm performed better than the corresponding optimal solutions for the traveling salesman problem.

We look to utilize sensor nodes called AquaNodes that have the ability to surface and use radio communication in order to communicate with a UAV that will act as a data mule. UAVs travel faster than AUVs thus offering shorter data collection times.

2.3 AquaNode

The sensor nodes that we use in this thesis are AquaNodes [4]–[6], [14], [23]. In normal operation an AquaNode is anchored to the seafloor and floats in the water mid-column sensing the environment. AquaNodes have the ability to adjust their depth in the water, have three types of communication mediums, and multiple sensors. The depth adjustment system allows for the AquaNode to dynamically rise and descend in the water column.

Figure 1 shows the AquaNode and the winch based depth adjustment system and Figure 2 shows the details of the depth adjustment system. AquaNodes are capable of radio communication at 57kbit/s within 3km, acoustic communication at 300b/s within 400m, and optical communication at 3Mb/s within 3m [5]. Radio communication is only used when the node surfaces while acoustic and optical are used underwater. For sensing, each AquaNode has a pressure sensor, temperature sensor, color camera, and the ability to add additional sensors [23]. The simulation environment models the unique features

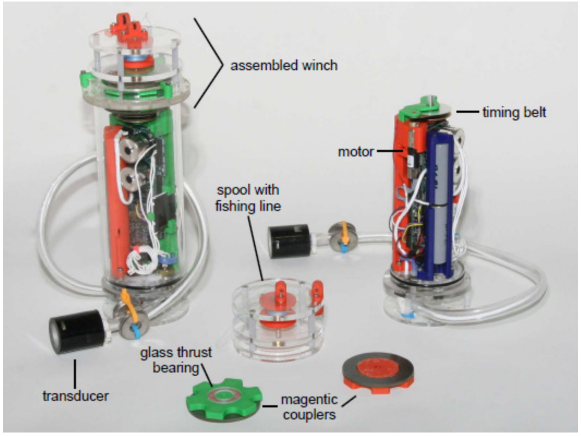

Figure 1: AquaNode and depth adjustment system [5].

of the AquaNode, the energy consumption values, and the communication ranges. Our work looks to utilize the depth adjustment system and surface radio communication of the AquaNode for UAV data muling.

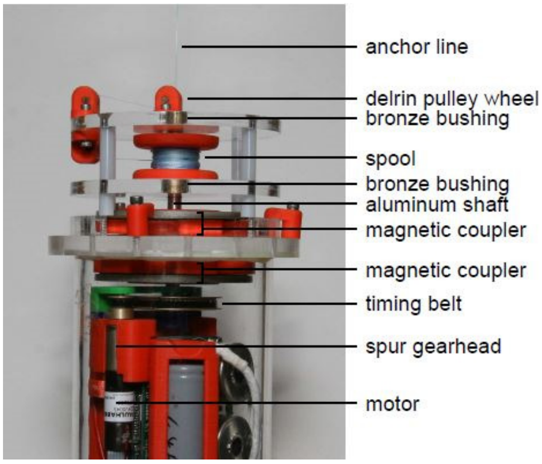

Figure 2: Depth adjustment system details [5].

Now that we covered the related work in the field we will now move on to discuss the work done in this thesis.

Chapter 3: Simulation

The simulation environment described in this chapter is built upon the work of Michael O'Rourke [15]. This simulation environment models multi-hop communication in an underwater sensor network that utilizes AquaNodes. The simulation environment routes a packet of data between two AquaNodes that are on opposite sides of the network. It uses underwater acoustic communication to calculate a multi-hop path between the nodes, the nodes rise, and then use radio communication to send the packet of data across the network. To calculate this path of this packet of data there are eight different communication routing algorithms that the user can choose between.

While calculating the path of the packet, the simulation environment tracks a number of metrics. The main performance metric of the system is energy consumption. However, other metrics such as messages sent and run-time of the entire process are also measured. In this chapter we discuss the fixes we made to the simulation environment, the addition of acoustic packet-loss, and the addition of three dimensional topologies.

This remainder of this chapter outlines the ideas in the following order. Section 3.1 discusses the initial implementation of the simulation environment. Section 3.2 will discuss the fixes we made to the simulation environment and how the fixes impacted the results. Section 3.3 will discuss the addition of acoustic packet-loss into the simulation environment, why we thought it was a useful addition, and the change in results. Section 3.4 will discuss the addition of three dimensional network topologies, the different approaches that we attempted, and how the results were impacted.

3.1 Initial Simulator

The initial simulator that Michael O'Rourke created was implemented as a set of MATLAB scripts and functions. The environment is a procedural environment and consists of scripts or functions. The first are initialization scripts; these initialize a data field such as node positions or parameter constants such as acoustic range. The next class of scripts or functions are modifiers; these are helper functions or scripts that manipulate the storage structures. The last class of scripts or functions are used in actually running the simulation environment. This could be the process of generating messages, determining the path of acoustic messages, or responding to commands.

Figure 3 shows the general flow of the simulation environment. There are two main phases of the simulation environment: preparation and execution. In the preparation stage constants such as number of nodes, communication power, and communication range are defined and initialized. Then the node positions are created based on a linear topology that follows rules set when the constants were defined. The next step involves creating two connectivity matrices that store which nodes are connected to each other through radio communication and acoustic communication. These connections are created based on communication range constants and node positions. The next step is initializing the message queues. In the execution stage the simulator first determines what communication routing algorithm had been specified by the user. If it was an optimal or centralized algorithm the full multi-hop communication path is determined, the nodes rise, and route the message across the network. In the case that it is a greedy algorithm the simulator calculates the multi-hop communication path node by node. The starting node communicates with its neighboring nodes and determines the next node in the sequence, then that node continues that trend, and the pattern continues until the destination node is reached. After each node determines the next node in the sequence it rises to the surface. Finally the message is forwarded via radio communication.

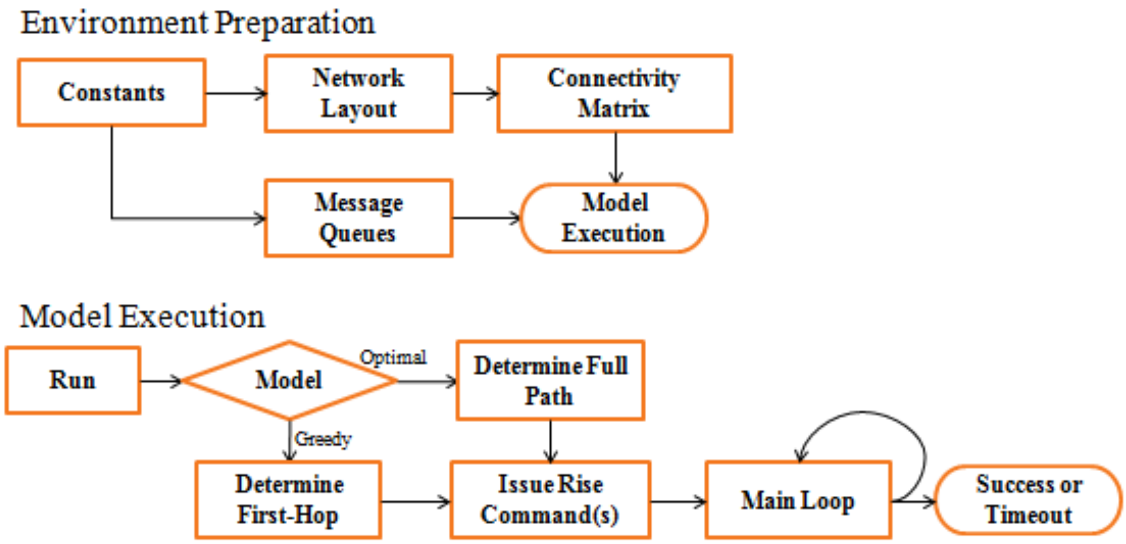

Figure 3: Flowchart of initial simulator

The initial simulator had eight different communication routing algorithms. These algorithms were separated into two different categories: acoustic-centric and radio-centric. Algorithms were considered acoustic-centric if they made decisions based on neighboring nodes within acoustic communication range and radio-centric if they made decisions based on neighboring nodes within radio communication range. There are two acoustic-centric algorithms, *Greedy Furthest Acoustic* and *Greedy Shallowest Acoustic*. The other six algorithms are radio-centric. They are *Greedy Furthest Radio*, *Greedy Shallowest Radio*, *Greedy Look-Ahead*, *Greedy Look-Back*, *Min-Hop Furthest*, and *Min-Hop Shallowest*.

There were four experiments performed on the initial simulation environment. These experiments measured and compared the energy efficiency of the eight communication algorithms while skewing the simulation settings. The settings being skewed were node spacing, radio communication range, and acoustic communication range. In the first experiment, a set of baseline data was generated. The second experiment analyzed the effect of varying node spacing. In the third experiment, the radio communication range was varied. The fourth experiment examined the energy efficiency while skewing acoustic communication range. In each experiment the settings not being skewed were set to the values used in the first baseline experiment.

In terms of decentralized algorithms, the results of the four experiments showed that the radio-centric communication algorithms were generally more effective than the acoustic-centric communication algorithms. In the case where each AquaNode only knew the location of its neighbors the most energy efficient communication algorithm was *Greedy Shallowest Radio*. If the AquaNodes had knowledge of the location of its neighbors and their neighbors, *Greedy Look-Ahead* was the best choice.

3.2 Fixes

This section discusses the issues with the initial simulation environment, how we fixed them, and how they impacted the results.

3.2.1 Power constants calculation error.

While doing modifications to the simulator we noticed that there was a mismatch in the units of the power constants for the AquaNode. The constants *RADIO_SEND_ENERGY*, *RADIO_RECEIVE_ENERGY*, *ACOUSTIC_SEND_ENERGY*, and *ACOUSTIC_RECEIVE_ENERGY* were in kilojoules per bit. The constant *WINCH_ENERGY* was in millijoules per meter. This was not supposed to be the case, all five power constants were supposed to be in millijoules per bit or millijoules per meter. We changed the five power constants to be in either joules per bit or joules per meter. These constants are used to calculate the energy usage of the AquaNode, depending on how many bits are sent through radio/acoustic communication and how many meters it travels using its depth adjustment system. Table 1 shows the changes in constants:

Table 1: Changes AquaNode to power constants

Variable	**Before Change**	**After Change**
$RADIO\_SEND\_ENERGY$	$0.00000016 J/bit$	$0.00016 J/bit$
$RADIO\_RECEIVE\_ENERGY$	$0.0000000457 J/bit$	$0.0000457 J/bit$
$ACOUSTIC\_SEND\_ENERGY$	$0.0001136 J/bit$	$0.1136 J/bit$
$ACOUSTIC\_RECEIVE\_ENERGY$	$0.000063 J/bit$	$0.063 J/bit$
$WINCH\_ENERGY$	$15000 J/m$	$15 J/m$

There was another power constant calculation error that needed to be fixed. This problem involved the processing power calculation of the AquaNode. The processing power constant represents the maximum power consumption of the AquaNodes CPU. The CPU is not always drawing full power but the overestimation helps take into account small components that we are not modeling. Before the fix, the processing energy constant was: $PROCESSING\_ENERGY = 3.3 * (.059) * (4 * numberNodes)$. After the fix it was:

PROCESSING_ENERGY $= 3.3 * (.059)$. In the simulation the processing energy is updated at every communication iteration. Before the fix the processing energy constant was based on the voltage of the CPU (3.3V), the maximum current of the CPU (.059A), and number of nodes in the network. It was also supposed to be in milliwatts to match the rest of the energy constants, but was actually in watts. After the fix the processing energy constant was only based on product of the voltage and current with the units being watts. The way that the simulation environment calculates processing energy is by adding the processing energy constant to a total processing energy variable for each AquaNode separately. This means that before the fix, the processing energy update for each individual AquaNode was based on the processing energy of 4 times the processing energy for the entire network. This did not make sense, so it was changed.

Power constants verification and results. Now we analyze the impact that these changes had on the results. For this set of tests we used the same set of topologies for all tests. We use network sizes of 30, 60, and 90 nodes. First, we verified the changes by analyzing the average AquaNode energy consumption breakdown. Second, we compared the average total AquaNode energy consumption for all communication algorithms before and after the change.

Test one: AquaNode energy breakdown. First, we analyzed the change in the average AquaNode energy breakdown. In Table 2 we can see a major change in the average AquaNode energy breakdown for the *GreedyShallowestRadio* communication algorithm in a 60 node network.

From a high level we can see that before the fix the total energy consumption of the average AquaNode (40.18 joules) is completely dominated by processing energy (15.5 joules)

Table 2: Average AquaNode energy consumption breakdown before and after fix

Power Constant	Before Change (J)	After Change (J)
Total	40.18	102.2
Processing	15.5	64.5
Radio Receive	$8.96 * 10^{-10}$	$8.96 * 10^{-4}$
Acoustic Receive	$8.47 * 10^{-6}$	8.47
Radio Send	$1.47 * 10^{-9}$	$1.47 * 10^{-3}$
Acoustic Send	$4.6 * 10^{-6}$	4.6
Winch	24.7	24.7

and the winch energy (24.7 joules). The communication does not even come close to having a significant impact on the results. This is because the communication energy constants were off by a factor of 10^6 due to a unit conversion error.

After the fix the total energy consumption of the average AquaNode increases to 102.2 joules. This value is surprisingly dominated by processing energy (64.5 joules). This stems from a couple of reasons. First of all, we are modeling the AquaNode at maximum processing power with no sleep mode. Second, the amount of data that is being transferred is small. Acoustic and radio transmissions range from 24-48 bits with far more acoustic transmissions than radio transmissions. Communication energy could end up dominating the energy consumption if enough data is sent; we see this later in the centralized algorithms. Winch energy stayed the same at 24.7 joules. The communication energy consumption increases by a factor of 10^6 and actually impacts the energy consumption. Acoustic send energy is 4.6 joules, acoustic receive energy is 8.47 joules, radio send energy is $1.47 * 10^{-3}$ joules, and radio receive energy is $8.96 * 10^{-4}$.

Test two: AquaNode energy consumption. Second, we examine the energy efficiency of the communication algorithms before and after the power constant change.

The results from before the fix can be seen in Figure 4. Here we see that the two acoustic-centric communication algorithms consume much more energy than the six radio-centric algorithms. For example in a 60 node network *Greedy Shallowest Acoustic* uses 122 joules and *Greedy Shallowest Radio* uses 40 joules.

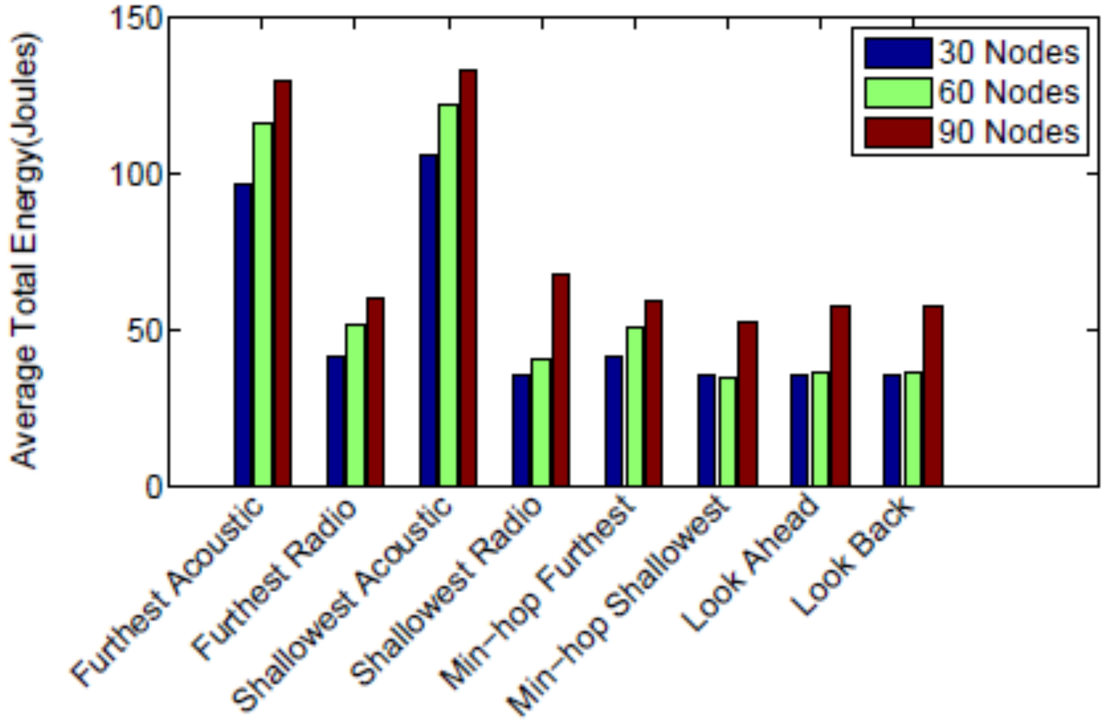

Figure 4: Communication algorithms vs average total energy before power constant fix

The results from after the fix can be seen in Figure 5. Here, we see a giant increase in the power consumed for the two centralized algorithms. For example, in a 60 node network *Min-hop Shallowest* uses 34 joules before the fix and 2129 joules after this fix. This change comes from the fact that the communication energy actually impacts the results. The two centralized algorithms also require much more communication than the decentralized algorithms because every message generated is sourced from the first node. This results in a great deal of message forwarding and energy consumption. The fix basically renders these algorithms ineffective. In Figure 6 we analyze the six remaining decentralized algorithms.

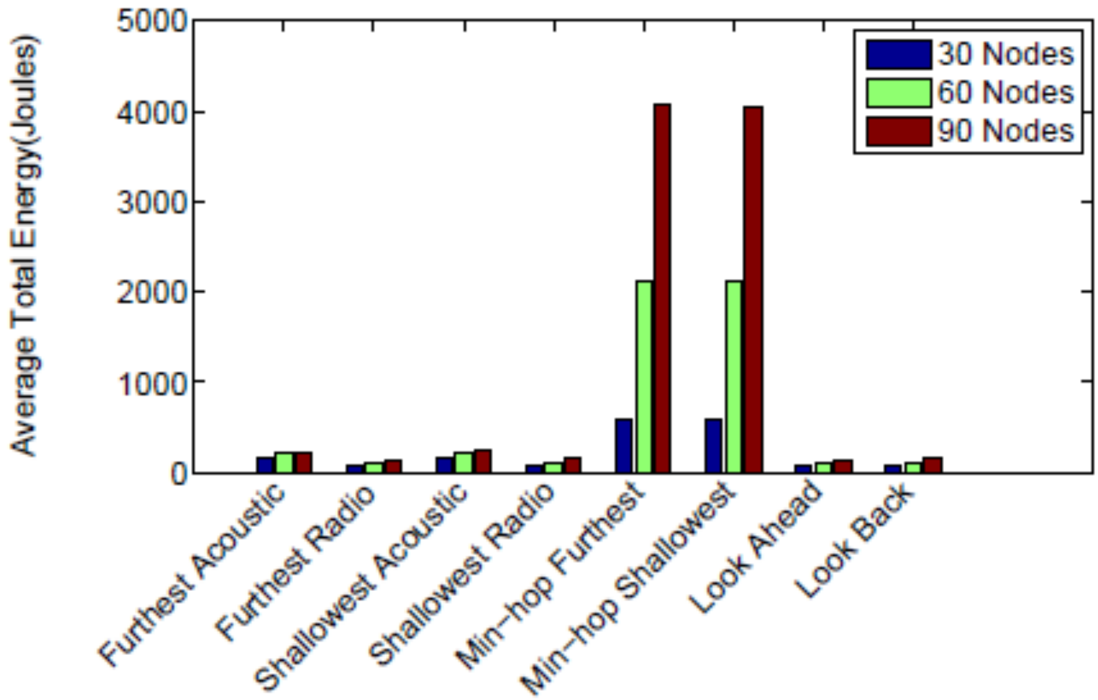

Figure 5: Communication algorithms vs average total energy after power constant fix

After removing the two centralized algorithms from the figure we are now able to see the results of the remaining algorithms. The ordering of the remaining algorithms is the same as it was before the fix.

3.2.2 Distance calculation error. There was another error that was found in the simulation environment, this time it involved the calculation of the distance between nodes. We noticed this while trying to expand the simulation environment to support 3D network topologies. The distance between nodes is used to generate two connectivity matrices: one for acoustic communication and one for radio communication. It is assumed that the surface of the water is calm and there are no waves. The connectivity matrices are used to

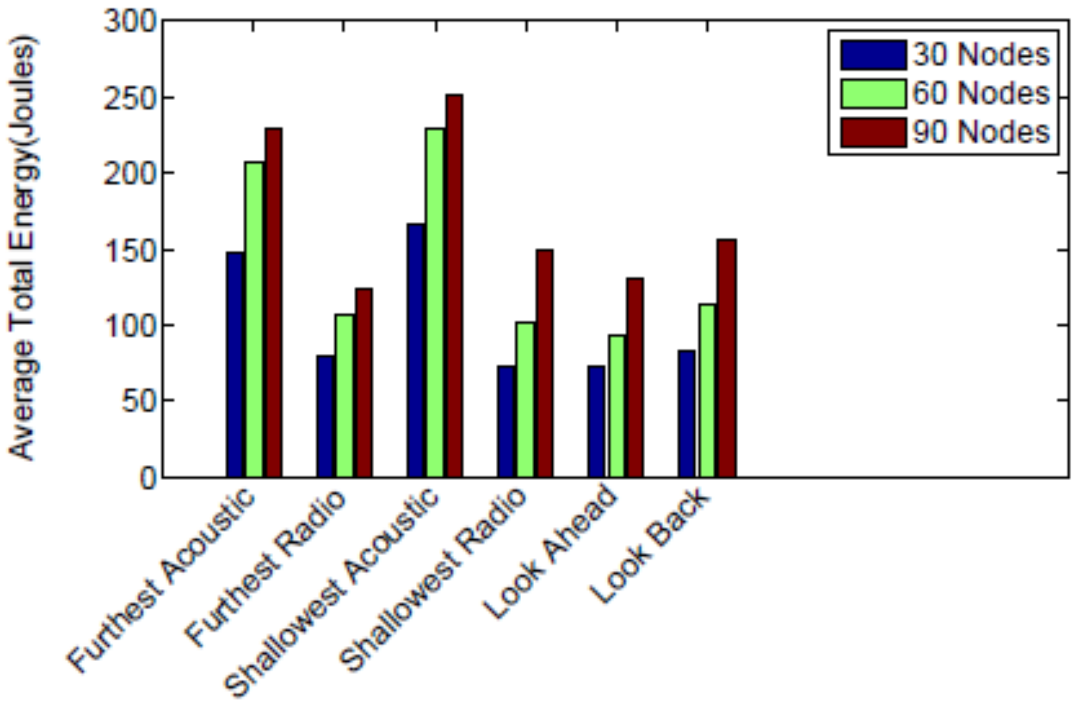

Figure 6: Decentralized communication algorithms vs average total energy after power constant fix

determine what nodes are within communication range of each other. The radio communication connectivity matrix was fine. The nodes need to rise to the surface of the water to communicate through radio so it only needs to use the euclidean xy distance between nodes. This is precisely what it did. However, the distance calculation to determine the acoustic communication connectivity matrix was incorrect. For acoustic communication the nodes are underwater and at different depths. The calculation should take into account the z distance (depth) and use the xyz euclidean distance between nodes to generate the acoustic connectivity matrix. Before the fix, depth was not taken into account, only the xy euclidean distance was used to generate the acoustic connectivity matrix. This ended up being a relatively minor error, as it had no effect on the previous results generated by O'Rourke. This is because, once the connectivity matrices are generated, the distance between nodes is ignored. Either the nodes can communicate or they cannot. The topologies used were randomly generated with the x distance between nodes being 30:60 meters, y distance set to 0, and z (depth) being 0:20 meters. At this point the network topologies were linear, which was why the y distance was 0. The maximum distance that two nodes could be apart was 63.2456 meters. The maximum acoustic communication range constant was set to 100 meters, as long as nodes were within 100 meters of each other they could communicate acoustically. This was modeled as a yes or no decision. With those settings there was never a case that the acoustic communication connectivity matrix was setup incorrectly. However, the problem could appear in corner cases. For example, if the maximum x distance between nodes was 99 meters instead of 60 meters the maximum potential distance between nodes would be 101 meters. In a case like this the connectivity matrix would incorrectly believe that those nodes were in communication range. After the fix, the acoustic communication connectivity matrix now takes into account depth.

3.2.3 Energy updates. The last major error found in the old simulation environment was incorrect acoustic send energy updates. In the simulation environment every time a packet of data is sent via acoustic communication it updates a variable that keeps track of the total energy consumed from sending data acoustically. However, the old simulator was updating the acoustic receive energy instead of the acoustic send energy. We noticed that the acoustic send energy was constant throughout all tests. The total energy used by each AquaNode in the simulation was correct, except the acoustic send energy was combined with the acoustic receive energy. This change did not affect the results seen by O'Rourke.

3.3 Acoustic Packet-Loss

We chose to extend the simulation environment to include acoustic packet-loss. We made this extension because underwater acoustic communication is real and significant. On the AquaNode platform our experimental results showed 44% packet-loss at typical communication range [6].

The rest of this section is as follows: first we discuss how acoustic packet-loss is implemented, second we run tests and analyze the results to confirm correct operation, and third we discuss potential extensions to the work.

3.3.1 Implementation. We implemented acoustic packet-loss by including a probability of the message not being received. If the message was not received then the transmitting node kept on resending until it was correctly received. This approach assumes that the message will eventually be successfully received. In the case where the packet of data is never successfully received, the simulator will infinitely loop until it is forced to stop. The rate at which acoustic packets are successfully transmitted is controlled by a variable ACOUSTIC_SUCCESS. This variable is initialized at the start of every simulation and is set by the user. Values can range from 0-1 with 0 being 0% success rate and 1 being 100% success rate. Every time an acoustic packet is sent, a variable tracking acoustic send energy is updated as well as a variable tracking total acoustic message propagation time. Equation 3.1 shows the acoustic send energy update equation. *ACOUSTIC_SEND_ENERGY* is the energy used per bit, *pkt_len* is the length of the data packet being transferred in bytes. *pkt_len* is multiplied by 8 to get the number of bits, and 24 represents the number of bits in the overhead.

$$E_{ACS} = E_{ACS} + ACOUSTIC\_SEND\_ENERGY * (24 + 8 * pkt\_len); \tag{3.1}$$

The constant *ACOUSTIC_SEND_ENERGY* is based on experimental results from the AquaNode that come directly from [6].

$$ACOUSTIC\_SEND\_ENERGY = .1136J \tag{3.2}$$

The acoustic message propagation time is updated based on the distance between the communicating nodes multiplied by a constant *SEND_DELAY*; this can be seen in Equation 3.4. The value for *SEND_DELAY* comes from [2]. As of now the acoustic message propagation time does not take into account the time that it would take for the network to verify a successful transmission. It is assumed that the transmitting node instantaneously knows whether or not the packet sent was successfully received or not. We make this assumption to simplify the calculation. The acoustic message propagation time was mainly used as a tool for verifying correct operation and to gain a basic understanding of the time it would take for the network calculate a communication path.

$$SEND\_DELAY = .00067s/m \tag{3.3}$$

$$T_{OAMP} = M_{OAMP} + N_{dist} * SEND\_DELAY \tag{3.4}$$

We added a couple of useful metrics to measure the total number of acoustic messages sent and the total number of failed acoustic messages. In the next section we analyze the simulation environment to ensure it operates as expected.

3.3.2 Analysis. In order to examine acoustic packet-loss, we designed and ran a series of four baseline tests. These tests examine: (1) the acoustic message count of the eight different communication algorithms with packet-loss, (2) the acoustic energy consumption of the six non-centralized communication algorithms with packet-loss, (3) the total energy consumption of the six non-centralized communication algorithms with packet-loss, (4) the average network energy consumption while varying network size with packet-loss. We did these series of tests to confirm that acoustic packet-loss was functioning as expected and then to see if it effected the ordering of the communication algorithms.

Acoustic message count with packet-loss. The first verification test verifies that the number of total acoustic packets sent increases as the acoustic packet-loss rate increases. In this baseline test we used a network size of 60 nodes, acoustic packet-loss rates of (0%, 44%, 80%), and averaged the results of 10 runs. We chose acoustic packet-loss rates of 0%, 44%, and 80% to gain an understanding of the affect of the variable in the best case scenario (0%), experimentally measured scenario (44%), and near worst case scenario (80%). Using 100% acoustic packet-loss renders acoustic communication completely useless and in turn breaks the simulation environment.

In all algorithms the trend is the same. As the acoustic packet-loss rate increases, the number of total acoustic packets sent increases and scales to expected values. For example, the algorithm *Greedy Shallowest Radio* sends 77 packets with 0% loss, 326 packets with 44% loss, and 1079 packets with 80% loss. We expect the amount of packets sent for 0% packet-loss be around 18.67% of the packets sent for 44% packet-loss and 6.67% of the

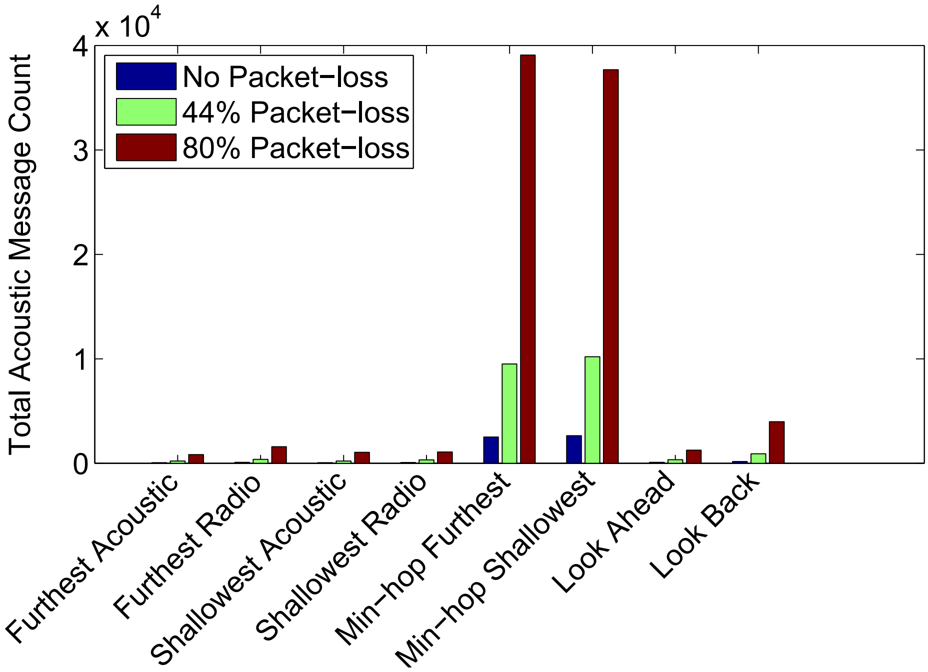

Figure 7: Algorithms vs number of total acoustic messages sent with different acoustic packet-success rates.

packets sent for 80% packet-loss. We expect these results because on average each node has three neighboring nodes within acoustic range and must successfully communicate with all of them. For example, at 80% packet-loss it takes five sent packets on average for a successful transmission. With three neighbors at five packets each, the transmitting node needs to send 15 messages on average for a series of successful trasmission to all three nodes. This results in 15 times more packets being sent for 80% packet-loss vs 0% packet-loss or 6.67% for the inverse. The results show that the amount of packets sent for 0% packet-loss are 23.6% of the number of packets sent for 44% packet-loss and 7.1% for 80% packet-loss. Figure 7 shows these results. Now that we know that the simulation environment is handling packet-loss correctly we now analyze the impact that it has on the network.

Acoustic energy consumption with packet-loss. The second baseline test compares acoustic packet-loss rates and the average amount of acoustic energy used per node in the network for six of the eight algorithms. We chose to remove the centralized algorithms because their energy results were so much larger than the decentralized algorithms. This baseline test had the same settings as the last, 60 node network size, acoustic packet-loss rates of (0%, 44%, 80%), and averaged results from 10 runs.

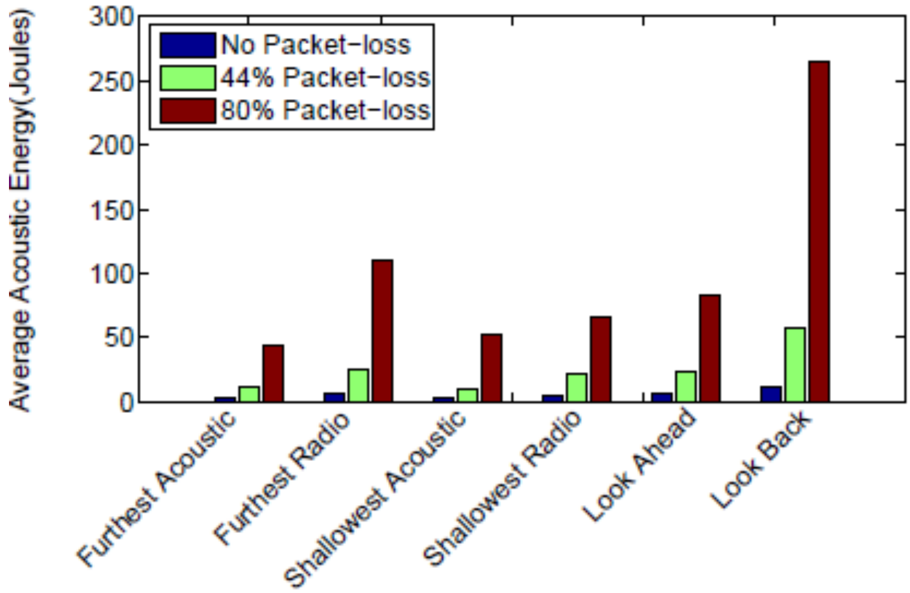

Figure 8: Non-centralized algorithms vs average acoustic energy with different acoustic packet-success rates.

The results can be seen in Figure 8. For all algorithms, average acoustic energy used per node increases as the acoustic packet-loss rate increases. It follows the same pattern as the first test. This is expected because the average acoustic energy used per node depends directly on the amount of packets being communicated. In terms of acoustic energy efficiency the two acoustic-centric communication algorithms are the most efficient. This is also expected because those algorithms are design to maximize acoustic communication efficiency.

Total energy consumption with packet-loss. The next test takes into account the energy efficiency of the AquaNode as a whole. The third baseline test is similar to the second. However, instead of comparing just the acoustic energy, the test considers all of the energy metrics. The total energy equation consists of processing energy, radio receive energy, acoustic receive energy, radio send energy, acoustic send energy, and movement energy.

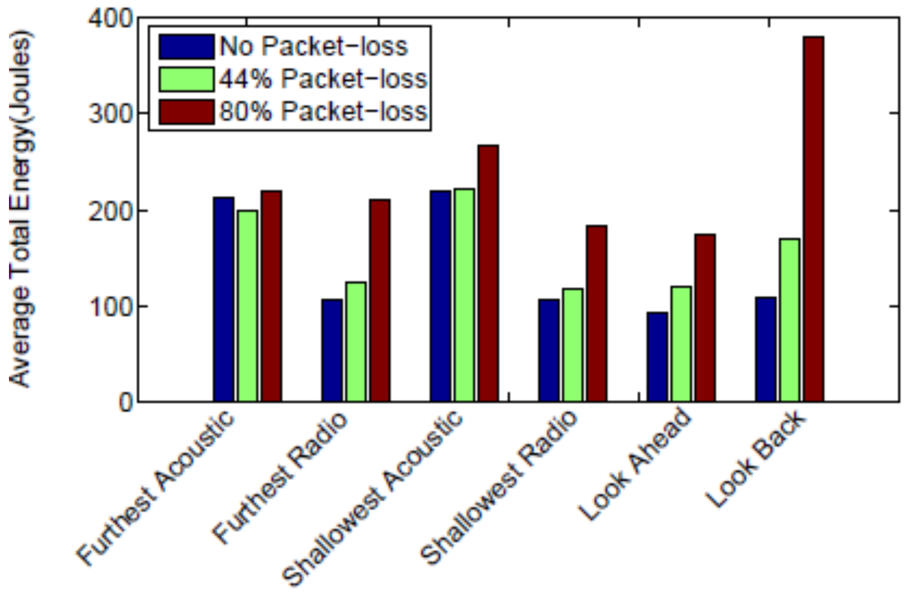

Figure 9: Non-centralized algorithms vs average total energy with different acoustic packet-success rates.

The resulting order of the algorithms in this baseline test is different from the last and can be seen in Figure 9. This is because all energy metrics were considered and acoustic send energy was only a small part of the total energy consumption. The most energy efficient algorithm overall was a toss-up between Shallowest Radio and Look Ahead. These results shows that acoustic packet-loss has a larger impact on the algorithms that use more acoustic communication.

Average network energy consumption with packet-loss. The fourth baseline test explored network size and its impact on average network energy consumption for 0% and 50% acoustic packet-loss. The configuration for this baseline was network sizes between 5-100 nodes in five node increments, 0% and 50% acoustic packet-loss, and results averaged from 10 runs.

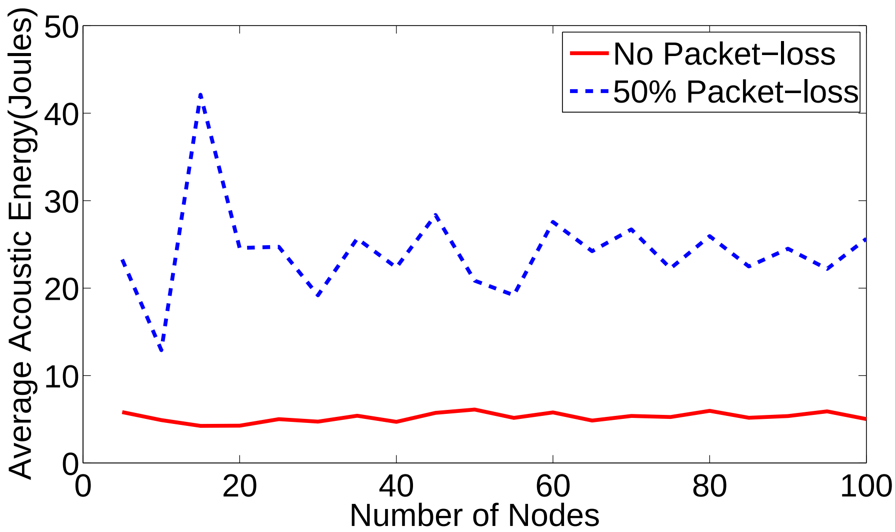

Figure 10: Average network acoustic energy consumption per node comparing 0% and 50% acoustic packet-loss

We first analyzed the impact that acoustic packet-loss had on the average network acoustic energy consumption. The results had a few data spikes; however they were clear enough to extract a general trend. With 0% acoustic packet-loss the average acoustic energy for all network sizes was approximately 5 joules. For 50% acoustic packet-loss it was around 25 joules. Figure 10 displays these results. Network size does not have an impact on average acoustic energy used.

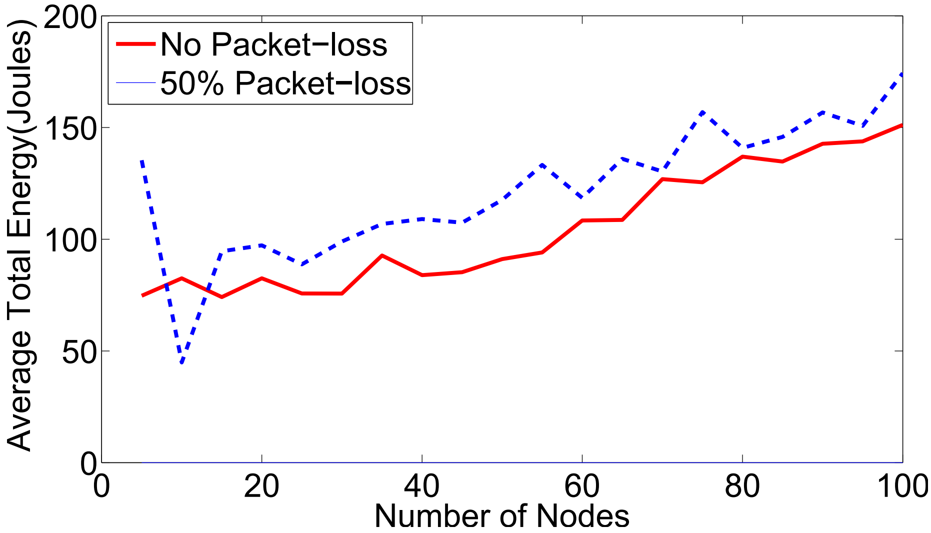

Figure 11: Average network total energy consumption per node comparing 0% and 50% acoustic packet-loss

Next, in Figure 11 we reexamine the results of the previous test to analyze the average network total energy consumption. These results also have large data spikes, but the general trend is still quite clear. As network size increases, the average total energy used linearly increases. As expected, 0% acoustic packet-loss results in less energy used compared to 50% acoustic packet-loss. Overall the difference between the two rates is constant and the two lines travel in a parallel fashion.

3.4 3D Network Topologies

The initial version of the MATLAB simulator created network topologies in a linear fashion. There were two dimensions: x (the nodes euclidean x distance away from the start node) and z (node depth). As not all underwater networks are in this configuration, it made sense to make the addition of 3D (three dimensional) network topologies. In this section we will discuss the addition of 3D network topologies.

The remainder of this section will discuss the implementation of 3D network topologies to the simulation environment, how we verified correct operation, and potential future work to make further three dimensional topology improvements.

3.4.1 Implementation. To add 3D network topologies to the simulation environment we needed to modify the way the topologies were generated and how the connectivity matrices were generated. The initial simulator nodes placed nodes in a linear fashion along the xz-plane. This could be done in a pseudo random fashion according to user specified limits or by loading in a preexisting topology.

To extend the simulation environment to include 3D we added the y-axis. In the pseudo random topology generation the position of each node is generated one by one and the x, y, and z coordinates are created. For the node's x position, the user can specify the minimum and maximum euclidean x distance between nodes. For example, if the minimum x distance is 30 meters and the maximum distance is 60 meters then the next node generated will be between 30 and 60 meters away from the previous node along the x axis. For the y positioning the user can specify a minimum and maximum value that the node will be placed between. For example, if the minimum value is 30 meters and the maximum value is 60 meters the y position of the node will be between 30 and 60 meters along the y-axis. The z-axis represents depth; 0 meters is the surface and a negative z position is how deep the node is. The z positioning can vary from a user specified minimum and maximum depth. The z position generated is the initial position of the sensor node.

3.4.2 Verification. We now verify correct operation of 3D topologies and analyze their impact on the ordering of the communication algorithms.

In order to verify that the 3D topologies were being generated correctly we examined the node positions generated. Figure 12(a) shows a typical 2D topology and Figure 12(b) shows a typical 3D topology. In the 2D topology we see that all nodes are in a plane where the y position is 0. In the 3D topology we see that the nodes vary between 30 and 60 meters along the y-axis.

Now that we know 3D topologies are being correctly generated we now analyze any changes in the ordering of the communication algorithms. In these tests we use network sizes of (30, 60, and 90) nodes and average the results from 10 runs. We ignore the centralized communication algorithms since we determined they were ineffective in Section 3.2.

Figure 13 shows the total average energy consumption for all decentralized communication algorithms in 2D topologies and Figure 14 shows 3D. The results are nearly identical. The small differences come from variations from the fact that the network topologies were not the same. In each test the results from 10 different network topologies were averaged. This type of 3D network topology does not change the ordering of the communication algorithms.

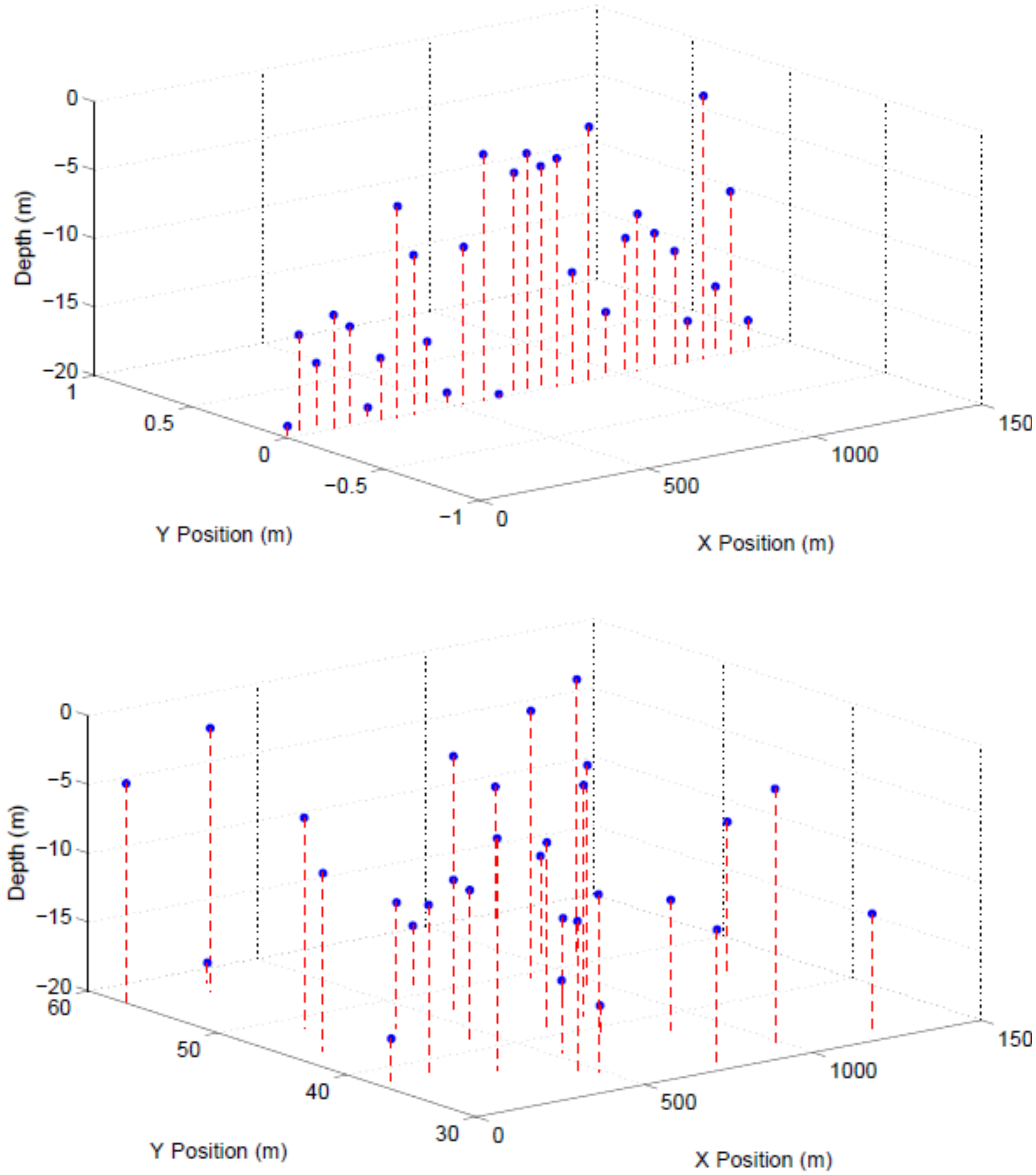

Figure 12: (a) 2D and (b) 3D Network Topologies

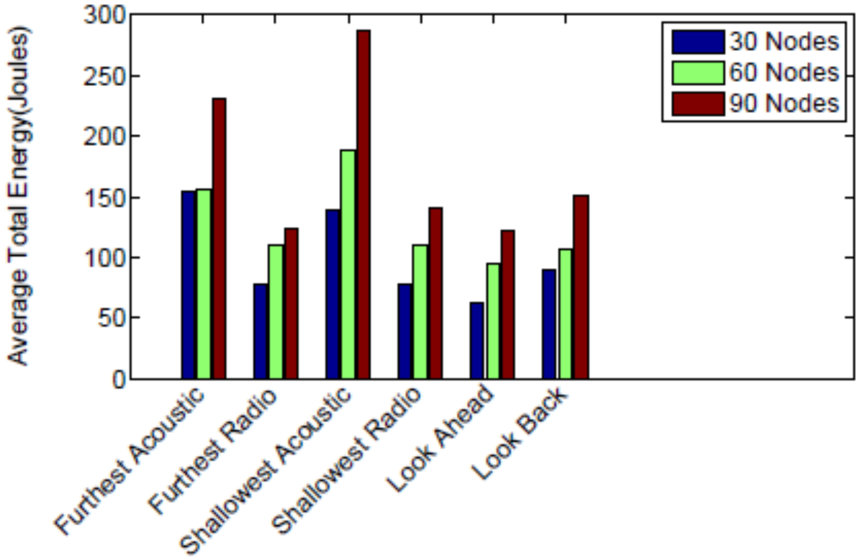

Figure 13: Total average energy for decentralized communication algorithms in 2D topologies

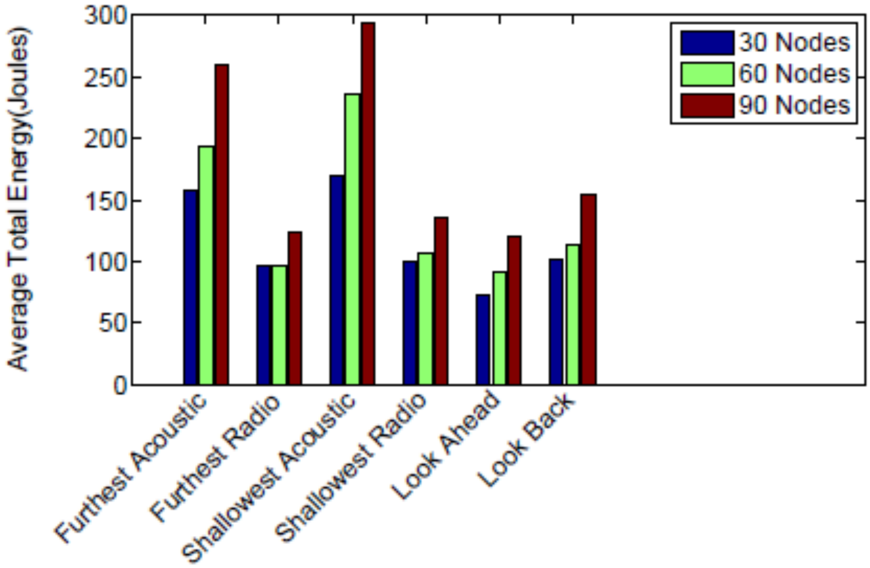

Figure 14: Total average energy for decentralized communication algorithms in 3D topologies

Chapter 4: UAV Data Muling

4.1 Overview

This chapter explores the viability of a UAV as a data mule in an underwater sensor network that utilizes AquaNodes. The idea is that the AquaNodes will rise to the surface on a synchronized schedule, the UAV will fly in to collect the data, and once data is collected the AquaNodes will dive back to their original depth and resume sensing. The UAV will fly in from a base station, traverse the network, hover above nodes to collect data, and then return back to base. The UAV and AquaNode communicate via 802.15 radio. This network configuration offers an alternative method for extracting data out of a deployed underwater sensor network. We use field experiments and simulation to determine the potential benefits and limitations of our proposed system.

The rest of this chapter is in the following order. Section 4.2 describes the hardware of UAV system and the underwater sensor network used in our experiments. Section 4.3 discusses the test configurations and methodology of our three field experiments to characterize radio communication, the results of those experiments, and analysis of the results. Section 4.4 characterizes the energy of the UAV and the sensor network then analyzes the energy consumption of system while varying key parameters. Section 4.5 analyzes the combination of communication and energy consumption results to deduce system policies.

4.2 System Overview

We use two main systems in our experiments, the first is a custom water node, Figure 15, and the second is an Ascending Technologies Hummingbird quad-copter, Figure 16.

Figure 15: Water node used in field experiments.

The custom water node is based on the AquaNode platform that Carrick Detweiler developed and discussed in Section 2.3. At the time of our testing the AquaNodes were non-functional so we built a temporary solution that mimicked the necessary characteristics of the AquaNode. Due to the nature of our experiments, the characteristics that we needed to inherit from the AquaNode were surface radio communication and the ability to float on the surface of the water. The water node enclosure was built from a Tupperware container, silicon caulking, duct tape, rope, and foam. The internals of the water node were comprised of a Seeeduino Stalker v2.3, 802.15 radio (same radio that the AquaNodes now use), 4 AA battery pack, and a micro SD card. The internals can be seen in Figure 17. The Seeeduino Stalker v2.3 was programmed in C and setup to continuously send packets of data via 802.15 radio that contained a sequence ID and device ID; it also logged all data sent and received on the SD card.

Figure 16: Ascending Technologies Hummingbird

We use an Ascending Technologies Hummingbird quad-copter UAV [1]. It has payload capabilities of 200g, a flight time of approximately 20 minutes, and can fly at over 14m/s. Outdoors, the Hummingbird uses GPS to maintain position and a pressure sensor altimeter for height control. This UAV can be controlled by a remote ground station via 802.15 radio or by using a small on-board computer for full autonomy. In this experiment, we controlled it remotely, added a secondary 802.15 radio (identical to the water node's) to communicate with the nodes, and Seeeduino Stalker v2.3 for on-board data logging. The Seeeduino Stalker v2.3 and the secondary 802.15 radio were configured in a fashion similar to the water node; the difference was that the UAV version was setup to reply to packets

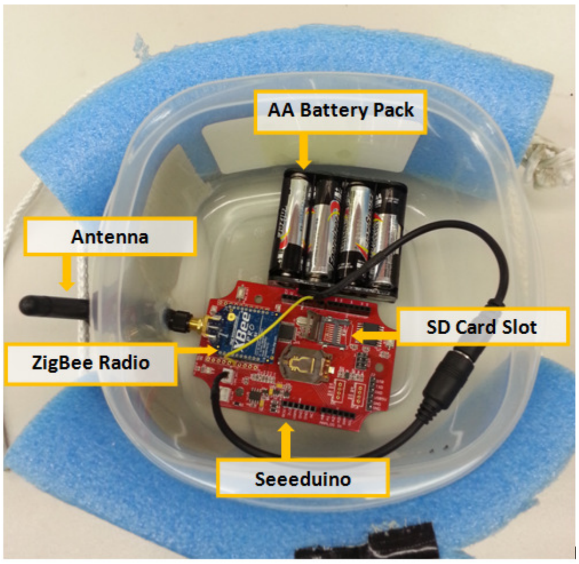

Figure 17: Water node internal components.

received.

4.3 Communication

In this section we characterize the radio communication between an UAV and an underwater sensor network with the UAV acting as a data mule. We perform three field experiments: (1) measure communication success rates between the UAV and a surfaced sensor node, (2) measure communication success rates between two surfaced underwater sensor nodes, and (3) measure communication success rates between two underwater sensor nodes on land. The first two experiments are directly used to characterize the radio communication within our desired system while the third experiment is used as a comparison between radio communication over land vs over water. After performing the experiments, we analyze the combined results from each.

4.3.1 UAV to water node radio communication on water. Our first experiment examines ranges, altitudes, and success rates between a surfaced underwater sensor node and an UAV.

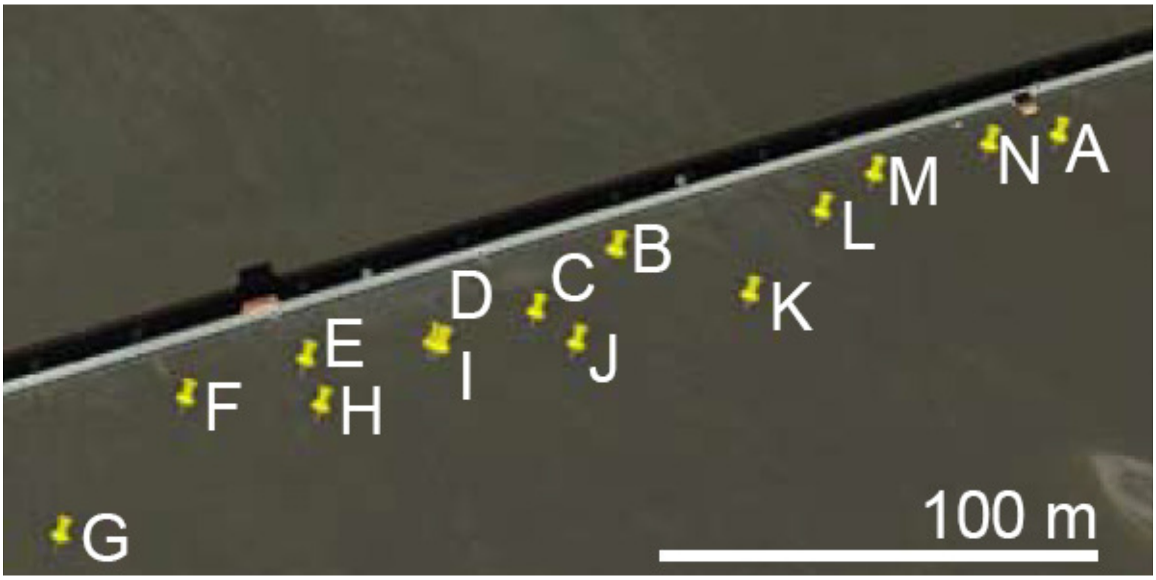

Figure 18: UAV to water node test configuration. The "A" marker represents the location of the static water node and rest represent the main locations of the UAV. Figure courtesy of Google Earth.

Test configuration. We performed this field experiment on the Platte River, Nebraska, USA; Figure 18 shows the experiment configuration. The bridge that we conducted the experiment on was about 20 meters above the surface of the Platte River. We placed our simplified AquaNode at location "A" seen in Figure 18 and tethered it to the bridge so that it would not move significantly.

For the UAV, we manually flew it at different heights and distances away from the static water node. Nodes "B" through "N" in Figure 18 outline the locations at which we varied the altitude. We started the experiment at around 100 meters (euclidean xy distance) between the static water node and the UAV. We then manually flew away from the static water node at fixed intervals along the bridge. At each interval location, we steadily decreased the altitude until the UAV was 3 to 5 meters above the surface of the river, increased the altitude until we noticed a significant decrease in the packet-success rate, and then returned the UAV to an altitude level with the bridge. We continued this pattern at increasing distances away from the static water node until we reached a point where we saw significant packet loss at all altitude levels. This happened at location "G", approximately 250 meters away. After reaching this point, we moved back towards the static node and repeated the pattern to gain additional data points. Since we started the experiment at around 100 meters away from the static water node, we continued approaching the static water node until we were approximately 20 meters away.

Throughout the experiment, we recorded the GPS location of the UAV, the packets sent from the static node, the packets received at the UAV, and the packets received at a secondary node used for confirming behaviors. The static water node continually broadcasted packets of data throughout the experiment, the packet of data consisted of a unique device ID and a packet sequence number (that incremented by 1 after each transmission). If the UAV received one of these broadcasted packets, it would send a similar packet with a different device ID but the same packet sequence number. This allowed us to monitor the communication in real-time using laptops outfitted with 802.15 radios. Both the static water node and the UAV recorded every packet sent and received on micro SD cards.

Test results. Figures 19 and 20 depict the results of the experiment. Figure 19 indicates the locations where the UAV did and did not receive packets. The x-axis is the horizontal distance between the UAV and the static water node; this is the Euclidean distance in the xy plane as the UAV did have some minor oscillations in y positioning due to manual flying and winds. The y-axis is the altitude of the UAV above the river surface. Blue circles indicate received locations while red triangles indicate that the UAV did not receive any packets. From this figure, we can see a clear region of success.

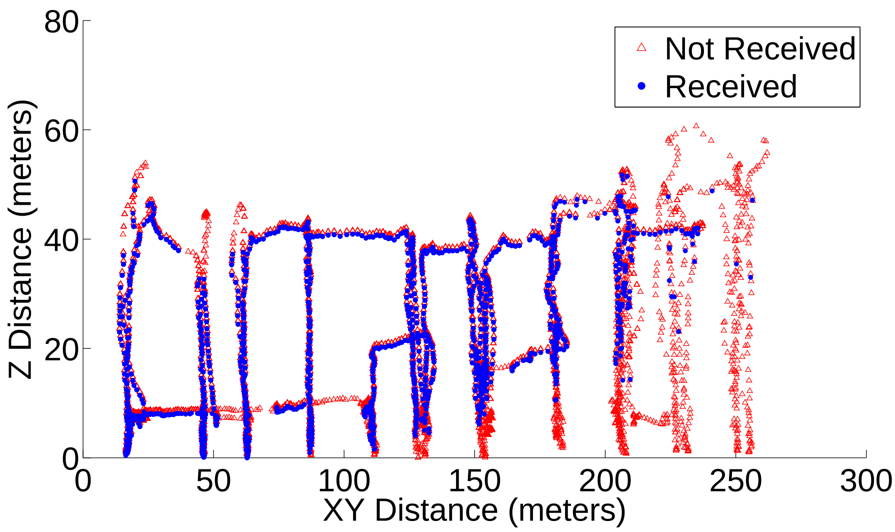

Figure 19: Radio sent and received messages between water node and UAV at different locations.

To better understand this region, we can examine Figure 20, which shows the packet-success rate of radio communication between the UAV and the water node at different vertical and horizontal distances. The axes are the same as the prior figure where the x-axis is the horizontal distance between the UAV and the static water node, and the y-axis is the altitude of the UAV above the river surface. The different colors represent the packet-success rate of radio communication with white-yellow colors signifying high success rates at that location and black colors signifying low to zero success rates.

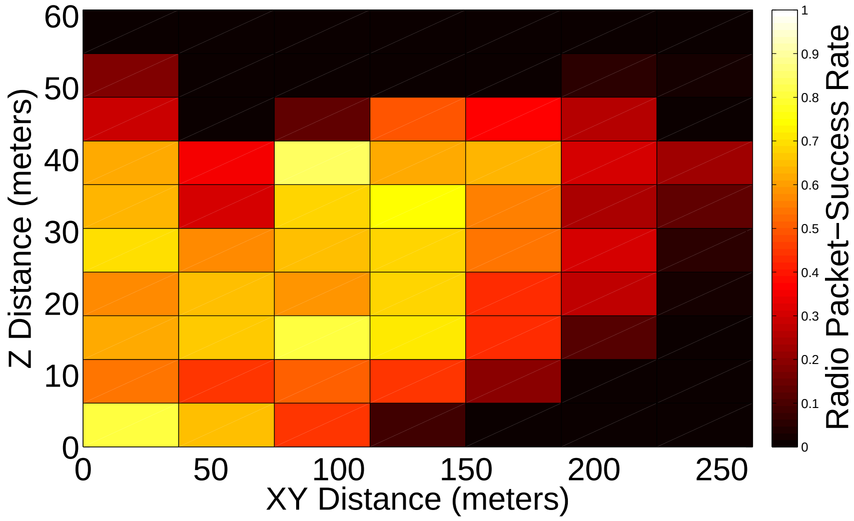

Figure 20: Radio communication packet-success rates between water node and UAV at different distances.

As this figure demonstrates, near the surface of the water at altitudes under 5 meters, the systems have a limited horizontal communication range. In this altitude region, we see a rapid decline in packet-success rate as the horizontal distance increases with the rate dropping below 50% at 75 meters. However, as the altitude of the UAV increases, the decline in packet-success rate slows down and the two nodes are able to successfully communicate at further horizontal distances. This increase holds until the UAV approaches 43 meters above the water surface at which point the packet-success rate drops off sharply. Horizontally, we still see communication until after 249 meters at an altitude of 32 meters; in this location we still see 20% success rates. Overall, our system achieved the peak packet-success rate of 94% at approximately 174 meters horizontally at 38 meters above the surface of the water.

4.3.2 Water node to water node radio communication on water.

Our second field experiment determines the communication ranges and packet-success rates between two surfaced nodes communicating via 802.15 radio.

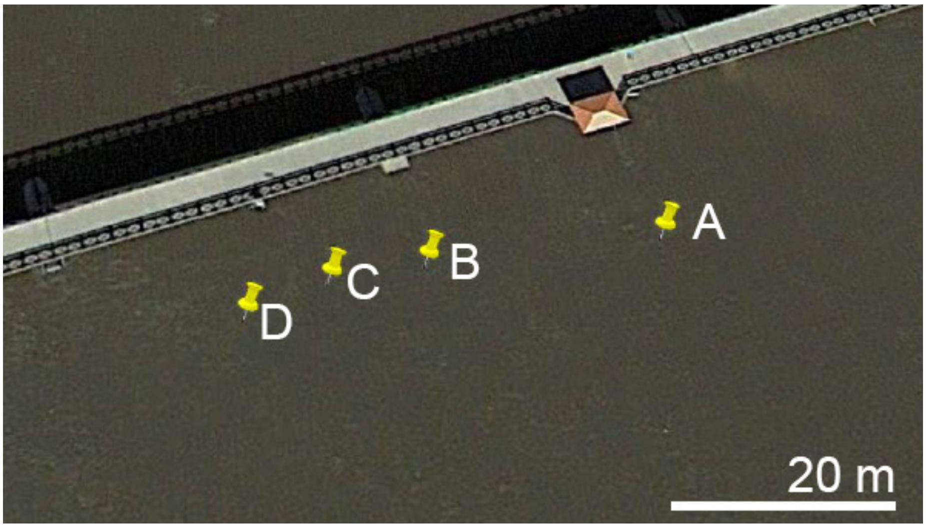

Figure 21: Water node to water node test configuration. Figure courtesy of Google Earth.

Test configuration. For this field experiment, we had two identical water nodes that floated on the surface. The first node remained static at position "A" depicted in Figure 21 while the second node moved from positions "B","C", and "D" also shown in Figure 21; these coordinates were measured by the same GPS used in the first experiment. The first position, "B," was 17 meters away from the static node; we then moved the node at discrete intervals until the two nodes could no longer communicate. At each possible position, the second node remained for approximately 5 minutes, logging data on packets received and retransmitting received packets to ensure bidirectional communication.

Test results. Figure 22 shows the results for the water node to water node radio communication. The two significant drops in packet-success rate at 25.8 and 32 meters indicate locations with limited measurements; as the node moved through the location on the way to the next fixed location, a small number of packets were sent but not received. As the figure shows, the nodes already experience heavy packet loss at 17 meters, with a packet-success rate of only 50%. While we see significant variation over the full distance, it never

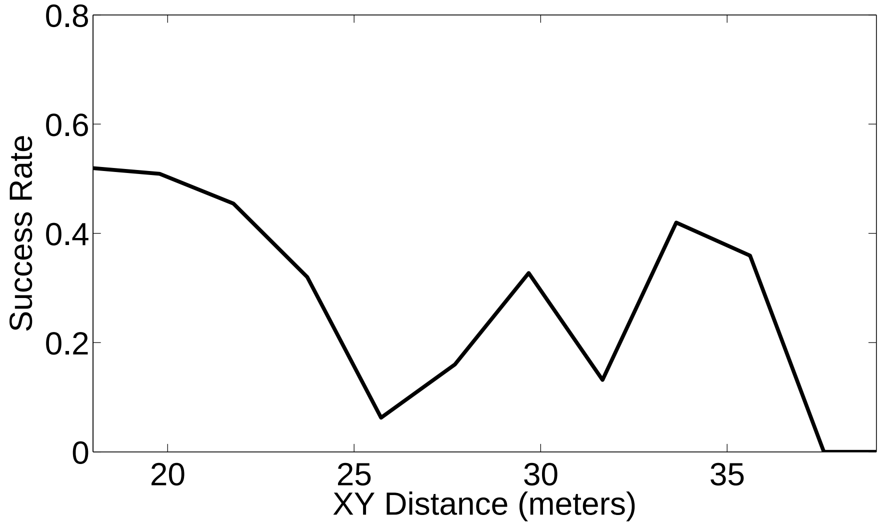

Figure 22: Radio communication packet-success rates between two water nodes on the surface of the water at different distances.

improves above 50% and, on average, degrades in performance. Once the distance between nodes increases past 37 meters, the nodes can no longer communicate.

4.3.3 Node to node radio communication on land. Our third field experiment determines the communication ranges and packet-success rates between two water nodes on land communicating via 802.15 radio. This experiment was done on a field in Stockton, CA and was used as a comparison to the water node to water node radio communication on water. We wanted to see if there were any differences in the communication characteristics between land and on the surface of the water.

Test Configuration. In this field experiment we used the same two water nodes used in the second field experiment. The main difference being that we conducted it on land instead of over the water. The configuration can be seen in Figure 23. The first node was static at position "A" while the second node moved from positions "B" through "I"; these coordinates were measured by a different GPS then the one used in the first two

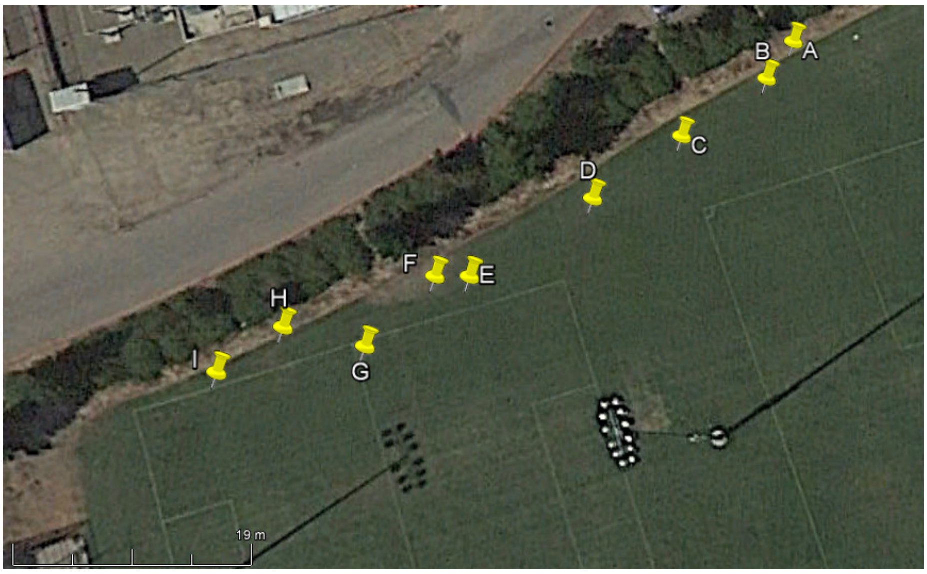

Figure 23: Node to node test configuration on land. Figure courtesy of Google Earth.

experiments. This is not an issue because we only consider relative positions.

We performed two iterations of ground experiments. On the first we placed the nodes directly on the ground and on the second we placed the nodes 0.1 meters off the ground. We wanted to see if raising the node's height by a small amount would increase communication range. In each iteration we moved the node further away at set intervals until the two nodes could no longer communicate. At each possible position, the second node remained for approximately 5 minutes, logging data on packets received and retransmitting received packets to ensure bidirectional communication.

Test results. Figure 24 shows the results for the node to node radio communication on land. The x axis is the euclidean xy distance between the two nodes and y axis is packet-success rate. The solid red line represents the results from the ground level iteration of the experiment and the dashed blue line represents the results from the iteration where the nodes were placed 0.1 meters off the ground.

As the figure shows, the nodes show less than 30% packet-loss up until approximately

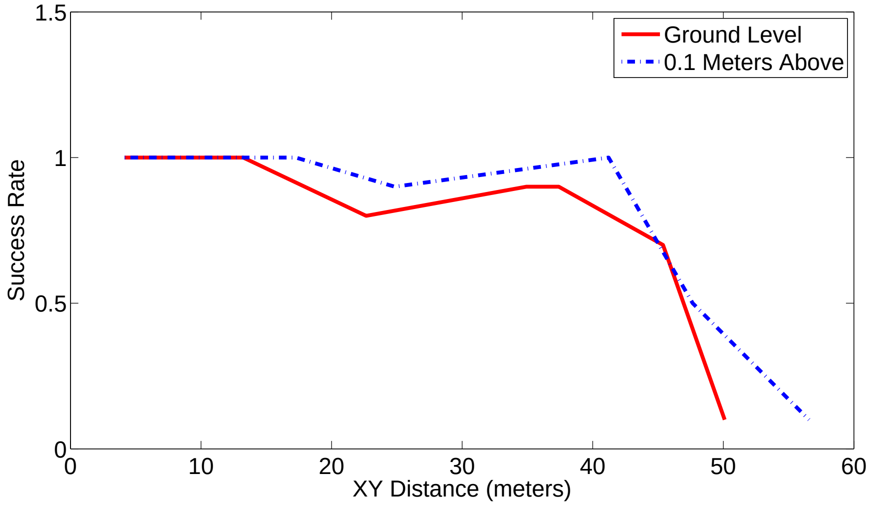

Figure 24: Radio communication packet-success rates between two water nodes on land at different distances.

45 meters. At this point both iterations of tests show a rapid decline in packet-success rates. When placed directly on the ground the two nodes were unable to communicate after 50 meters and when placed 0.1 meters above the ground the two nodes were unable to communicate after 57 meters. On ground we see higher packet-success rates and further distances then we do over water. This means that we do in fact see different communication characteristics on land than we do over water.

4.3.4 Water analysis. We now analyze the combined results to understand possible topologies and configurations when using an UAV with an underwater sensor network.

Given the limited node-to-node ranges over water, it makes sense to use an UAV within the system to expand the coverage area. Depending on the UAV flight altitude, we see a spectrum of different packet-success rates. Table 3 summarizes the optimal distance between the UAV and surfaced nodes for given altitudes as well as the packet-success rate at that distance. For example, if the UAV flies at an altitude of 43 meters, its flight plan needs to pass within 212 meters of each node to achieve 75% packet-success rate. If instead, the UAV flies lower at 14 meters, it needs to pass within 137 meters to achieve 70% packet-success rate.

Table 3: Optimal radio communication ranges: water surface to UAV

Altitude (m)	**Optimal XY Distance (m)**	**Success-Rate Threshold**
Water surface	0-34	40%
6	0-87	60%
14	0-137	70%
25	0-149	75%
38	0-174	94%
43	0-212	75%
49	0-237	10%
50+	none	0%

With this information, we can analyze the options for maximizing communication success while also maximizing network coverage area. Table 4 outlines the optimal UAV altitudes to achieve this for two different network configurations.

The first configuration requires the underwater nodes to reach at least one other node via radio such that internode communication exists in the system. In this configuration, the nodes use an evenly spaced, rectangular, grid topology where the UAV communicates with at least two nodes. Based on our field experiment data, the optimal node spacing to maximize network coverage area is 34 meters and the optimal UAV altitude to maximize communication success can range from 6-43 meters. This allows for internode communication with 42% packet-success rate and node to UAV communication with 54%-81% packet-success rate depending on the altitude.

Table 4: Optimal UAV altitudes and optimal node spacings for maximizing network coverage

Internode communication	**Optimal UAV Altitudes (m)**	**Optimal Node Spacing (m)**
Yes	6-43	34
No	43	424

The second network configuration removes the requirement of internode communication between surfaced nodes. In other words, nodes can be spaced further apart but as a consequence, they lose their capability to communicate via radio to each other. In this configuration, only the UAV provides surface communication for the nodes, which still use an evenly spaced, rectangular, grid topology. The UAV then flies in a pattern that bisects pairs of nodes so that it can gather data from two nodes at each hover spot. Here, our field experiment data indicates that the optimal node spacing for maximizing network coverage area is 424 meters and the optimal UAV altitude is 43 meters. This means that at each hover spot, the UAV is 212 meters away from each of the two nodes it is data muling from and at 43 meters the UAV achieves 75% packet-success rate.

Table 5 describes how these results relates to possible coverage areas of the network. For a network of 25 nodes, requiring internode communication allows the network to cover an area of 544 m^2; relaxing that requirement allows coverage of 6784 m^2. Overall, adding an UAV to the system allows the network coverage to expand by 1247% with the same amount of nodes. This is a significant increase in coverage area and can be useful when sensor nodes are limited and a large area needs to be covered.

4.3.5 Land vs water analysis.

For the node to node communication over water compared to over land we see that over land we attained higher packet-success rates and

Table 5: Maximum area covered by specified network

	Area of Coverage (Meters^2)	
Network Configuration	**Internode Communication**	**No Internode Communication**
25 Nodes	544	6784
64 Nodes	1666	20776
100 Nodes	2754	34344

further ranges. For this analysis we will be referring to the water node to node experiment as “Test One,” the land ground level node to node experiment as “Test Two,” and the 0.1 meter node to node experiment as “Test Three.” The differences between Tests One, Two, and Three are highlighted in Table 6. Between 17 and 37 meters, Test Two and Test Three show approximately double the packet-success rate of Test One. At 37 meters, where in Test One the two water nodes are unable to communicate; we see success rates of 90% and 97% for Test Two and Test Three respectively. At 45 meters we start to see a decline in the packet-success rates of Tests Two and Test Three. Test Two shows 70% packet-success rate and the Test Three shows 75% packet-success rate. After 50 meters the nodes in Test Two are unable to communicate while the nodes in Test Three still have 40% packet-success rate. After 57 meters the nodes in Test Three are unable to communicate.

The reason behind the low range between nodes in test one is still unknown. However, if it is because of rough waters coming between the direct communication path, then raising the antenna could help to alleviate this problem. We can see that by raising the node or antenna on land even by 0.1 meters, we increase communication range between the nodes. We can also see from Table 3 that the UAV flying at higher altitudes increases communication range between node and UAV. The combination of these results seems to imply that raising the antenna of the water node higher out of the water would increase

Table 6: Radio communication success-rates at different distances over water and land

XY Distance (m)	**Water "test one"**	**Ground "test two"**	**0.1 Meters "test two"**
17	50%	90%	100%
30	33%	85%	93%
34	42%	90%	95%
37	0%	90%	97%
45	0%	70%	75%
50	0%	10%	40%
57	0%	0%	10%
58+	0%	0%	0%

communication range at the surface of the water and at lower altitudes.

4.4 Energy

After exploring the communication aspects of the system, we now examine the energy usage. We first identify experimentally measured parameters that characterize the energy and then define an energy model to explore tradeoffs between network size, distance between nodes, the amount of data each node has, and how far the network is from the UAV base station. With this model, we explore the impact that these parameters have on the energy usage.

4.4.1 Energy characterization. To understand the energy use of the combined UAV and sensor network system, we measure key parameters needed for characterization. Table 7 outlines these parameters for the UAV, the 802.15 radio, and the AquaNode sensor node.

We measured the UAV data through flight tests and post-flight analysis. These measurements indicate the optimal flying speed of the UAV, the energy used to travel at that

Table 7: Measured energy characterization parameters

Energy Parameters	**Value**
UAV Power at 10m/s (*travel_power*)	115 Watts
UAV Hover Power (*hover_power*)	92.3 Watts
UAV Travel Distance Per UAV Battery	7300 Meters
UAV Battery Capacity	83900 Watt-Hours
Radio Transmit Power (*max_transmit*)	63 mWatts
Radio Max Physical Data-rate	57600 Bits/Sec
Radio Max Transmit Data-rate (*radio_true_data_rate*)	4800 Bits/Sec
AquaNode Battery Capacity	172800 Watt-Hours
AquaNode Depth Adjustment Power	15 Joules/Meter
AquaNode Acoustic Receive	0.063 Joules/Bit
AquaNode Acoustic Transmit	0.1136 Joules/Bit
AquaNode Flash Memory	512000 Bits

speed, and the power used when hovering. In addition, we determine, for a given battery capacity, the UAV travel distance on a single battery.

The *radio_true_data_rate* parameter is based on lab measurements. The radios operate at the maximum transmit power ($max\_transmit = 63mW$) in order to maximize range. Although the physical layer transmits at 57600 bits per second (bps), the ZigBee radios transmit each packet multiple times to increase the likelihood of successful reception, and our experiments showed a true transmission data rate of 4800 *bps*. For the AquaNode, the radio is a new feature; our previous work experimentally determined the parameters for depth adjustment and acoustic communication [6].

4.4.2 Simulation setup. Now that we have the energy parameters, we now model the energy system of the UAV in order to understand trade-offs. We created the simulation environment in MATLAB and use it to understand energy consumption of the UAV depending on the underwater sensor network configuration.

For the UAV to communicate with the underwater sensor network, the UAV needs to travel to the network, traverse the network, hover above nodes, transmit and receive data via radio, and then fly back. We assume that no communication with the sensor network occurs while the UAV is traveling. One of the main purposes of the energy model is to ensure that the UAV has enough energy to return home safely.

The simulation run-time is as follows. First, the energy constants are initialized. Second, the network topology is generated based on the under defined variables *numberNodes* and *nodeSpacing*. Third, each node is assigned the amount of data that needs to be muled. This value is based on the user defined variable *dataPerNode*. Fourth, the UAVs total distance traveled is calculated. The total travel distance varies depending on the UAVs flight path. Fifth, the total amount of time that the UAV hovers is calculated. Finally, the total UAV energy consumption is calculated.

When running the simulator there are four main variables that the user has control over. These are the number of nodes in the network (*numberNodes*), the amount of space between nodes (*nodeSpacing*), the amount of data per node (*dataPerNode*), and the distance out to the network from the UAVs base station (*networkDistance*). By varying these variables and utilizing the experimentally measured energy values outlined in Table 7, we are able to analyze the impact that each variable has on the energy consumption of the UAV.

The simulation environment outputs a number of useful values for the UAV. These include total energy consumed, hover energy, travel energy, travel energy, radio communication energy, remaining battery, total distance traveled, and hover time.

Now that we know the basics of the simulator we examine the UAV energy usage equations.

4.4.3 Energy model. In our system configuration the total energy expended by the UAV depends on the energy required to travel a certain distance, the energy required to hover for a certain amount of time, and the energy required for radio communication. The following equation models the overall energy expenditure of the UAV system:

$$E_{UAV} = E_{travel} + E_{hover} + E_{radio} \tag{4.1}$$

The first parameter, travel energy, depends on the energy the UAV requires to fly out to the network, traverse the network, and return to base. This depends on the UAV power at 10 m/s ($travel\_power$) and the total meters traveled (D); thus the UAV travel energy is $E_{travel} = travel\_energy * D$.

The second parameter of the UAV energy equation is hover energy. Hover energy depends on how long the UAV hovers and the power required; the UAV hover energy is $E_{hover} = hover\_power * hover\_time$. Hover time depends on the amount of data (num_bits) that it needs to collect and how fast the data can be collected; the resulting hover time is $hover\_time = num\_bits/radio\_true\_data\_rate$. The data collection speed and hover power used are the experimentally measured parameters from Table 7.

There are three main reasons why we model the hover energy. First, the hover power is different than the power required to fly at 10 m/s, 92.3 vs 115 watts respectively. Second, if the network is following the second network configuration discussed in 4.3.4, where the nodes are spaced 424 meters apart and the UAV is flying in the pattern that bisects nodes, the area that the UAV needs to be in to communicate with each pair of nodes is relatively small since the UAV is already near the maximum effective radio range. This network configuration does not give the UAV a chance to execute a fly by (the situation where the UAV does not stop to data mule) and the UAV must hover to data mule. Fly byes would be only be possible in certain situations, based on our field experiment data and experimentally measured parameters, the maximum amount of data that a UAV could mule from a node while traveling in a straight line directly over the node would be 203520 bits. In this situation the UAV is traveling at 10 meters per second in a straight line over the node and there are 424 meters that the UAV can communicate with the node, allowing 42.4 seconds of connectivity. Since the maximum radio transmission rate is 4800 bits per second, the maximum amount of data that the UAV could mule in this time frame is 203520 bits. The main limitation for this approach is if the node contains more than 203520 bits (the AquaNode has 512000 bits of flash memory) then the time it takes for one fly by would not be long enough to collect all the data off the node so the UAV would need to hover anyway. Third, it does not hurt to be able to model the hover energy, even if there are cases where hovering may not be needed. Future work may involve analyzing the energy consumption of a UAV doing data muling fly byes.

The third parameter is radio energy. The amount of energy the radio consumes depends on the data-rate, the amount of data that is being transmitted, and the power consumed. The maximum transmit data-rate of radio is 4800 bits per second and the radio transmit power is 63 mWatts, as seen in Table 7; the resulting radio energy equation is $E_{radio} = (max\_transmit/radio\_true\_data\_rate) * num\_bits$.

With these three parameters of the system energy use defined, we now have an overall energy model of the UAV system. In the analysis section we explore varying the number of nodes in the sensor network, distance between nodes in the sensor network, distance out to the sensor network, and time spend hovering to collect data at each node while leaving the transmission data rate fixed at the experimentally measured value.

4.4.4 Energy analysis. Given the energy model, we now explore the impacts of the variables on the system topology.

The key variables defining the energy of the UAV are: number of nodes in the sensor network, distance between nodes in the sensor network, distance out to the sensor network, and time spend hovering to collect data at each node.

We examine the impacts of each of these parameters on the UAV energy; skewing them one at a time while fixing the rest. Our fixed default parameters are a network size of 100 nodes, a node spacing of 20 meters between nodes, a network distance of 500 meters from shore, and a hover time of 2 seconds (which is 9.6 kbits of data transferred).

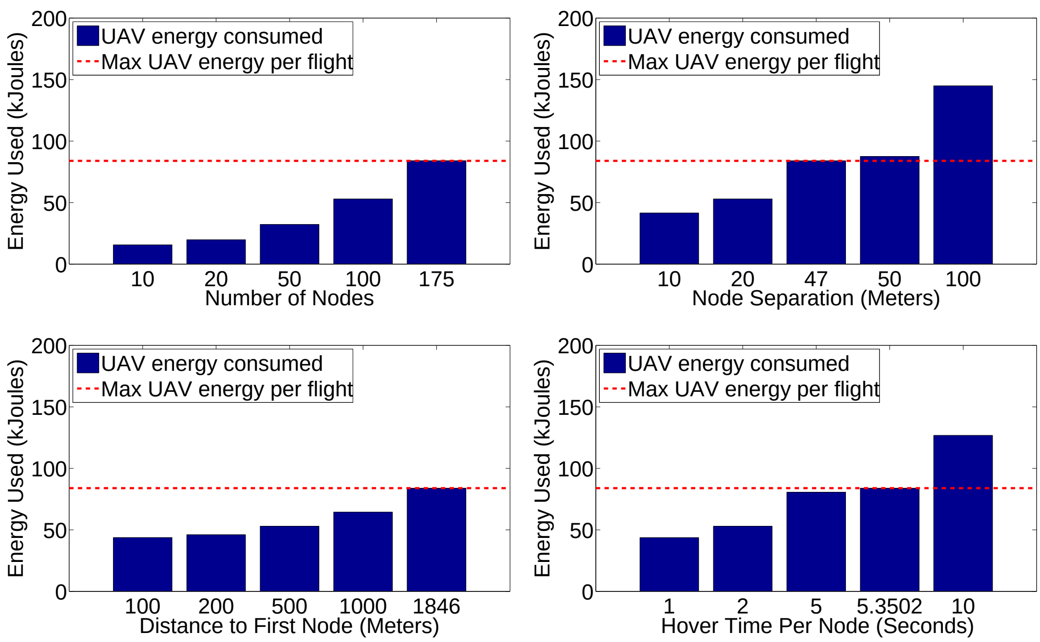

Figure 25: Total UAV energy while varying parameters: (a) network size, (b) node spacing, (c) distance to network, and (d) hover time

Figures 25(a) through 25(d) outline the results of skewing each variable. Each bar graph represents a single parameter change; the red dotted line in each represents the maximum energy of the UAV.

We first look at the number of nodes in the network with network sizes of 10, 20, 50, and 100 nodes as shown in Figure 25(a). At 175 nodes a single UAV does not have enough energy to collect all the data in 1 trip.

Figure 25(b) shows the results with node spacings of 10, 20, 50, and 100. With this configuration, a single UAV can handle up to 47 meter node spacing.

Next, Figure 25(c) moves the network further away from shore while fixing the remaining parameters. In order for the UAV to return home safely the maximum distance the network can be away from the UAV base station is 1846 meters.

Finally, we examine the hover time; Figure 25(d) varies the hover time from 1, 2, 5, and 10 seconds (or data transmission values per node of 4.8, 9.6, 24, and 48 kbits). The best a single UAV can achieve is 5.3 seconds of hover, which is equivalent to a data transfer of 25681 bits per node.

Overall, the key factors affecting the overall energy level are the separation between the nodes and the hover time the UAV spends at each node seen in the greater energy usage over the increased values.

These results do not consider communication; we can now examine the effective ranges of these values once we constrain the internode distance based on communication.

4.5 Overall Analysis and Policies

After examining communication and energy separately, we can now explore the complete system and the policies for systems combining underwater sensor networks and UAVs. The communication ranges and ensuring the UAV returns safely to the home base are the two key limiting factors that constrain the system topology and installation. Given these two constraints plus an area to cover, we define the number of nodes in the network, the possible distance of the network from shore, and the amount of time the UAV visits each node (which defines the amount of data it can collect).

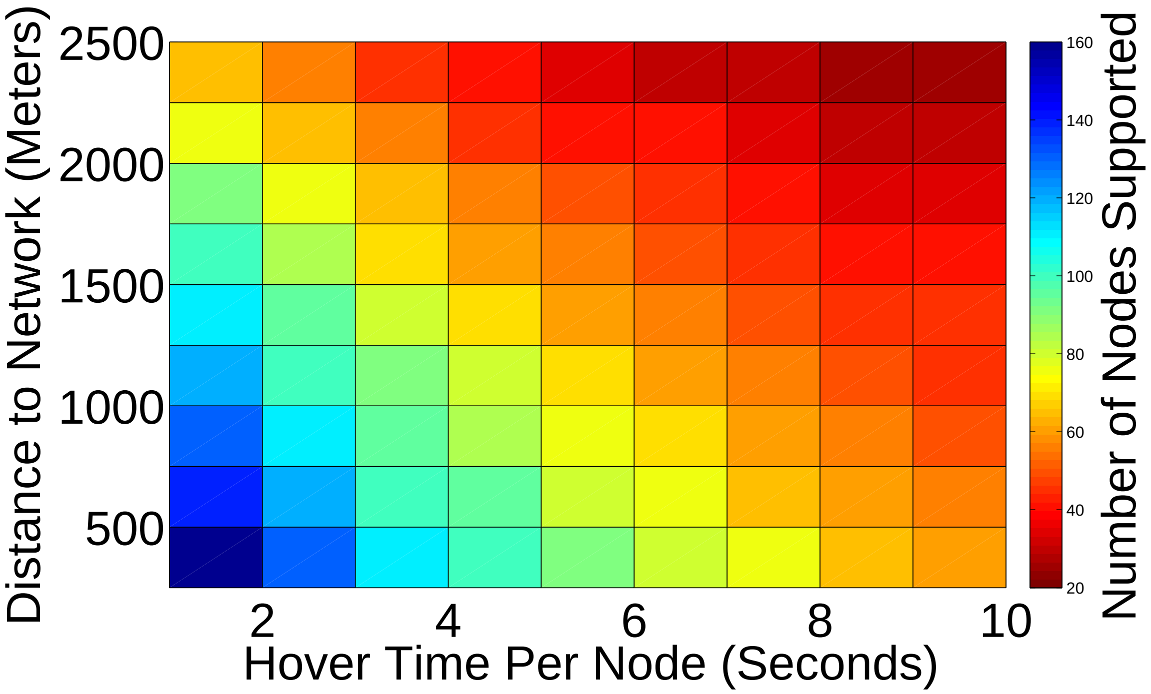

Figure 26: Network size a single UAV can support for 34 m node spacing.

We first assume that, in addition to collecting data with the UAV, we want the sensor network to communicate using radios, limiting the node spacing to a maximum of 34 meters. Figure 26 shows the results of this spacing as we limit the parameters to ensure the UAV has sufficient energy to return to base. Within this limitation, the maximum network size possible is 160 nodes; any more than that and the system no longer functions (either the UAV has insufficient energy, the network is at the shore, or the UAV cannot visit the nodes for even a second). Additionally, the distance from shore ranges from 250 to 2,500 meters and the UAV hovers 1 to 10 seconds (equal to 4,800 to 48,000 bits of data). For a network of only 20 nodes, this indicates that the closest node can be 2,500 meters from shore and the UAV can collect 48,000 bits of data from each node. However, for a network of 160 nodes, the closest node of the network needs to be within 500 meters of shore and the UAV can only collect 9,600 bits of data from each node.

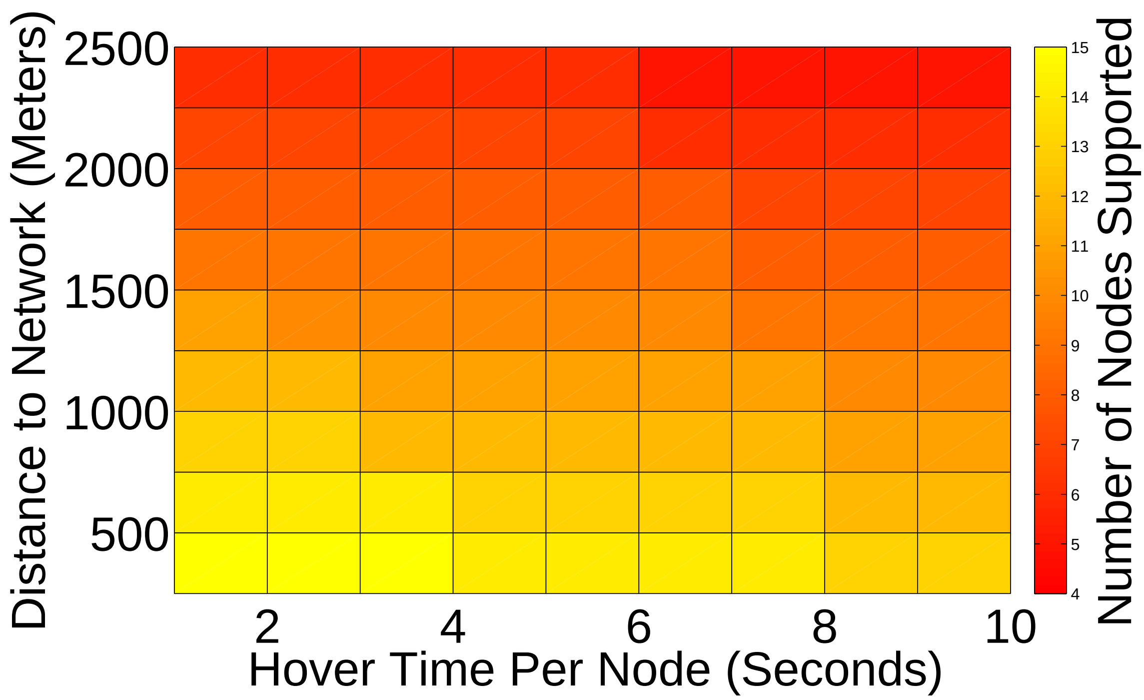

Figure 27: Network size a single UAV can support for 424 m node spacing.

If we relax the internode communication requirement, we can space the nodes at 424 meters apart as determined in Section 4.3.4. Figure 27 shows the results of this spacing for the requirement that the UAV has sufficient energy. Here the hover time and range from shore remain the same, but the number of nodes the network can contain reduces to a minimum of 4 and a maximum of 15. Placing the network 400 meters from shore and hovering only 1 second (collecting 4800 bits of data), the UAV can support a network of 15 nodes. With a minimum of 4 nodes, the network can exist 2500 meters from shore and the UAV can hover 10 seconds per location.

The above results assume the key policy factor is maximizing area covered while minimizing the likelihood of losing the UAV. We instead could define the key policy as the density of nodes we can place within a fixed area, which allows the communication spacing to vary between 34 and 424 meters. Figure 28 shows the possible network sizes in a 2km^2 area that a single UAV can handle in a single trip.

Surprisingly, for a network that is 2500 meters from shore where the UAV hovers 10 seconds per node, the UAV cannot support any network size and still return to shore. However, as long as the network is within 1250 meters from shore and the UAV hovers 1 second per node, the network can contain 400 nodes.

Overall, these figures outline the possible options for the network configuration and topology based on whether the key factor is the maximum spacing between nodes or a specific area. Without consideration of area the largest network size the system can support is 160 nodes at 34 meter spacing, the largest area for coverage while still allowing the UAV to provide communication connection 3392 m^2, and a sensor network covering a 2 km^2 area cannot exist at over 2500 meters from shore.

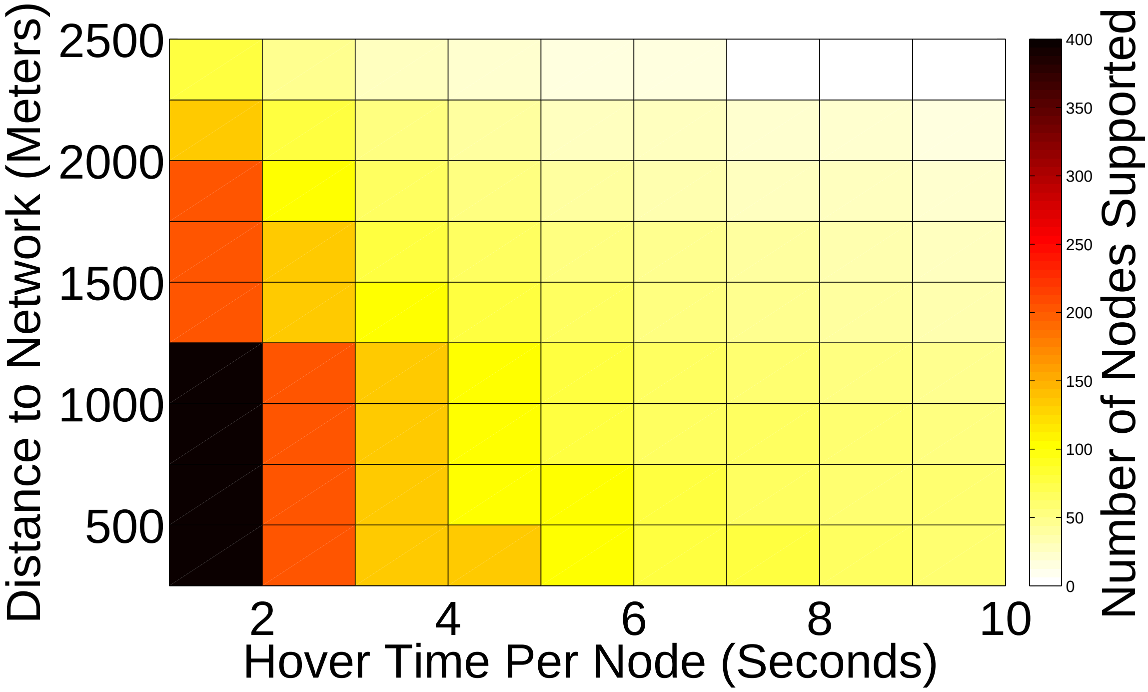

Figure 28: Network size a single UAV can support in a 2 km^2 area.

Another interesting comparison is between using an UAV to extract data out of the network and radio hopping the data out. Our system has a couple of benefits when compared to an underwater sensor network that uses radio hopping to get data out from the network. If both networks are using AquaNodes then our UAV data muling system enables the network to be further away from the base station and be more sparse. In a radio hopping scenario the network would need to have at least one AquaNode within radio communication range of the base station or a radio tower. The best case would be where a radio tower is at an elevation of 43 meters; in this case the closest node would need to be within 212 meters in order to communicate the data out. As seen in our previous results if a UAV is used then the network could be much further out than 212 meters. Another benefit to using an UAV would be that the AquaNodes would deplete energy at a much more even rate. This is because the AquaNodes communicate directly to the UAV. In the case where there is no UAV then the AquaNodes closest to shore would deplete their battery much faster than the rest because they need to relay more data. Using a UAV results in a longer network lifetime.

In summary we conducted three field experiments to characterize radio communication between an UAV data mule with an underwater sensor network, between two surfaced underwater nodes, and between two water nodes on land. We then developed a simulation environment based on experimentally measured energy values in MATLAB to model the energy usage of an UAV. Finally, we deduced that an UAV is indeed an effective data mule for certain underwater sensor network configurations.

Chapter 5: Conclusions and Future Work

The field of underwater wireless sensor networks is a blossoming field and has many important applications dealing with monitoring bodies of water. However, due to the underwater environment, one of the main challenges that the field faces is communication. In this thesis we worked to improve underwater sensor network communication. The remainder of this chapter will discuss the main conclusions from this thesis as well as potential future work.

5.1 Conclusions

This section will discuss the key conclusions from our research on improving underwater sensor network communication. We will discuss the AquaNode to AquaNode communication simulation environment and exploring UAV data muling in an AquaNode underwater sensor network.

5.1.1 Simulation environment. In this thesis we expanded the capabilities of an existing simulation environment created by Michael O'Rourke. The simulation environment is designed to model and simulate multi-hop node to node communication within an underwater sensor network that utilizes AquaNodes. We expanded the simulation environment by fixing problems, adding packet-loss to acoustic communication, and adding 3D topologies.

We found that the initial simulation environment had three main issues that needed to be address. After these issues were fixed we found a reordering of the communication algorithms; the algorithms that communicated more now show a higher energy consumption. The two centralized algorithms consumed so much energy that they are not viable options anymore. The decentralized radio centric algorithms are still more energy efficient than the decentralized acoustic algorithms.

We then added acoustic packet-loss into the simulation environment, verified that it functioned correctly, and found that acoustic packet-loss affects the communication algorithms that do more acoustic communication more than the ones that do not.

Finally we added 3D topologies into the simulation environment, verified that they were being generated correctly, and found that ordering of communication algorithms in terms of energy efficiency did not change.

5.1.2 UAV data muling. In addition to expanding a simulation environment we also explored using an UAV as a data mule in a underwater sensor network that utilizes AquaNodes. We performed three field experiments to characterize radio communication, modeled the energy consumption of the UAV with experimentally measured values, and created overall policies for using an UAV with an underwater sensor network that utilizes AquaNodes.

We found that node to node radio communication on the surface of the water had less range and a lower packet-success rate than on land. On the surface of the water the node to node radio communication had a range of 34 meters at 42% packet-success rate. This is less than what we see on land where we see a range of 45 meters at 70% packet-success rate. For node to UAV radio communication over water we saw that there was limited communication range when the UAV was flying at lower altitudes and we saw an increase in communication range when the UAV's altitude increased up until 43 meters. At an altitude of less than 5 meters we saw a rapid decline in communication success-rates as the horizontal distance between the node and the UAV increased; at 75 meters we saw less than 50% success-rate. At an altitude of 43 meters we saw a communication success rate of 75% when the UAV was horizontally 212 meters away.

In addition to performing the field experiments we also modeled the energy consumption of the UAV while data muling. The energy consumption of the UAV mainly depended on the energy required to travel to the network and traverse it, the energy required to hover nodes to collect data, and the energy required to use radio communication. We found that the key factors that affect the overall energy usage of the UAV are the separation between nodes and the amount of time that the UAV hovers above nodes.

After analyzing the communication and energy separately we then analyzed the complete system and developed usage policies for systems combining underwater sensor networks and UAVs. The limiting factors for these policies were the communication ranges and ensuring that the UAV has enough energy to make it home safely. These policies show how many nodes can be supported by a single UAV in one trip while skewing distance away from the network and the amount of time that UAV hovers at each node to collect data. We developed three sets of policies; the first two policies look to maximize area covered while ensuring UAV safely and the third examines how many nodes can be supported in a 2 km^2 area. We found that an UAV data mule is indeed a viable and effective option for extracting data out of an underwater sensor network for certain underwater sensor network configurations.

5.2 Future Work

There are a wide variety of options for future work for the AquaNode node to node communication simulation environment, exploring UAV data muling in an AquaNode underwater sensor network, and the combination of the two.

For the simulation environment packet-loss could be added to radio communication and a scheme for handling packet-loss could be implemented. Currently the simulation environment sends packets until they are successfully received or the environment times out. The packet-loss handling scheme could be set up to send a finite amount of messages. If the message is not successfully received within that amount of attempts the transmitting node could reroute the data through a different neighboring node. This would improve the robustness of the simulation environment. The rerouting idea would also enable the implementation of node failures within the network which is another possible option for future work. In order to do this the simulation environment would need to model multiple multi-hop communication iterations. Currently it only models a single data transfer across the network via radio communication. With multiple communication iterations we could then track network lifetime and determine when nodes have exhausted their battery life. The rerouting scheme would need to be robust enough to handle node failures and packet-loss issues.

With the addition of multiple communication iterations the simulation environment could also be combine the UAV data muling work. With that combination it would be possible to directly compare the two data extraction methods of multi-hopping the data out via radio and using an UAV to data mule the network.

In the simulation environment the UAV work could also be expanded to have different visiting schedules and different network traversal algorithms. Future work could also be done on physically prototyping a system that models the entirety of parts of the simulation environment.

REFERENCES

[1] Ascending technologies.

[2] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review*, 1(2):3–8, 2004.

[3] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.

[4] C. Detweiler, M. Doniec, I. Vasilescu, and D. Rus. Autonomous depth adjustment for underwater sensor networks: Design and applications. *IEEE Transactions on Mechatronics*, 17(1):16–24, 2012.

[5] Carrick Detweiler, Marek Doniec, Iuliu Vasilescu, Elizabeth Basha, and Daniela Rus. Autonomous depth adjustment for underwater sensor networks. In *Proc. of the Intl Workshop on UnderWater Networks (WUWNet)*, pages 12:1–12:4, 2010.

[6] Carrick James Detweiler. *Decentralized sensor placement and mobile localization on an underwater sensor network with depth adjustment capabilities*. PhD thesis, Massachusetts Institute of Technology, 2010.

[7] M. Doniec. *Autonomous underwater data muling using wireless optical communi-*

cation and agile AUV control. PhD thesis, Massachusetts Institute of Technology, 2013.

[8] John Heidemann, Milica Stojanovic, and Michele Zorzi. Underwater sensor networks: applications, advances and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):158–175, 2012.

[9] John Heidemann, Wei Ye, Jack Wills, Affan Syed, and Yuan Li. Research challenges and applications for underwater sensor networking. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 228–235. IEEE, 2006.

[10] J. T Isaacs, S. Venkateswaran, J. Hespanha, U. Madhow, J. Burman, and T. Pham. Multiple event localization in a sparse acoustic sensor network using uavs as data mules. In *Globecom Workshops (GC Wkshps)*, pages 1562–1567. IEEE, 2012.

[11] D. J. Klein, J. Schweikl, J. T. Isaacs, and J. P. Hespanha. On uav routing protocols for sparse sensor data exfiltration. In *American Control Conference (ACC)*, pages 6494–6500. IEEE, 2010.

[12] Liu Lanbo, Zhou Shengli, and Cui Jun-Hong. Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing*, 8(8):977–994, 2008.

[13] K. Li, C.-C. Shen, and G. Chen. Energy-constrained bi-objective data muling in

underwater wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS)*, pages 332–341. IEEE, 2010.

[14] Michael O'Rourke, Elizabeth Basha, and Carrick Detweiler. Multi-modal communications in underwater sensor networks using depth adjustment. In *Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems*, page 31. ACM, 2012.

[15] Michael J. ORourke. Simulating underwater sensor networks and routing algorithms in matlab. Master's thesis, University of the Pacific, 2012.

[16] Ivan Stojmenovic. Position-based routing in ad hoc networks. *Communications Magazine, IEEE*, 40(7):128–134, 2002.

[17] R. Sugihara and R. K. Gupta. Improving the data delivery latency in sensor networks with controlled mobility. In *Distributed Computing in Sensor Systems*, volume 5067 of *Lecture Notes in Computer Science*, pages 386–399. Springer, 2008.

[18] R. Sugihara and R. K. Gupta. Optimizing energy-latency trade-off in sensor networks with controlled mobility. In *International Conference on Computer Communications (INFOCOM)*, pages 2566–2570. IEEE, 2009.

[19] R. Sugihara and R. K. Gupta. Optimal speed control of mobile node for data collection in sensor networks. In *Mobile Computing*, volume 9, pages 127–139. IEEE, 2010.

[20] R. Sugihara and R. K. Gupta. Speed control and scheduling of data mules in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(1):4, 2010.

[21] Guang Tan, M. Bertier, and A.-M. Kermarrec. Visibility-graph-based shortest-path geographic routing in sensor networks. In *INFOCOM 2009, IEEE*, page 17191727, April 2009.

[22] S. K. Teh, L. Mejias, P. Corke, and W. Hu. Experiments in integrating autonomous uninhabited aerial vehicles (uavs) and wireless sensor networks. In *Australasian Conference on Robotics and Automation (ACRA)*. The Australian Robotics and Automation Association Inc., 2008.

[23] Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus. AquaNodes: an underwater sensor network. In *Proc. of 2nd WUWNet*, pages 85–88, Montreal, Quebec, Canada, 2007. ACM.

[24] Iuliu Vasilescu, Keith Kotay, Daniela Rus, Matthew Dunbabin, and Peter Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165. ACM, 2005.