Denial-of-Service Attacks and Countermeasures in Underwater Acoustic Networks

Michael E. Zuba, Ph.D.
University of Connecticut, 2014

ABSTRACT

Advances in underwater acoustic telemetry have paved the way for emerging technologies, such as underwater networked systems. Now underwater devices and sensors can interact with each other by forming wireless underwater acoustic networks. These devices are networked together through the use of acoustic communication. Although wireless networking is not a new idea, applying these principles to the underwater domain has become a novel and challenging problem. There are many reasons for these challenges but the two most important ones are the inherent properties of using acoustic signals for underwater communication and the development of large-scale underwater networked systems.

These new grand challenges have become the shaping factors behind the design of new system architectures, protocols and algorithms. Entire network protocol suites and testbed systems have been designed and are currently being used in the real-world for experimentation and evaluation purposes. However, the unique deployment environments of such systems allow for easier exploitation and as is usually the case, the security of said systems has not been considered. Further, the unique challenges posed by the use of underwater acoustics make existing security solutions unusable.
This dissertation work focuses on denial-of-service attacks and countermeasures for underwater acoustic networks and covers three major research thrusts, namely: 1) physical layer vulnerability; 2) network layer vulnerability; and 3) countermeasures.

We first investigate the vulnerability of the physical layer of underwater networks through use of jamming attacks. Jamming attacks inject malicious signals into the communication channel to corrupt or block legitimate communication from occurring. We define two types of attackers with four different attack methods. These attacks are evaluated on three commonly used underwater acoustic modems in a real-world experimental testbed in Mansfield Hollow Lake, Mansfield, Connecticut and in a lab testbed.

Secondly, we investigate the vulnerabilities of the network layer of underwater networks through use of spoofing or cheating attacks. We study commonly used pressure routing protocols and design a spoofing attack to observe the effects on network performance. A spoofing attack is where a node in the network sends out fake location information, claiming to be somewhere other than its actual location with intent to disrupt network operations.

Thirdly, we make use of the knowledge gained by our first two works to design a resilient pressure routing scheme for underwater acoustic networks, known as RPR. This protocol aims to maintain routing services in the presence of malicious adversaries. It utilizes cryptographic mechanisms, implicit acknowledgments (I-ACKS), geographic constraints and randomization to achieve its resilient packet delivery.
Denial-of-Service Attacks and Countermeasures in Underwater Acoustic Networks

Michael E. Zuba

B.S., University of Connecticut, 2010

M.S., University of Connecticut, 2013

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
at the
University of Connecticut
2014
APPROVAL PAGE

Doctor of Philosophy Dissertation

Denial-of-Service Attacks and Countermeasures in Underwater Acoustic Networks

Presented by

Michael E. Zuba, B.S., M.S.

Co-Major Advisor

Jun-Hong Cui

Co-Major Advisor

Zhijie Shi

Associate Advisor

Song Han

University of Connecticut

2014
ACKNOWLEDGEMENTS

This journey through the Ph.D. program has been a roller coaster ride of emotions and experiences. It has allowed me to further myself in every aspect of my life from professionally to personally. I have learned something new everyday and that in itself is my most valued accomplishment. These past four years could not have been possible without the support of many people along the way.

First, I would like to thank my academic advisor, Dr. Jun-Hong Cui, who has provided a nurturing environment with an abundance of opportunities to continue my life long effort of learning. Dr. Cui has provided invaluable mentoring and guidance and I am greatly appreciative of every second she has invested into my learning and my personal and professional development. I also want to thank my co-major advisor Dr. Zhijie Shi for his guidance on research efforts. Additionally, I express my gratitude to my associate advisors: Dr. Song Han, Dr. Reda Ammar and Dr. Shengli Zhou for their input on this dissertation and their continual help and teaching throughout the last four years.

I have been extremely lucky to have been part of a large and diverse research group and wish to express my thanks to my fellow labmates: Dr. Zheng Peng, Dr. Jun Liu, Haining Mo, Li Wei, Xiaoyan Lu, Zigeng Wang and many others. They have provided a comfortable learning environment, technical support and great conversations over the last few years. My life as a graduate student was also made much easier by the support of the CSE Department and I wish to extend my thanks to Debra Mielzareck, Howard Ellis, Rebecca Randazzo, Karen Slater and Joy Billion for their patience and help on my almost daily visits.
I’ve been honored to take part in various external research activities and wish to express my gratitude to those opportunities and their hosts. I want to thank the Office of Naval Research (ONR) and the National Science Foundation (NSF) for their financial support in the NREIP and EAPSI programs. Thank you to Dr. Zaihan Jiang and Dr. Jeff Schindall from the U.S. Naval Research Lab (NRL) for hosting me during the NREIP program and providing valuable training and field test experience. Thank you to Dr. T.C. Yang from National Sun Yat-sen University in Taiwan for hosting me during the EAPSI program and providing a once in a lifetime opportunity.

I extend my personal gratitude to Alberto De La Rosa Algarín who has been my partner in crime here at UConn during the Ph.D. program. He has provided me with valuable advice and many enjoyable times these past years. His friendship has provided me with a better appreciation of life and helped me to always remember to take a moment to laugh, even in the most inappropriate times.

My biggest source of everyday support comes from Tierney Meehan who has been my co-pilot throughout these past years. She has kept me grounded and provided me with heartfelt love and support, even when forced to review my overly wordy writing style. I cannot thank her enough for all of her help and contributions to my learning and growth as a person.

I would not be here if it were not for my family. My siblings have provided me with everlasting inspiration and my mother and father have raised me and supported me in all my dreams and pursuits. I cannot thank my parents enough for their sacrifices, support and love. They are truly the best parents in the world. Thank you for everything. I would also like to thank my grandmother Dolores Marie Smith for her continual support and encouragement.

Finally, I wish to dedicate this dissertation to my grandfather Frederick Eells Smith, my grandfather Edward Frank Zuba and my grandmother Irene Zuba who are forever watching over me.
# TABLE OF CONTENTS

Chapter 1: Overview

1. Introduction and Motivation ................................................................. 1
   1.1.1 Underwater Acoustic Networks ............................................. 3
   1.1.2 Underwater Communication .................................................. 4
   1.1.3 Applications ................................................................. 6

1.2 Contributions of This Dissertation ................................................... 7

1.3 Dissertation Roadmap ................................................................. 8

Chapter 2: Background

2. Network Security ...................................................................................... 9
   2.1 Unique Factors of UANs .............................................................. 11
   2.1.2 Denial-of-Service ................................................................ 13

2.2 Physical Layer Attacks ........................................................................ 13
   2.2.1 Eavesdropping ..................................................................... 14
   2.2.2 Jamming .............................................................................. 14

2.3 Network Layer Attacks ........................................................................ 14
   2.3.1 Spoofing ............................................................................. 15
   2.3.2 Replaying ............................................................................ 15
   2.3.3 Wormholes ........................................................................ 15

2.4 Security Requirements .......................................................................... 17
   2.4.1 Confidentiality .................................................................... 17
## 2.4 Authentication

- 2.4.2 Authentication ........................................ 17
- 2.4.3 Integrity ................................................ 18
- 2.4.4 Freshness ............................................. 18
- 2.4.5 Availability .......................................... 18

## 2.5 State-of-the-Art in UAN Security

- 2.5.1 Physical Layer ...................................... 19
- 2.5.2 Network Layer ....................................... 20

### Chapter 3: Launching Jamming Attacks

- 3.1 Attack Models ........................................... 24
- 3.2 Network Metrics ........................................ 25
- 3.3 Experimental Study - Signal Jammer ................. 26
  - 3.3.1 Benthos Acoustic Modems ...................... 27
  - 3.3.2 AquaSeNT Acoustic Modems .................. 31
  - 3.3.3 Linkquest Acoustic Modems .................. 33
  - 3.3.4 Discussion ........................................ 33
- 3.4 Experimental Study - Deceptive Jammer ............ 35
- 3.5 Theoretical Study - Jammer Deployment ............ 36
- 3.6 Summary ................................................ 40

### Chapter 4: Spoofing Geographic Routing

- 4.1 Related Work ........................................... 43
- 4.2 Vulnerability Analysis ................................ 44
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example UAN Architecture</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Real Gateway Buoy</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Example Wormhole Attack</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Example UAN Specific Wormhole Attack</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Jammer Hardware</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Field Test Deployment</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>Throughput of Jamming Attacks</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>General Signal Diagram</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>OFDM Field Test Deployment</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>General Signal Diagram of OFDM Transmission</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>Preamble Time Constraints</td>
<td>34</td>
</tr>
<tr>
<td>12</td>
<td>Distance Requirements for Constant Jamming Attacks</td>
<td>39</td>
</tr>
<tr>
<td>13</td>
<td>Visualization of a Typical DBR Scenario</td>
<td>45</td>
</tr>
<tr>
<td>14</td>
<td>Visualization of Preliminary Attack Model</td>
<td>47</td>
</tr>
<tr>
<td>15</td>
<td>Comparison of Holding Times vs. Propagation Time</td>
<td>49</td>
</tr>
<tr>
<td>16</td>
<td>Intersection of Holding Times vs. Propagation Time</td>
<td>49</td>
</tr>
<tr>
<td>17</td>
<td>Simulation Results for 5 Random Networks</td>
<td>52</td>
</tr>
<tr>
<td>18</td>
<td>Simulation Results for Various Spoofed Depths</td>
<td>52</td>
</tr>
<tr>
<td>Page</td>
<td>Section Title</td>
<td>Page</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>19</td>
<td>Improved Attack Model Results for Previous 5 Networks</td>
<td>55</td>
</tr>
<tr>
<td>20</td>
<td>Search Positions and Data Transmission Route</td>
<td>55</td>
</tr>
<tr>
<td>21</td>
<td>One Example of Node Mobility over 6000 seconds</td>
<td>58</td>
</tr>
<tr>
<td>22</td>
<td>Performance of Improved Attack on a Mobile Network</td>
<td>59</td>
</tr>
<tr>
<td>23</td>
<td>RPR Algorithm Flow</td>
<td>63</td>
</tr>
<tr>
<td>24</td>
<td>Packet Delivery Ratio</td>
<td>72</td>
</tr>
<tr>
<td>25</td>
<td>Average End-to-End Packet Delay</td>
<td>73</td>
</tr>
<tr>
<td>26</td>
<td>Average Network Transmissions</td>
<td>73</td>
</tr>
<tr>
<td>27</td>
<td>Branching Benefits</td>
<td>74</td>
</tr>
<tr>
<td>28</td>
<td>Attack Performance</td>
<td>75</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network Performance</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>Start Time of Jammer vs. Distance from Receiver</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>Deceptive Jammer Attack 1 (JSR=5s)</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>Deceptive Jammer Attack 2 (JSR=3s)</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>Deceptive Jammer Attack 3 (JSR=1s)</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>Improved Attack Model Performance</td>
<td>57</td>
</tr>
<tr>
<td>7</td>
<td>Improved Attack Model Performance on a Mobile Network</td>
<td>59</td>
</tr>
</tbody>
</table>
Chapter 1

Overview

1.1 Introduction and Motivation

Oceans on Earth are estimated to cover roughly 71% of the planet’s surface. With such enormity it is easy to understand that oceans play an integral role in the development and sustainability of life. Further, more than 95% of the underwater world remains elusive and unexplored. The water world is becoming a major focal point in our current society as humans become more concerned about anthropogenic disturbances, such as pollution and ocean acidification, monitoring of severe weather for improved storm warning systems and port detection of smugglers and protection of disputed borders [1]. In the United States alone, one of every six jobs is marine-related and over one-third of the U.S. Gross National Product originates in coastal areas [2].

In recent years, efforts have been focused on improving underwater acoustic modem technology to extend our reach and presence in the water world. Acoustic communication is the primary method for underwater communication because radio waves suffer from high attenuation
underwater and optical waves are prone to heavy scattering [3–12]. As communication technology grew, researchers began to bring the wireless networking paradigm to the underwater domain through use of underwater sensor networks (UWSNs), also known as underwater acoustic networks (UANs) [13–28]. UANs have gained much attention from academic, industrial and military researchers due to the potential benefits of these systems. The initial research on UANs has shown that an abundance of new applications can become feasible, such as undersea exploration, disaster prevention, mine reconnaissance, coastal surveillance and environmental monitoring [29–33]. However, the potential benefits of UANs bring distinct challenges that are inherent to the use of underwater acoustics. Characteristics of underwater acoustic communication introduce additional design constraints and complexity at every layer of the network protocol stack [17, 25, 27, 34–39]. Common problematic features are that of low communication bandwidth, long and variable propagation delays, high levels of underwater noise generated from ships and sea life, higher error probabilities in the channel, sensor node mobility and energy constraints [6, 9, 40–42]. All of these features must be considered when developing a network protocol or system.

In this dissertation work, we address the important problem of security in UANs, as the unique features of the underwater environment do not allow for direct application of existing security mechanisms. As UANs are an emerging technology, now is the time to call attention to the potential threats for underwater networks. Malicious adversaries could disrupt critical services in any network. These attacks can obtain sensitive information, block all communication from occurring, disrupting routing services and even inject false information into the network. With the preliminary development of underwater systems and protocols, information on security threats will be greatly beneficial as it can be used to help mold the secure design of these systems and protocols. Often, inadequate designs or delayed attention to security measures leads to inefficiencies in network performance. The goal of this dissertation work is to call
early attention to security concerns such that all future protocols can be designed with security features in mind.

1.1.1 Underwater Acoustic Networks

UANs are composed of various sensors, vehicles and devices which cover large spatial areas of aquatic environments. Each individual node in the network can perform its own unique tasks, such as monitoring a chemical spill or collecting data from other nodes in the network. Each node is networked together using an underwater acoustic modem. The standard hardware on a node consists of battery packs to power internal devices, a power distribution board to send specific power to each unique device, a microcontroller computer to run the networking software, as well as other software related to any attached sensors, and an acoustic modem for communicating with other nodes. Nodes can also be fitted with individual sensors (temperature, chemical, etc.) for any tasks that might be needed. These internal components are packed into a waterproof container on the deployment platform. The deployment platform can come in many forms: 1) a surface or gateway buoy which sits on top of the water making use of RF modems and GPS; 2) an a-frame structure anchored to the sea floor; or 3) a free floating sensor that passively moves with the ocean current. Many experimental and prototype testbed systems have been developed and follow the above designs and use of hardware and software equipment [22–25, 43–47].

A common UAN deployment architecture can be seen in Figure 1 and a real gateway buoy system can be seen in Figure 2. In the deployment architecture from Figure 1 we can see that the network is actually partitioned into two unique networks. The first one is the underwater acoustic network. These nodes are all underwater and are performing some specific tasks. The information collected at each node is then relayed through other nodes until it reaches the surface or gateway node. The gateway nodes act as aggregators, collecting all the data from
below the surface. The gateway nodes are equipped with RF modems and GPS for location information and make up the second network, the surface radio network. These nodes can send the collected information to a command center in the form of a ship or station on shore using the RF modems. This type of hybrid architecture is one of the most common styles [48–53].

1.1.2 Underwater Communication

Acoustic communication is the primary method used for underwater communication. This form of communication involves the propagation of sound which is created by an acoustic
transducer. The reason acoustic communication has become the primary method is due to the performance issues of other communication methods [3, 54–56]. For example, electromagnetic waves suffer from high attenuation underwater and to combat this problem, a transmitter must be equipped with a large antenna and power source. However, this is not feasible in the underwater environment. The other alternative is optical waves, which are used in some underwater applications, but suffer from heavy scattering caused by objects in the environment, such as sea life or vegetation. The performance impact by this scattering phenomena makes this method unsuitable for reliable long distance communication purposes. These challenges by other communication methods have lead acoustics to be the the primary method of underwater communication.

Underwater acoustic communication is not without its own obstacles. Many inherent challenges plaque the underwater acoustic channel [3, 4, 6, 9, 40–42]. The propagation speed of acoustic signals in water is about $1.5 \times 10^3$ m/s, which is five orders of magnitude lower than the radio propagation speed in air, which is $3 \times 10^8$ m/s. Additionally, the delay caused by propagation speed can be highly dynamic since some nodes can be actively or passively moving. Noise generated by ships, sea life and other underwater users, such as sonar, also impacts the acoustic channel. The direct result of the presence of these other players is interference in the channel and potential loss of communication between nodes. Further, multipath and path loss are other challenges that are still being investigated. Multipath is the process in which acoustic waves can reach a certain point through multiple paths. This is a result of various factors such as wave reflections from the sea surface or sea floor. Path loss is the loss of energy along the propagation path, which is caused by geometric spreading, absorption and scattering. The direct result of this is a decrease in signal strength. All of these above factors contribute to the high error probability nature of the acoustic channel. A final key challenge is the limited bandwidth available for transmissions. This is dependent upon the operating frequency and
transmission range and therefore users can not expect to be able to send out large amounts of data.

1.1.3 Applications

The potential applications for UANs is growing as the technology matures and real systems become better developed. The common applications discussed in literature classify UAN applications into two broad categories: 1) long-term and non-time-critical aquatic monitoring; and 2) short-term time-critical aquatic exploration [13–18]. As the categories suggest, applications are generally divided by on-demand or permanent placement of nodes, volume and type of data being transmitted or collected and time constraints. Specific aquatic monitoring applications mentioned in literature include oil/gas monitoring, pollution detection, oceanography and marine biology studies, seismic predictions and deep-sea archeology. In the other category, aquatic exploration, literature mentions submarine detection, anti-submarine military missions, harbor and port anti-drug smuggling protection, disaster recovery, natural resource discovery and ship/treasure discovery and recovery.

The classification of the application helps to determine the design of the network protocols being used in the system [14]. In long-term monitoring applications, nodes are deployed over large spatial areas for continuous monitoring. Energy conservation is a key issue for this type of application and needs to be the basis for each protocol design used in such an application. This issue is elongated by the fact that in long-term monitoring applications, the nodes will most likely have to perform a localization process in order to know their exact positions. Data from monitored events is often required to be tagged with geographic information from where it was recorded or else it might not be useful. Further, in monitoring applications, efficient multi-access techniques and routing techniques must be used.
In short-term exploration applications, energy conservation is no longer the main concern as reliability and multi-access techniques are more important. Additionally, real-time data transfer becomes an issue. Exploration applications can require varying transmission rates as the information being transmitted can range from small alert packets to long packet trains that contain captured images.

1.2 Contributions of This Dissertation

In this dissertation work, we investigate security concerns in UANs, namely, denial-of-service security attacks and countermeasures. UANs have potential to be used in many sensitive applications and therefore it is necessary to understand the vulnerabilities of these networks to produce effective countermeasures. This dissertation covers three major pieces of work.

First, we investigate the vulnerabilities of the physical layer to denial-of-service jamming attacks. Jamming attacks are launched on UANs in an experimental setting by using a developed jamming system and signal. Attacks are tested against three commercially available acoustic modems. Additionally, weaknesses in the signal structure of each acoustic modem are studied through observation of how the network responds to jamming attacks. To the best of our knowledge, this is the first work in literature to study jamming effects in real-world settings on commercially available acoustic modems.

Second, we perform a vulnerability analysis of the network layer in UANs with specific focus on geographic based routing protocols. An attack model is developed to mimic real protocol behavior and demonstrates the impact of location based spoofing attacks. The attack model is improved by a proposed placement algorithm that seeks to discover bottlenecks in the network topologies. Insecure heuristics used in geographic based routing protocol designs are identified. To the best of our knowledge, this is the first work in literature to study spoofing attacks on underwater networks and underwater geographic routing.
Last, to address the concerns of denial-of-service attacks on both the physical and network layers of UANs, we propose a Resilient Pressure Routing scheme, known as RPR. This protocol aims to maintain routing services in the presence of malicious adversaries. RPR utilizes cryptographic mechanisms, implicit acknowledgments (I-ACKS), geographic constraints and randomization to achieve its resilient packet delivery.

1.3 Dissertation Roadmap

The remainder of this dissertation is organized as follows. In Chapter 2, we present background on security concerns, vulnerabilities on the physical and network layer of the protocol stack and existing work. In Chapter 3, we present our work on jamming attacks on the physical layer. Attack models are classified and various hardware devices are attacked in both lab and real-world experiments. The critical sections of the signal structure of different acoustic modems are shown. In Chapter 4, we describe our work on spoofing attacks on the network layer. We present our attack model and show the impact on network performance. In Chapter 5, we present our resilient pressure routing scheme known as RPR. Using information from our first two peices of work, this protocol is designed to survive attacks from malicious adversaries. Finally, we conclude this dissertation and discuss future work in Chapter 6.
Chapter 2

Background

2.1 Network Security

Underwater acoustic networks (UANs) provide a promising solution to safely and efficiently explore and monitor underwater environments. Much progress in the development of real systems, communication techniques, networking protocols and protocol suites has been achieved. However, vulnerability studies of these networks have been limited [57–65]. Most practical applications of UANs in the commercial, scientific and military domains can be classified as sensitive. Outsiders should not be allowed access to such information, as a malicious adversary may wish to tamper with UANs by attaching unauthorized nodes to capture data or disrupt a subset or potentially all functions of the network.

These attacks can come in the form of an outsider or insider attack. The form of the attacker determines which type of attack should be used. An outsider attack is one in which the attacker is not part of the original network and is trying to attach itself to the network or cause external disruptions. For example, a network could be deployed in the Atlantic Ocean to monitor the
boundaries of international waters. A malicious adversary could place their own network node into the water or pilot an autonomous underwater vehicle (AUV) into the network area to perform a security attack that blocks communication from occurring in the network, allowing an intruder to pass through the network. An insider attack is where a malicious adversary tampers with a legitimate network node to gain control over it and then perform more discrete attacks. In this case, the legitimate network node is now controlled by an attacker and the network will have a hard time distinguishing the presence of an intruder. The specific attacker could be trying to discover and collect sensitive data, manipulate the functions of the network or disrupt network services.

The largest contributor to network vulnerabilities is the use of wireless communication to link devices and nodes together. Literature on wireless communication is frequented with the knowledge that wireless communication itself has many insecure features [66]. The open nature of the communication channel allows outsiders to passively listen to any communication that is occurring, known as eavesdropping, inject falsified information and replay or retransmit overheard packets to degrade network performance. Another contributor commonly observed in literature is that malicious adversaries can easily tamper with devices [66, 67]. Outsiders can capture deployed devices and physically overwrite memory to gain access to the system. Tamper resistant devices should not be a major assumption when developing network protocols. This argument is further justified as making a device tamper resistant adds extra cost per-unit [66] and violates the benefits behind sensor based networks, which is that they are intended to be cheap. In short, tamper resistance is a viable option but not a general solution to the problem of communication and network security. Network protocols should be designed without assumptions of “one size fits all” solutions and must consider the unique characteristics of the wireless communication channel.
2.1.1 Unique Factors of UANs

Network security has been well studied in terrestrial networks and improved solutions are continually being developed. However, the nature of the underwater acoustic communication channel makes existing terrestrial defenses unable to be directly applied to UANs [64]. We can categorize five unique factors of UANs that will influence the design of network protocols. These factors are high bit error rates, large and variable propagation delays, narrow bandwidth, computational and energy constraints and a lack of accurate full-dimensional location information [64]. Giving attention to these factors will result in individualized and tailored solutions.

**High bit error rates** are a result of channel effects often caused by multipath and fading. Further, environmental noise such as ship activity, sea life, currents and water turbulence can also attribute to these higher error rates. Said factors play an influential role in the link quality between a sender and receiver, causing packets to be potentially lost or corrupted during transmission. In extreme scenarios, connectivity between nodes can be completely impaired for hours. The problem with high bit error rates is that they are not easily predicted and might be sporadic or frequent because of the dynamic nature of the environment. Underwater channel conditions are rarely identical, even over short time periods. This influences the design of security mechanisms as the network can not always assume perfect reliability or rely on the use of security/control/data packets. Similarly, authentication and verification schemes may also be affected.

**Large and variable propagation delays** are caused by the speed of sound in water. The propagation speed of acoustic signals in water is much slower than electromagnetic waves in the air. This increases the amount of time it takes a signal to reach its destination which in turn means that a malicious adversary has more time to block or manipulate the signal. Further, it also means the network has less time to respond or combat certain security threats. Nodes in the network can also be affected by ocean currents making the propagation delays
highly variable. Empirical observations propose that underwater nodes can move at a speed of $10^{-3}$ to $10^3$ cm$^2$/s in the vertical direction and from $10^{-3}$ to $10^5$ cm$^2$/s in the horizontal direction [68]. This factor will impact any potential security defense that might want to make use of propagation delay knowledge, such as Time of Arrival (TOA) or Time Difference of Arrival (TDOA) schemes.

*Narrow Bandwidth* is a large concern in underwater acoustic channels and data rates are inherently limited. As the transmission range increases, the data rate will decrease which limits the amount of information that can be sent and increases packet size overhead. The design of security mechanisms will have to consider this feature as oftentimes data packets are padded with security information which will eat up the useable data rate.

*Power Constraints* are observed in terrestrial networks but are much more rigid in the case of UANs because of the inaccessible nature of deployments. Further, the use of acoustic communication exacerbates this problem because the generation of acoustic signals is a “power hungry” operation. Energy consumption of acoustic transmissions is higher than radio transmissions. In combination with harsh deployment conditions, UANs lifetime power is already stressed. Secure protocol schemes will want to be mindful of these constraints in order to avoid unnecessary and costly transmissions.

*Lack of accurate full-dimensional location information* is the last major influential factor of UANs. GPS does not work underwater and therefore every node, except surface nodes which sit on the water surface, can not guarantee reliable positioning information. The only current exception to this statement is depth information, which can be obtained through an attached sensor. This negatively impacts security designs as knowledge of the *exact* location of nodes can be used to create node verifiers and location checking for trust management schemes. While localization in UANs is well studied and maturing rapidly, the availability of frequent
and accurate full-dimensional positioning information remains a daunting task. Additionally, this issue is elongated by the potential for node mobility.

2.1.2 Denial-of-Service

A malicious adversary has a multitude of techniques and avenues to pursue when attacking a wireless network. The most menacing is that of a denial-of-service (DoS) attack. A DoS attack occurs when an adversary seeks to make a service or services provided by the network unavailable. For example, a gateway buoy on the water surface provides the service of aggregating data from various network nodes and then sends this data to the base station on shore. A DoS attacker can target this gateway buoy and saturate it with its own communication requests. In this manner, the gateway buoy is not able to respond to legitimate requests or responds noticeably slower, depending upon the saturation rate of the attacker. The attacker may not wish to send any real data but simply keeps requesting an opportunity to send data to the gateway buoy. The end result is a denial-of-service for data collection. This is one specific example of how DoS attacks function and how various attacks are designed around the core idea of making specific services unavailable to legitimate users. DoS attacks are destructive because often the attack can be simple and mimics the behavior of a legitimate node, making it harder for the network to detect malicious intruders.

2.2 Physical Layer Attacks

The physical layer of the network is in charge of the communication between nodes in the network. A malicious adversary can overhear transmissions between nodes in the network due to the open nature of the underwater wireless channel. With this knowledge, the adversary can perform several attacks such as eavesdropping and jamming.
2.2.1 Eavesdropping

Eavesdropping or network sniffing is a passive attack that consists of listening in on transmissions between at least two nodes and capturing information from these transmissions. If a node is sniffing the network it could easily read the data being transmitted and search for sensitive information. Since the underwater acoustic link is open to any node within the transmission range, an attacker can observe communication occurring in the network. Consequently, it is difficult to detect if eavesdropping is actually occurring since the attacker does not need to send out any signals itself.

2.2.2 Jamming

A jamming attack is a common DoS attack where the adversary will inject malicious signals into the channel to create interference and block legitimate network traffic. In many cases, nodes in the network receiving transmissions cannot distinguish authorized signals from jamming signals. Malicious attackers can constantly send data packets that appear legitimate to a node with the intent to jam the channel. This type of attack is known to cause the most disruption in any wireless network. We elaborate more on jamming attacks in Chapter 3 where we introduce various jamming models and the effects of jamming attacks on UANs.

2.3 Network Layer Attacks

The network layer of the network is in charge of routing data from point A to point B. Various mechanisms for this routing exists, such as static routing and dynamic routing. Static routing is where routes are predetermined and do not change, while dynamic routing is where routes are determined using specific metrics and can be adjusted if the network experiences node failure or link quality loss.
2.3.1 Spoofing

Spoofing attacks can occur when an adversary places a malicious node in the network or when a legitimate node is compromised by an attacker. This node then masquerades as a legitimate node and can access sensitive information or change network dynamics such as routing information. One such attack is where a node claims to be in a certain location, even though it is physically in another. False location claims are often used to manipulate routing paths in the network. This node can also flood the network with its own transmissions and degrade the performance of the network. In short, spoofing attacks are used to falsify information, such as identity, location or data. As one of the contributions of this dissertation, we explore the effects of spoofing attacks on UANs with the design of a location based spoofing attack in Chapter 4.

2.3.2 Replaying

Replaying attacks are sometimes associated with jamming attacks. A replay attack is when a malicious attacker listens to what a legitimate node is transmitting, records this transmission and then replays it to the network. As an example, node A wants to prove its identity to another node, node B. Node B requests that node A send its password as proof of identity. Node A sends this password, after some security transformation, to node B. If an attacker, node C, is in the transmission area eavesdropping, it can record this password from node A and later replay this password packet claiming to be node A. The attacker would now have complete access to the network and operations being performed.

2.3.3 Wormholes

Wormhole attacks are a common concern in terrestrial networks and have expanded as a viable concern into the underwater domain. An example of a wormhole attack is when a group
of malicious nodes broadcast to the network that they have a better routing path in terms of transmission delay. This path is known as the wormhole. As we can observe in Figure 3, the wormhole is connected through a “better” link, commonly using a cable connection that can provide less delay. Using a wormhole, the malicious attackers can cause packet collisions, packet corruption or packet modification. Wormhole attacks can be launched in a more novel way in UANs because of the nature of underwater networks. In an underwater network, wormholes can be created using the surface or gateway buoys above the water. Since radio waves in air are faster than acoustic waves in water, the surface buoys can be used to significantly bypass parts of the network and create more powerful wormholes. This approach is shown in Figure 4 and has yet to be studied in literature.

Figure 3: Example Wormhole Attack

Figure 4: Example UAN Specific Wormhole Attack
2.4 Security Requirements

Through the development of terrestrial networks, researchers and practitioners have worked to generate a common set of requirements for network security. These requirements extend to UANs and are the following: confidentiality, authentication, integrity, freshness and availability. These requirements cover all types of network vulnerabilities but it can be difficult to accomplish all five requirements. What is often observed in literature is security mechanisms will specifically focus on providing a subset of these requirements. In some cases, entire protocol suites have been developed to cover all or almost all of the listed requirements.

2.4.1 Confidentiality

Confidentiality is the process of making sensitive data secure and hidden. In order to ensure that data is hidden or secure, a malicious adversary should not be able to obtain any useful information from the data or be able to alter the sensitive message. Common ways to provide confidentiality is through the use of a key distribution system and a type of encryption.

2.4.2 Authentication

Authentication is the process of guaranteeing that the message or data comes from a trusted and authorized source. In many cases where encryption and key distribution are used, authentication is also provided. This is due to the fact that we need to know the origin of the message to make sure it is legitimate. A malicious attacker that knows the secret key used for encryption can then send its own messages to other nodes in the network.
2.4.3 Integrity

Integrity is required to guarantee the data received at any node in the network has not been altered after being transmitted from its source. Many network attacks involve the use of changing small amounts of information in the transmitted message which can drastically jam or lower the performance of the network.

2.4.4 Freshness

Data freshness is required to guarantee the messages being received by a node are new or fresh messages. A malicious attacker could eavesdrop on a network, even one using encryption, and copy that message and replay it later.

2.4.5 Availability

Availability is known as a method of assurance that the network is robust enough to provide services even when the network suffers from an attack or performance degradation. Various solutions to network attacks for terrestrial networks involve “containing” the attack in such a way that as long as the network conditions or performance allows nodes to work near normal performance then it is okay to suffer from an attack.

2.5 State-of-the-Art in UAN Security

Adequate solutions for secure communication and networking in terrestrial networks have been proposed, ranging from complex cryptographic operations to trust based privacy schemes
However, existing solutions from terrestrial networks can not be directly applied to UANs because of the unique features of the underwater environment and channel dynamics. In addition to present challenges, research on security related issues and vulnerability studies are lacking. In this section we present a few related works.

2.5.1 Physical Layer

A key generation scheme for UANs by exploiting physical layer properties to generate a secret key between a sender and a receiver is presented in [58]. The key generation is done by using robust secure fuzzy information reconciliators (RSFIR); an adaption of the work in [72] to fit the unique needs of the ocean environment. The work claims that traditional key generation schemes, such as Diffie-Hellman, will not meet the needs of UANs and this motivates their work. The work assumes a reliable acoustic channel between two nodes and transmits a pure tone signal over a period of time, synchronously, from one node to another. Using this transmitted tone, each node can produce a secure key that an eavesdropper cannot compute. The authors analyze their scheme numerically using two nodes transmitting to each other at a depth of 55m in a medium-range (no exact transmission distance is provided). The results show that key generation is successful, even in the presence of errors. The work claims that this key generation scheme is also computationally and energy lightweight but no analysis is provided to support this claim.

In [57] the authors study effectiveness of multipath routing in the presence of unintentional jamming attacks. In this work, the authors are interested in using acoustic communication for long transmissions in intruder detection scenarios. However, the frequency band being used (4 to 8 kHz) is highly affected by noise originating from the propellers of boats, which could be intruders or friendly ships. This noise is classified as unintentional jamming because it can disrupt network communications but is not an active threat or adversary. The performance
of three proactive routing protocols is investigated in the presence of this unintentional noise. Proactive routing makes use of message exchanges to find routing paths before network operations begin. The authors argue feasibility given the static network topologies being used. The three protocols studied are Single-Path routing (SP), Restricted Flooding (RF), and Multi-Sink Routing Protocol (MSRP). Analysis of these protocols is performed using a simulation environment and the authors do not consider MAC protocols or error control schemes. The authors explore the jamming noise from 120 dB re uPa (low power) to 180 dB re uPa (high power) and provide a comparison of each protocol using results from packet delivery ratio, packet overhead, delivery delay and number of hops traveled. RF and MSRP are shown to perform better than SP and RF is shown to perform better than MSRP in the presence of high jamming noise. However, replication of data packets is performed to ensure data delivery, which increases the average delay delivery of packets. While not shown but mentioned briefly, the performance of RF is expected to decrease if the routes (hop count) are increased between sinks.

2.5.2 Network Layer

The effects of wormhole attacks on UANs are investigated in [61]. It is shown that in underwater networks, wormhole links of any length (longer or shorter than one hop) can be implemented and disrupt communication and networking services while incurring low cost to the attacker. Analysis is shown using theoretical (geometric) analysis on wormhole impact and effect of the length of wormhole and then shown in simulation using CSMA at the link layer and AODV at the network layer. Simulations show the packet delivery ratio (PDR) decreases from about 90% to less than 10% when wormhole pairs increase from 0 to 8. When more than 2 wormhole pairs exist, the PDR falls to less than 50%. Existing solutions to addressing this attack in terrestrial networks are presented, such as RF watermarking, packet leashes which use temporal and geographical information to limit the transmission range of packets, distance
bounding schemes and directional antennas. The authors provide a detailed review of existing wormhole work in terrestrial networks and well separate the differences between terrestrial and underwater networks. The work also briefly investigates the use of network wise localization to detect wormhole links but conclude this incurs heavy overhead which reduces the ability to use current localization schemes.

A distributed technique, known as Distributed Visualization of Wormhole (Dis-VoW), to detect wormhole links in underwater networks is presented in [60]. Dis-VoW has sensors collect distance estimations of its neighbors using a simple protocol to determine the upper and lower bounds of distance estimations between two nodes. Then, each node will reconstruct the local network topology using Multi-Dimensional Scaling (MDS). Utilizing this technique, distortions in edge lengths and angles among neighboring nodes can be reconstructed to locate fake connections. No special hardware is needed to perform this technique. The approach is validated using simulations results and shows that Dis-VoW can detect most of the fake connections without introducing many false positive alarms. While the technique performs relatively well, it suffers from deterioration in detection accuracy as more wormholes are introduced. Additionally, without special hardware, estimation errors will increase, which are shown to affect the false positive alarms. While this approach is useful to detect wormhole links in underwater networks, the feasibility of implementing this technique in a real underwater network is not known. How will the message overhead and computation at each node affect normal operations and lifetime of the network? These items are not discussed.

A set of neighbor discovery protocols that are resilient to wormhole attacks is introduced in [59]. The objective of this work is to prevent fake neighbors from establishing neighboring relationships and detect wormhole links. The work defines a basic attack model where the adversary aims to interrupt neighbor discovery and network operations by creating fake links.
using wormholes. Using Direction of Arrival (DOA) estimates of acoustic signals, which requires added physical hardware (an array of hydrophones in the case of this work), the authors formalize a model to prove that nodes are true neighbors if they satisfy some geometric relationships. The idea for exploiting this signal property is given by the nature that DOAs are solely dependent on the relative locations of signal transmitters and receivers, which cannot be easily manipulated like power and transmission time. However, the work assumes that all nodes have the same X, Y and Z axis orientations with negligible errors. Exploring orientation errors is still an open problem for wormhole discovery. Simulation analysis is provided on three different neighbor discovery protocols. The first protocol involves only two nodes, the second involves three nodes, which results in better wormhole resilience, and the third accommodates node mobility. The results demonstrate that fake wormhole links can be minimized and detected with high probability. The authors show that existing wormhole detection and protection techniques, such as distance-bounding solutions, do not work in UANs given its unique features. The authors also support that cryptographic methods are not able to solve many security problems. Interestingly, the authors do assume that their protocols use a key-distribution scheme to provide authentication. Another important note is secure localization in UANs is mentioned as a current difficult problem with no available solutions and has potential to help protect against many attacks.
Chapter 3

Launching Jamming Attacks

The underwater acoustic channel is an open environment, making Underwater Acoustic Networks (UANs) vulnerable to security concerns. In many cases, a receiver of a transmission can not distinguish an authorized signal from a malicious signal. Further, channel conditions can be affected by environmental properties, such as temperature, salinity and depth. The dynamic nature of the underwater channel can result in high levels of packet error rates. Since many factors can effect the underwater channel, it is necessary to study the feasibility of jamming attacks on UANs to better understand how malicious signals can interfere with normal network operations.

UANs are subject to jamming attacks when an adversary injects unwanted signals into the communication channel. These malicious signals can corrupt packets being transmitted, if the malicious signal and legitimate signal overlap, or they can occupy the communication channel and prevent legitimate signals from being transmitted. A jamming attack is known as an effective network disruption method because of the three following reasons:
An attack can be performed by listening for network communication, then broadcasting in the same area as the network, on the same frequency band;

A well planned attack can significantly degrade the performance of a network while incurring only a small cost to the attacker; and

In general, no special hardware is needed to launch an attack.

In order to understand how jamming attacks effect UANs, it is necessary to develop our own jamming attacks and attack a real network system. Much of the research conducted on UANs is through the form of simulation and modeling. This is generally because it is cheaper and faster to perform experiments and gather results than through the use of real-world systems and field experiments. However, due to the inherent properties of using acoustic communication underwater, it is difficult to accurately model the networks and signal characteristics in simulations [73]. Many research initiatives are working to solve these problems [73–78]. For this reason, we focus on an experimental field test approach using Aqua-TUNE [25], a testbed system, to conduct our study on jamming attacks. This will allow us to perform real-world field tests to accurately study the characteristics of jamming attacks.

3.1 Attack Models

A jammer can be categorized into two types based on the signals being used to try and jam the network. These two types are Dummy (Signal) Jammer and Smart (Deceptive) Jammer.

- **Dummy (Signal) Jammer**: This type of jammer does not know anything about the underlying protocols being used in the network. It will simply generate a signal that can be considered as noise or replay a packet to overlap transmissions to jam the network.
• **Smart (Deceptive) Jammer**: This type of jammer knows some information about the protocols being used in the network and will try to manipulate this information. For example, the jammer might know the underlying medium access control (MAC) protocol of the network and refuse to follow this protocol or take advantage of the scheduling features. This type of jammer pretends to be a legitimate node in the network, unlike the dummy jammer.

These jammers can both perform the following attacks, similar to those mentioned in [79].

• **Constant Attack**: A jammer will continually inject signals into the channel to corrupt packets or introduce congestion into the network.

• **Random Attack**: A jammer will alternate between sleeping and jamming the channel.

• **Reactive Attack**: A jammer will remain idle while the channel is idle until a transmission is observed. Then the jammer will start transmitting to jam the network. This attack is considered the most sophisticated.

### 3.2 Network Metrics

In literature a few network metrics have been identified as being potentially affected by the presence of a jammer [79]. We will study these metrics to observe whether or not they provide preliminary insight into the effects of jamming attacks in UANs. We introduce and define a few metrics recorded in our analysis.

• **Packet Delivery Ratio (PDR)** can be measured in a few ways. The first and most common measurement is at the receiver as the ratio of the number of packets that pass the Cyclic Redundancy Check (CRC) with respect to the number of packets received overall. It can also be measured at the sender as the ratio of acknowledgments received
to those packets sent out by the sender or as the number of packets sent by the sender to the receiver over the number of packets the receiver actually received.

- **Packet Send Ratio (PSR)** is measured as the ratio of packets that are sent successfully by a source node compared with the number of packets that were intended to be sent, which is recognized at the MAC layer of the network.

- **Network Throughput (TP)** is related to PDR. TP is the average rate of successful message delivery over the channel. This metric can also be measured as system throughput which is the sum of data rates delivered to all nodes in the network.

### 3.3 Experimental Study - Signal Jammer

In this section we will provide an overview of the jammer hardware and signals used for our study. We perform three case studies, each on different commercially available acoustic modems. Additionally, an effective jamming scheme that limits energy consumption and transmission time will be presented.

The jammer type used in these studies is the **dummy (signal) jammer**. Using this jammer type we utilize four jamming models: constant, random, reactive and white noise. The jamming signal used for our jamming attacks in these experiments is a recorded version of a regular transmission from each modem. To be more specific, we transmit a packet using each acoustic modem and record this transmission. This will result in three recorded transmissions which we will use as the jamming signal for each modem respectively. The jammer was created using limited hardware that consisted of an amplifier, to increase the signal power for proper underwater use that would match that of the acoustic modems being used, an ITC-1032 deep
water omnidirectional transducer and a 12-V battery to power the device. The power consumption of our jamming equipment is 4 watts during idle state and 4.476 watts at full transmission power. The jamming hardware can be seen in Figure 5.

![Figure 5: Jammer Hardware](image)

### 3.3.1 Benthos Acoustic Modems

In this case study we make use of the Benthos ATM-885 acoustic modem [80] which is based on multiple frequency-shift keying. Our study is performed in Mansfield Hollow Lake, Mansfield, CT USA and the field test deployment can be seen in Figure 6. This figure shows the locations of the sender and receiver nodes and the variations in the distances of the jammer from the receiver. In our experiments the network consisted of one sender and one receiver which are spaced 298.8 meters apart. The jammer was placed in four different positions: 400 meters, 298.8 meters, 167.3 meters, and 75 meters away from the receiver. These positions correspond to JP1, JP2, JP3, and JP4 respectively in Figure 6. Broadcast Mac (BC MAC) was the MAC protocol used in the network, with a sending rate of 0.04. BC MAC is a simple MAC protocol that does not involve any feedback mechanisms. If a sender has data to send, it
will send the data. This protocol does not consider any information about network conditions. The sending rate is based on a Poisson traffic generator that runs on the application layer. The traffic generator sends out data packets in such a way that the inter-departure time between two consecutive packets follows the Poisson distribution. Each jamming test was run for ten minutes with the power level of the nodes and jammer being set to 1 (an average transmit power of 2 watts), and the sender used packets with a length of 400 bytes. Tests were carried out for longer durations but we omit the results as there was no difference when increasing the length of the experiment.

In our preliminary investigation we analyzed the variations of the network metrics in several network conditions. The results can be summarized in Table 1. The jammer was placed 298.8 meters away from the receiver for these tests. We measured the average PDR (in %), PSR (in %), and TP (in bits per second) in normal network conditions where no jammer is present and in network conditions where a jammer is present varying the attack model. In each test we allowed the first packet to get through to the receiver to ensure packets could be delivered. After the first packet was delivered the jamming attack was started.

Figure 6: Field Test Deployment
Using the information, as organized in Table 1, interesting observations can be made. First, we were unable to jam the network using an increased white noise level as no packets were lost. The network performed approximately the same as it would during normal operation. Second, constant and reactive attacks perform the best, although the reactive attack model is more energy efficient because it does not need to continually jam the network. Third, while we could jam packets from being received at the receiver, we were unable to stop the network from actually transmitting packets. In Figure 7 we plot the network throughput of each model from the entire first test, which is summarized in Table 1.

<table>
<thead>
<tr>
<th>ATTACK</th>
<th>PDR (%)</th>
<th>PSR (%)</th>
<th>TP (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Operation</td>
<td>100</td>
<td>100</td>
<td>140.0660</td>
</tr>
<tr>
<td>Constant Jammer</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Random Jammer</td>
<td>54</td>
<td>100</td>
<td>30.09</td>
</tr>
<tr>
<td>Reactive Jammer</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>White Noise</td>
<td>100</td>
<td>100</td>
<td>139.8205</td>
</tr>
</tbody>
</table>

Table 1: Network Performance

Figure 7: Throughput of Jamming Attacks
3.3.1.1 Efficient Jamming Scheme

We further studied the effects of jamming on UANs by investigating when an attack needed to be launched to effectively jam the network while minimizing energy consumption and the probability of detection. A diagram of the transmission signal being sent from the Benthos ATM-885 acoustic modem can be seen in Figure 8. The signal consists of two main parts. The first part is a preamble or header, which lasts for roughly 1.5 seconds and sets up synchronization. The second part of the signal, the payload, is where the actual data is sent. The length of this transmission will vary based on the size of the packet.

![General Signal Diagram](image)

Figure 8: General Signal Diagram

Using the characteristics of this signal, we performed another set of experiments to determine the most effective time to attack the network. Our results can be summarized in Table 2. In this table we compare the starting time of the jamming attack versus the distance of the jammer from the receiver. The power levels were the same as the previous experiment discussed in Section 3.3.1. We observe that the most effective time to jam the network is during the preamble. Specifically, launching a jamming attack in the first second will ALWAYS jam the communication channel. During our experiments we tested a jamming signal consisting of 1 second of a recorded preamble and 1 second of our jamming signal. Both started at 0.5 seconds into the legitimate transmission signal. Each jamming signal was able to jam the network and prevent the receiver from obtaining any information from the sender.
The payload portion of the signal, where the data is transmitted, is much more robust to jamming attacks. This is due to the design characteristics of the signal. In fact, the Benthos modem makes use of Frequency Hopping (FH) techniques that allows the payload portion to be more resistant to interference and jamming attacks. FH is the process of placing short continuous wave tonals at specific locations in the frequency band as time passes [81]. Essentially, the transmitter uses a pseudorandom sequence to rapidly switch a carrier among the various frequency channels. This pseudorandom sequence is also known by the receiver. Since FH is employed in the payload portion of the signal, a different jamming attack scheme will be required to jam the data transmission. One potential scheme would be to analyze the waveform and try to uncover the hopping pattern. While this is a challenging task, it is certainly the most effective. Another scheme is to increase the power of the jammer to be more than the legitimate sender. However, this approach is not power efficient. Regardless of the scheme, the preamble portion is an easier target since it has shown to be less tolerable to jamming attacks.

### 3.3.2 AquaSeNT Acoustic Modems

Multicarrier modulation in the form of orthogonal frequency division multiplexing (OFDM) has prevailed in recent broadband wireless systems over radio channels [10]. The success of OFDM in radio channels has motivated research in applying OFDM in underwater acoustic channels. Using a prototype version of an Aqua-SeNT OFDM acoustic modem [82], we conducted another set of experiments to analyze the effects of jamming attacks on these modems.
The hardware used for the jammer in these experiments is an OFDM modem that is programmed to act as a jammer instead of a legitimate node. The jamming signal was a recorded sample of a preamble from the OFDM modem, lasting 245 milliseconds.

The field test deployment can be seen in Figure 9. In this experiment we had one sender and one receiver, which are placed 200 meters apart. The jammer was placed 200 meters away from the receiver. We use the same network parameters mentioned in Section 3.3.1.

![Figure 9: OFDM Field Test Deployment](image)

We conducted similar experiments as in Section 3.3.1.1 to develop an efficient jamming scheme. A general signal diagram of the OFDM signal transmission can be seen in Figure 10. The OFDM signal transmission consists of a detection preamble, a brief pause followed by a synchronization preamble and then transmission of the data in blocks.

We introduced our jamming signal into the network during three different phases of the signal transmission. The first test was to transmit the jamming signal during the preamble of a...
legitimate transmission. The result of this experiment demonstrated the jammer was successful as it caused a collision with the legitimate packet. The receiver did not receive any information and was unaware of the transmission. The second test was to transmit the jamming signal after the preamble during the legitimate transmission of the first data block. This caused the first data block to become corrupted and the receiver reported a decoding and CRC error. No data was able to be received. The final test was to transmit the jamming signal during the transmission of the second data block. This attack caused the same result as the previous attack. The receiver reported a decoding and CRC error and no data was able to be received.

3.3.3 Linkquest Acoustic Modems

A similar set of experiments were performed using a different brand of commercial modems. Specifically, the LinkQuest UWM2000H [83]. Similar location and parameters were used as in the previous case study. We recorded the signal transmission of these modems to use as our jamming signal and attempted to jam the network at different phases in the transmission. However, we were unable to effectively jam the network while using these modems. While packet loss was experienced, this loss was too infrequent to confirm it was due to our jamming attack scheme. When analyzing the signal, we observed this modem also employs some type of code hopping technique to counter communication interference. It is apparent a more effective jamming scheme involving some type of signal processing to determine the hopping pattern is required.

3.3.4 Discussion

Our experiments have shown that commercial brand modems and the OFDM modem prototype are vulnerable to malicious jamming attacks. However, it is important to note that reactive attacks could perform differently on the commercial brand modems depending on how
long it takes an attacker to sense the channel is being used. In our experiments, the attacker could sense the transmission almost immediately and was able to start transmitting its jamming attack without much delay. If an attacker takes longer to sense the channel transmissions then it is possible that the attacker will miss the opportunity to jam the channel.

![Diagram showing Preamble Time Constraints]

In Equation 1 we formulate a timing constraint for jamming attacks. This scheme must be followed by the underlying hardware and software in order to effectively jam legitimate transmissions. When a node starts a legitimate transmission, a malicious attacker will require a small period of time to sense or detect the start of this transmission. We denote this as $t_{\text{scan}}$. The jammer will then use a smaller time slot, $t_{\text{start}}$, to start transmitting its jamming signal. The jamming signal will then last for a period of $t_{\text{jam}}$, which includes signal propagation time or delay. These events must be shorter or at least the same time period as the entire preamble. We visualize these events in Figure 11.

$$t_{\text{scan}} + t_{\text{start}} + t_{\text{jam}} \leq t_{\text{preamble}}$$ (1)

It is also evident from our experiments that UANs can be jammed by overpowering the legitimate nodes with its own signal transmission. Essentially, a jammer could raise its power level to be greater than a legitimate node and this will often overwhelm the intended receiver, causing it not to hear the legitimate transmission. However, this approach is not efficient in regards to energy conservation and detection constraints.
3.4 Experimental Study - Deceptive Jammer

In the previous sections we have studied the effects of dummy (signal) jammers on UANs and have shown that these types of attacks have a significant disruptive impact on the network. As mentioned in Section 3.1, another type of jamming attack is the smart (deceptive) jammer. Recall that this attack is where the jammer pretends to be a legitimate node in the network. This jammer has some knowledge of the underlying protocol stack and therefore, instead of jamming with a signal or noise, it will attempt to jam the network using control or data packets. In our deceptive attacks we disguise our jammer as a legitimate node in the system that tries to jam the communication channel using an increased sending rate.

This experiment was carried out in a shallow water lab tank setting using the OFDM modems. The test consisted of one sender, one receiver and one jammer. The same MAC protocol and packet size as the previous sections were used and the sending rate of the legitimate jammer was fixed at 15 second intervals. The jammer sending rate (JSR) was varied at 5 seconds (attack 1), 3 seconds (attack 2), and 1 second (attack 3). We present the results of these experiments in Table 3, 4, 5 respectively, where ERROR and LOSS refer to decoding/CRC errors and packet loss respectively. We observe when two packets are sent out simultaneously they will collide and introduce packet loss. Additionally, when a packet is sent out just after a previous packet, it will result in a decoding/CRC error.

<table>
<thead>
<tr>
<th>NODE</th>
<th>ERROR(%)</th>
<th>LOSS(%)</th>
<th>PDR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legitimate</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Jammer</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3: Deceptive Jammer Attack 1 (JSR=5s)

<table>
<thead>
<tr>
<th>NODE</th>
<th>ERROR(%)</th>
<th>LOSS(%)</th>
<th>PDR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legitimate</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jammer</td>
<td>6</td>
<td>0</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 4: Deceptive Jammer Attack 2 (JSR=3s)
### Table 5: Deceptive Jammer Attack 3 (JSR=1s)

<table>
<thead>
<tr>
<th>NODE</th>
<th>ERROR(%)</th>
<th>LOSS(%)</th>
<th>PDR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legitimate</td>
<td>93</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Jammer</td>
<td>7</td>
<td>18</td>
<td>75</td>
</tr>
</tbody>
</table>

#### 3.5 Theoretical Study - Jammer Deployment

The outcomes of our experimental studies have proven the effectiveness of a jamming attack revolves around the timing and placement of the attack signal. For this reason, we briefly investigate the theoretical placement for a jammer. Leveraging the existing acoustic signal propagation model in water, we can determine where a jammer can be placed to disrupt communications. We make use of [84] to derive a similar approach. The passive sonar equation characterizes the signal-to-noise ratio (SNR) of acoustic signals at the receiver [85]:

\[
SNR = SL - TL - NL + DI
\]  

where \( SL \) is the source level, \( TL \) is the transmission loss, \( NL \) is the noise level, and \( DI \) is the directivity index. The quantities in Equation 2 are in \( dB \text{ re} \mu Pa \), where the reference value of 1 \( \mu Pa \) is \( 0.67 \times 10^{-22} \text{ watts/cm}^2 \) [85]. For convenience, we use \( dB \) to represent \( dB \text{ re} \mu Pa \).

The ambient noise level is dependent upon many factors of the environment such as shipping traffic and wind level. We consider shallow water environments where a representative average value for the ambient noise level \( NL \) is about 70 \( dB \) [84]. We further assume omnidirectional hydrophones are used, the directivity index, \( DI \), is then 0. Therefore, we have:

\[
SNR = SL - TL - 70
\]  

The propagation of acoustic signals in shallow waters generally follows a cylindrical pattern in which they are bounded by the water surface and the sea floor. This means the spreading
of the signal will be cylindrical. We can use the equation provided in [85] to approximate the transmission loss $T_L$ for cylindrically spread signals as follows:

$$T_L = 10 \log d + \alpha d \times 10^{-3}$$  \hspace{1cm} (4)

where $d$ is the distance between the source and receiver in meters, $\alpha$ is the frequency dependent medium absorption coefficient, and $T_L$ is in $dB$. The value of $\alpha$, taken from [84] at temperatures between $4^\circ$C and $20^\circ$C, is listed as follows:

$$\alpha = \begin{cases} 
0.0601 \times f^{0.8852} & 1 \leq f \leq 6 \\
9.7888 \times f^{1.7885} \times 10^{-3} & 7 \leq f \leq 20 \\
0.3026 \times f - 3.7933 & 20 \leq f \leq 35 \\
0.504 \times f - 11.2 & 35 \leq f \leq 50 
\end{cases}$$  \hspace{1cm} (5)

where $f$ is in kHz and $\alpha$ is in dB/Km. We can see that $\alpha$ increases as $f$. Hence, higher frequency signals attenuate faster. This absorption is used for seawater environments and will not apply in our experiments, since they are conducted in a freshwater lake. Additionally, the assumed noise level and spreading mechanism can differ substantially from actual conditions. According to Equation 3, we have:

$$SNR = SL - \left(10 \log d + \alpha d \times 10^{-3}\right) - 70$$  \hspace{1cm} (6)

With Equation 6 and an expected $SNR$ at the receiver, $SL$ can be estimated, as well as the transmission power at the source. In jamming attacks, the signals from the jammer can be considered as noise at the receiver. The jamming signals also attenuate, and much faster than radio does in air. In order to corrupt the packets at the receiver, an acoustic jammer has to be in a close range of the target to generate stronger jamming signals.
Using the SNR equation above, we can determine the jamming region of a constant jammer in shallow water. We show a simple case, where there is only one sender, one receiver and one jammer. Both the sender and jammer transmit at frequency of 12 kHz, representative of the commercial modems used in our field tests. The distance between the sender and receiver is \( d \). The distance between the jammer and receiver is \( d' \). The power level and transmission loss of the sender (and jammer) are \( SL \) and \( TL \) (and \( SL' \) and \( TL' \)), respectively. The noise level caused by the jammer at the receiver is \( JM \). SNR at the receiver can be computed as \( SNR = SL - TL - 70 - JM \) where \( JM = SL' - TL' - 70 \).

According to Equation 4, we have \( TL = 10 \log d + \alpha d \times 10^{-3} \) and \( TL' = 10 \log d' + \alpha d' \times 10^{-3} \). Therefore:

\[
SNR = (SL - 10 \log d + \alpha d \times 10^{-3} - SL' + 10 \log d' + \alpha d' \times 10^{-3})
\] (7)

If we assume the jammer and the sender use the same power level, i.e. \( SL = SL' \), we can further reduce the above to:

\[
SNR = 10 \log \frac{d'}{d} + \alpha (d' - d) \times 10^{-3}
\] (8)

If the receiver requires \( SNR \) to be at least 10 dB to decode the data packet correctly, then the goal of the jammer is to make \( SNR \) at the receiver less than 10, i.e:

\[
10 \log \frac{d'}{d} + \alpha (d' - d) \times 10^{-3} \leq 10
\] (9)

Given a specific \( d \), the maximum distance between the jammer and the receiver can be determined. This idea is illustrated in Figure 12. From this figure, we can see that the jammer can be located farther away from the sender, in other words, the jamming range is larger than the transmission range. As the distance between sender and receiver increases, the advantage
of the jammer decreases until a certain point at which point remains constant. For instance, if the sender and receiver are about 100 meters away, the jammer can be located at 300 meters away, three times the distance between the sender and receiver. However, when the distance between the sender and receiver is 1000 meters, the jammer has to be located within 2450 meters, roughly two and a half times the distance between the sender and receiver. While this trend remains consistent for even larger distances, it is dependent upon the frequency.

Another observation from Figure 12 is the effectiveness of a jammer is not constant. For example, a jammer can be placed 1500 meters away from the targeted receiver if the sender is 1000 meters away. However, because of node mobility, the sender may get closer to the receiver. If their distance is less than 600 meters, the jammer will not be effective any more.

Figure 12: Distance Requirements for Constant Jamming Attacks
3.6 Summary

In this chapter we performed experimental real-world field tests to introduce and analyze various types of jamming attack models in UANs. The effectiveness of different jamming attack models were compared using network statistics. We created our own jamming attack signals and hardware to study the characteristics of each jamming attack and the performance of the network being attacked. Our tests were conducted on commercial brand modems. Additionally, a placement scheme for the maximum distance between a jammer and receiver was provided. Experimental results demonstrate that jamming attacks on UANs can easily be launched and drastically degrade the performance of the network. This work provides the first study of jamming attacks in UANs and will allow researchers to use new insights to develop adequate detection and mitigation schemes for these attacks.
Chapter 4

Spoofing Geographic Routing

Routing protocols move data from point $A$ to point $B$ within the network. As such, these protocols sit on the network layer of the protocol stack and are active targets for malicious adversaries. Routing protocols are generally categorized as reactive routing, also referred to as on-demand routing, or proactive routing. However, protocols in these categories do not work well in Underwater Acoustic Networks (UANs). In on-demand routing, the routing procedure is initiated by the communication demand at a source node. During route discovery, the source seeks to establish a route towards the destination by flooding route request messages. In UANs this is costly given the long and variable propagation delays, leading to a higher collision probability, the “power hungry” operation of acoustic communication and higher bit error rates. This degradation increases as the scale or size of the network increases. Proactive routing, which makes use of routing tables and has nodes constantly updating their tables, also suffers from the same costly issues. Further, the effect of mobility on nodes amplifies the above issues. Many aquatic applications are not constrained to a specific area and deployed nodes often drift due to ocean currents and sea conditions. Network topologies can rapidly change even with
small displacements in node positions. This makes multi-hop packet delivery challenging in UANs [14].

In order to provide scalable, efficient and robust routing in UANs, researchers are looking towards geographic routing mechanisms. Geographic routing relies on location information to send data to specific geographic destinations instead of using network addresses [86]. For geographic routing to be successful, each node is required to know some location information about itself and the sender must have a specific location in mind for the destination of data. Using this technique, a message can be routed to any location without knowledge of the network topology or prior established routes. For the purposes of UANs, geographic routing has become an effective technique in order to solve routing issues and extend underwater applications such as aquatic monitoring and exploration, which are only useful with location-aware data. This is due to the necessity to associate sampled data with the 3D position it originated from, in order to spatially reconstruct characteristics of an event [14]. Therefore, the availability of location information is a requirement which helps to enable geographic routing. Further, geographic routing is proven to be more efficient than pure flooding in UANs and helps to lower the impact of node mobility on routing performance [14].

Geographic routing protocols are generally based on the broadcast nature of the acoustic channel. This can create collisions in packet forwarding in which most protocols require self-adaptation techniques to minimize the collision probability if possible. More importantly, from a security standpoint, broadcast based protocols are considered vulnerable to security attacks because packet information can be overhead by passive intruders or unauthorized nodes. In this section we investigate these vulnerabilities with emphasis on existing UAN geographic routing protocols, using Depth-Based Routing (DBR) [87] as a case study. Then we derive a preliminary attack model specific to geographic routing protocols and present the attack performance
using simulations. Finally, we develop an improved attack model to sample locations in the network to maximize attack effectiveness.

4.1 Related Work

The security of geographic routing in terrestrial networks has not gained much attention [69] but defenses against potential attacker have been proposed. In [66], the authors survey routing mechanisms for terrestrial networks and discuss potential vulnerabilities. Two geo-routing protocols, Geographic and Energy Aware Routing (GEAR) and Greedy Perimeter Stateless Routing (GPSR) are shown to be insecure against Sybil attacks, which are attacks that forge fake location information. However, because the focus is on general routing mechanisms, no specific analysis of the attacks is given. A resilient geographic routing scheme based on probabilistic multipath routing and trust management is proposed in [69]. Location verification, involving the use of Radio Signal Strength (RSS), Time of Arrival (TOA), Time Difference of Arrival (TDOA) and Angle of Arrival (AOA), is used to mitigate false neighbors. Additionally, each node holds a routing table with an associated trust value, which is updated at each transmission. Securing geographic routing in vehicle networks is explored in [70]. Using various sensors in the network, the trustworthiness of a node’s claimed position can be estimated. This approach does not need a dedicated infrastructure but does require that every node knows its exact location with use of GPS. Similar work on secure geographic routing in vehicle networks is presented in [71]. This work also addresses secure vehicle communication by using GPS to gather exact position information and then enforces plausibility checks, such as time and velocity requirements, to ensure messages are coming from legitimate locations. However, these existing defenses are not able to be directly applied to UANs because of the inherent issues with UAN communication and system constraints, as mentioned in Section 2.3.
4.2 Vulnerability Analysis

In this work we will focus on a standard geographic routing protocol for UANs, known as Depth-Based Routing (DBR). DBR [87] is a depth-based protocol that utilizes the unique properties of UANs: specifically, that data sinks are generally located at the water surface. DBR greedily forwards data packets towards the water surface based on the depth information of each node. In DBR, a data packet will record a unique ID when first created and the depth information of its most recent forwarder. The depth information is updated at every hop with the depth of the node forwarding the packet. The basic principle of DBR is that when a node receives a packet for the first time, it will queue this packet for forwarding if that node’s depth is less than the depth recorded in the packet from the sender (i.e. \( d_{\text{receiver}} < d_{\text{sender}} \)) and the depth difference between that node and the sender is greater than some predefined threshold (TH) (i.e. \( (d_{\text{sender}} - d_{\text{receiver}}) > \text{TH} \)). Otherwise, it will discard the packet. However, depending on the topology, a packet may be forwarded along multiple paths to the sink or void areas, which is not addressed in DBR.

In order to reduce redundant forwardings, DBR includes a mechanism to suppress redundant packets. This mechanism is known as the holding time. When a node receives a packet, there might be multiple nodes in the same area that qualify for forwarding, therefore each node will hold the packet for a certain amount of time after it is queued, the holding time. A node that is closer to the surface of the water will have a shorter holding time, resulting in a higher priority to forward the packet. When this node forwards the packet, its neighbors will receive the packet. Any node that receives a duplicate packet during its holding time will check a similar constraint as above. If \( d_{\text{receiver}} \leq d_{\text{sender}} \) and \( (d_{\text{sender}} - d_{\text{receiver}}) > \text{TH} \), it will update its holding time with the minimum holding time of the two. Otherwise, it will drop this packet. Therefore, neighbors at a lower depth, for example physically below the optimal receiver, who
already have the packet but are in the holding phase, will become suppressed and discard the packet. The neighbors at a higher depth will receive the packet and enter their own holding phases. This process will continue until the packet reaches the intended sink at the surface.

Figure 13: Visualization of a Typical DBR Scenario

We further explain how DBR works using Figure 13, where we have one sender, S, and three receivers, R1, R2, R3. The sender will record its depth information into the data packet and broadcast this packet, in which all nodes inside the sender’s transmission range (the dashed line) will hear, R1, R2, R3. Each receiver will compare their depth with the depth recorded in the packet they receive. In this scenario, R3 will immediately discard the data packet since it came from a node that is closer to the surface. R1 and R2 will compare the depth information and both observe that they are potential candidates to forward the packet. Both nodes will then hold the packet for their calculated holding time. R2 is closer to the surface than R1 and therefore has a shorter holding time and will forward the packet first. It is the optimal next hop in the network. Once R2 forwards the packet, R1 will receive this packet during its holding time and then discard the original packet from S because it is no longer a good candidate to forward.
4.2.1 Security Vulnerabilities

Geographic routing protocols route packets based on distance information and do not need to consider connectivity or state information. This technique increases security concerns because of the use of location information, which is included in each data packet transmitted in the DBR protocol. Since no governing mechanism exists to verify that a node is in fact at the position it is claiming, malicious attackers can easily exploit the system. The specific weakness, or attacking point, of DBR comes from its heuristics to save energy by deterring redundant transmissions of data packets. The holding time, used to schedule the forwarding of data packets, allows for malicious users to exploit this protocol and implement various routing disruptions on the network.

While this work focuses on the use of DBR as the routing protocol, other geographic routing protocols designed for underwater networks use similar techniques to schedule forwarding. For example, Vector-Based Forwarding (VBF) [88] uses a desirableness factor in order to calculate how long the protocol should wait before forwarding a packet, known in VBF as $T_{adaption}$. This is a similar technique compared to DBR’s holding time. HH-VBF [89] is another protocol using this technique. Further, HydroCast [90] uses the broadcast nature of the acoustic channel for nodes to overhear transmissions and calculate their own forwarding priority and then compare it against the distance in that transmission. Therefore, we claim the work in this paper can easily be adapted to other underwater geographic routing protocols.

4.3 Preliminary Attack Model

In this section we discuss the attack model for our adversary, assuming that the network of interest is stationary or static. We evaluate the case of mobile nodes in a later section. Also, we assume the transmission range for legitimate nodes and the attacker is the same. We formally
define $d_{surface}$ as the depth of the surface node with a value of 0 and $d_{floor}$ as the depth of the ocean floor. The network is bounded by $d_{floor}$ and $d_{surface}$ such that all node depths, $d$, are that of $d_{surface} \leq d_{floor}$. We also define the following, $d_{attacker}$ as the actual depth of the attacker, $d_{spoof}$ as the spoofed or fake depth of the attacker, $d_{sender}$ as the depth of the sender of the message, $d_{optimal}$ as the depth of the receiver or optimal next hop in the network and $R$ as the maximal transmission range of a node. We assume that the attacker is in range of an optimal node and a sender. We start with a network such that $d_{surface} < d_{optimal} < d_{attacker} < d_{sender} < d_{floor}$. This is represented in Figure 14, where $S$ is the sender, $A$ is the attacker’s actual position, $SA$ is the spoofed (fake) attacker position and $OR$ is the optimal receiver or next node.

The attacker will start in a passive mode where he will try to eavesdrop network transmissions between at least two nodes in his area. A sender or forwarder sends a message and the optimal receiver will get this message, calculate its holding time and then forward this message at the end of its holding time. Since the optimal receiver always forwards first, all other nodes in the area that received the original message will become suppressed upon hearing the optimal receiver’s message. Each packet in DBR will contain the value of $d_{optimal}$ and by having the attacker eavesdrop the transmissions in an area, it can determine the depth of the original sender or forwarder and its optimal receiver (the next forwarder). The attacker can then continue to listen to see if he hears any more forwarders. In some instances, the attacker might
be able to hear more than one hop given the topology conditions. In either case, the attacker will know the depths of the nodes sending or forwarding. When the attacker receives another packet from the original sender, he can then immediately forward that packet with his fake depth, $d_{\text{spoof}}$, encoded into the packet. Every node in the attacker’s transmission range will hear this message and then drop their packets. This is because they received the same packet from a node claiming to be at a better position. It does not matter if the legitimate packet or the attacker’s packet is heard first, in either case the packet will be dropped as long as $d_{\text{spoof}}$ is a better position than their own. This attack suppresses network traffic and ends the flow of data through this area completely because the attacker has mimicked that he is located above them and forwarding upwards.

The choice of $d_{\text{spoof}}$ decides the attack performance. Depending on many factors or the amount of knowledge known about the network, attackers may choose different values. If the attacker does not know much about the network at first, to ensure that no node will forward a packet, the attacker should set $d_{\text{spoof}} \leq (d_{\text{optimal}} - R)$. This will make it such that any node that receives this packet will instantly drop the original packet from its queue or ignore it (if not queued) since the depth in the packet is better than the node that might receive it. This is due to the fact that the depth fakes the location of a node just outside of the optimal receiver’s communication range. Since the attacker itself cannot communicate to the depth of $d_{\text{optimal}} - R$, because $d_{\text{optimal}} < d_{\text{attacker}}$, it is guaranteed that any node who receives the packet will be at depths greater than the depth in the packet. Therefore, the design of DBR ensures that no one will forward the packet. Furthermore, the constraints mentioned above in Section 4.2.1 for a node to be considered a forwarding candidate, enforce a minimum depth bound for the attacker. In order to be successful in attacking the network, the attacker must use a value for $d_{\text{spoof}}$ of at least the depth of the optimal receiver (DOR), otherwise it will be ignored. We
confirm this notion and further explore the performance of different values of $d_{spoof}$ in our experiments in Section 4.3.1.

This attack is powerful because it is easy to perform on a network and also maintains the integrity of the data being transmitted in the network. There is no need to manipulate or modify any data encoded in the packet, other than the depth information, which is already modified at every step of the forwarding process. Essentially, the attacker’s packet is the legitimate data but with a fake depth encoded into the packet. This limits the noticeable traces of the attack.

Figure 15: Comparison of Holding Times vs. Propagation Time

Figure 16: Intersection of Holding Times vs. Propagation Time
Another item to mention is that the time frame before a node sends or forwards a packet can be large. In order to calculate the holding time (which decides the time frame), DBR uses the following equation:

\[
f(d) = 2\frac{\tau}{\delta} \times (R - d)
\]  

(10)

where \(d\) is the difference between the depth of the sender and the node that received the packet and \(\tau = \frac{R}{v_o}\), where \(v_o\) is the propagation speed of sound in water, \(R\) is the transmission range and \(\delta\) is a parameter set by the network operator from 1 to \(R\). The \(\delta\) parameter is used to set how DBR should operate. With a larger value, each node will hold the packet for a shorter time, decreasing average end-to-end delay but increasing redundant forwardings and with a smaller value, each node will have a longer holding time with an increased average end-to-end delay. Simply stated, a smaller \(\delta\) minimizes unnecessary transmissions and decreases energy consumption.

In Figure 15 we compare the holding time using different values of \(\delta\), as the difference in depth between the sender and receiver increases, and the propagation time, as distance increases. We note that, given the scale of the graph, it is difficult to observe the propagation time is increasing over distance. As we observe from this graph, when DBR is not in a flooding mode, for example \(\delta\) is not larger than 50, the holding time is much longer than the propagation time. Therefore, an attacker has a larger buffer range to adjust its attack or wait longer to be more discrete. While the attacker does not need to know the exact holding time, as we send out our attack packet immediately after receiving a packet, picking a more discrete time to send is an interesting problem of its own. We leave this for future work. However, in Figure 16 we zoom in on the boxed area in Figure 15 and show that when a receiver is nearing the edge of the sender’s transmission range, its holding time is smaller than the propagation time. Therefore, the attacker has no room for waiting and must send its fake packet immediately.
4.3.1 Attack Performance

In order to test our proposed attack scheme we developed a simulation environment with a graphic visualizer that displays the network and how messages propagate. This environment is used to simulate the DBR routing protocol, the underwater communication channel and the attacker model. We developed the simulator using Java and refer the reader to [91] for architecture, implementation and fidelity details.

For our experimental analysis we generated 5 random fully connected networks, meaning that at least one path to the surface exists, in a 1000 meter $\times$ 1000 meter $\times$ 1000 meter area. Each deployment had 35 randomly deployed network nodes, 3 surface nodes, 1 randomly deployed attacker close to a network path, a transmission range of 200 meters, $\delta = 5$, a send depth = 900 meters and a threshold = 10 meters. A send depth means only nodes deeper than 900 meters will send new data, where as the remainder of the nodes will act as forwarders. A threshold is a parameter set by DBR that sets a constraint on how far away the next node must be in order to forward. In this case, a node must be greater than 10 meters above the sender/forwarder that it received the packet from in order to consider forwarding this packet.

The experiment was run for 300 seconds where data packets were sent randomly every 5 to 10 seconds for the first 90 seconds of the experiment, the rest of the time was for packets to finish propagating through the network. We average these results over 50 simulations. The results are displayed in Figure 17, where each trial number is the network number. The blue bar is the attacker efficiency, defined as the number of packets stopped by the attacker divided by the number of packets the attacker received or could stop. The green bar is the number of packets the attacker stopped or killed divided by the total number of packets sent out in the network and the red bar is the number of packets received at the surface nodes divided by the total number of packets sent out. We can draw from these results that the attack works well but the topology will play an important role in attacker effectiveness.
As mentioned in Section 4.3, $d_{\text{spoof}}$, or the faked location, will have an impact on the effectiveness of the attack. We have experimented with various spoofed depths on network 4 from the previous experiment. The settings and simulation details are the same. The results can be seen in Figure 18. In the first experiment, the attacker spoofs a depth of 0. The second experiment has the attacker spoofing the depth of the optimal receiver (DOR) in the routing path minus the transmission range (i.e., if the optimal receiver has a depth of 300 meters, the attacker spoofs a depth of 100 meters). We note that DOR is the depth of the optimal receiver in the attackers range. The third experiment uses a depth of DOR-10 meters, which is a depth just above the optimal receiver (a depth of 290 meters using the previous example). In the fourth experiment, the attacker pretends to be at the same depth as the optimal receiver, the fifth experiment is 1 meter below the optimal receiver and the final experiment is DOR+TH (a depth of 310 meters using the previous examples). The results show that any depth smaller than
DOR or DOR itself will perform at almost 100% effectiveness in this network. As expected, depths larger than DOR have no effectiveness. The reason for this is because of the constraints (discussed in Section 4.2) set by DBR on how a node decides if it will forward a packet or not.

We further tested the use of DOR as the value of $d_{spoof}$ on the other 4 networks from the first experiment. These results were also the same as using a value of DOR-TR for $d_{spoof}$. These results do not suffer major performance degradation because no other nodes exist in the attackers transmission range that might be a better candidate. In Section 4.4 we perform this experiments again using our improved attack model.

Another experiment was performed to analyze the effects of $\delta$ in the DBR protocol. This parameter is used to calculate the holding time as shown in Equation 10. Again, a larger value for $\delta$ will result in a shorter hold time, increasing redundant forwardings, and a smaller value will result in a longer holding time with reduced forwardings. We tested our attack scheme on network 4, the same network used in the depth experiments. All parameters and simulation settings are the same. We varied $\delta$ from 1 to $R$ and each value resulted in the same attack performance. The attackers efficiency was 100% and the number of packets killed was between 96% and 98%. It is clear that $\delta$ has no effect on attack performance.

### 4.4 Improved Attack Model

Network topologies are highly dynamic. Therefore, we have developed an improved attack model to traverse any given deployment area and find an optimal location to attack the network. This is a difficult task as the attacker has no knowledge of exactly how well its attack is performing on the network as a whole or even at a given area. However, our approach does not need to know this information since we can exploit the nature of underwater geographic routing by trying to locate routing paths.
We introduce our approach as follows: the input is a deployment area, which is then partitioned into a grid region by the transmission range of the attacker using a depth of 1 meter as the top plane since we do not want the surface nodes to hear our attacker. We assume the attacker is mobile, in this case an autonomous underwater vehicle (AUV), and can enter the deployment area from any side or the top of the network and can record its movement for future navigation. Once the search area has been input, the attacker will move to the first position in the grid and begin listening for packets to discover routing paths. The goal is elegant: the attacker will traverse across the plane it is on and try to find a single routing path. It will listen for data transmissions in each grid position on the current plane as it traverses across until it reaches the last position on the plane. If the attacker heard transmissions at only one location then it will move to this location, as it determines this is the best spot in the network to attack.

Given the routing nature, from some depth to the surface, discovering the plane with only one routing path implies you have found a bottleneck in the topology as all data must be transmitted through this area. However, if the attacker found more than one routing path, he can set these locations as the new search space bounds and begin searching in the next plane (below or above, dependent on its current location) to try and find a converging path. If the distance between the two furthest boundaries is greater than twice the attacker's transmission range then it should go up or down two planes since it is unlikely those two paths will converge in the next plane given their distance apart. If the attacker moves down and searches between the new bounds and does not find a converging path but finds that he stops hearing traffic at one of the boundaries, he will expand his search in that direction until he picks up the path again, correcting the bounds. If he does not correct this, the attacker could think he found a single path when really one path just moves farther away with depth. In the case that a single path cannot be found, the attacker will move to the location where it heard the most traffic when its energy has reached a threshold used for searching. In cases of multipath routing, when
no convergence can be found, the attacker will have to settle for the location with the most observed traffic. This opens up one direction for future work with distributed and collaborative attackers. We make use of existing energy requirements from a well-known AUV, known as the REMUS [92], which can operate for 8 hours at 5 knots or up to 20 hours at 3 knots. We also note that in each experiment, the network is assumed to have been operating before our attacker enters the deployment area.

![Figure 19: Improved Attack Model Results for Previous 5 Networks](image)

![Figure 20: Search Positions and Data Transmission Route](image)

55
4.4.1 Attack Performance

The results of our improved attack model on the previous 5 network topologies from Section 4.4 can be seen in Figure 19. This figure shows that the attacker was able to find an optimal position for all networks, increasing the attack performance over random placement and comes close to a 100% packet kill rate (except in network 1). Networks 2 through 5 all had convergence points while network 1 had two paths. Therefore, network 1 only allowed for roughly 50% effectiveness. We further summarize our improved attack model performance on these networks in Table 6, which shows the amount of time, in seconds, it took our attacker to find the best position and how many locations were checked before finding the best spot. We note that the time to find a location is a result of how fast the attacker could traverse the network (we use a speed of 2.57222 meters/second, taken from the REMUS AUV) combined with the listening time at each position, in these experiments the listening time is 60 seconds. In the case of network 2, the best position was the last position on the plane, while in network 3 the best position was at the first position of the plane and therefore the attacker had to go all the way back to reach this position.

As we observe from the search time, the energy consumption by searching is small (move time in Table 6), leaving the rest of the energy life for attacking the network. In Figure 20 we show the positions our attacker searched and the flow of data packets in the network for network 4. We note that the blue nodes are regular network nodes, the green nodes are surface nodes and the red nodes are the attacker positions. We do not show the sender on the right side of the network as it becomes a void zone where no path to the surface exists.

Additionally, we wanted to further test how setting \(d_{\text{spoof}}\) affects the performance of the attack. We ran our improved attack model again on each of the 5 networks and when the best location was found, the attacker started its attack phase. This time, instead of using DOR-TR as the spoofed depth, the attacker used DOR (which is the depth of the legitimate optimal receiver...
in the area). The results were the same as using a depth of DOR-TR as there were no nodes in the attackers transmission range that had better positions. For static networks, spoofing a depth of DOR is quite powerful, especially given that it mimics the depth of the legitimate receiver.

<table>
<thead>
<tr>
<th>Network #</th>
<th>Move Time (s)</th>
<th>Total Time (s)</th>
<th>Positions Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>388.769</td>
<td>588.769</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>233.26</td>
<td>383.26</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>544.27</td>
<td>724.27</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>466.52</td>
<td>646.52</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>311.01</td>
<td>491.01</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 6: Improved Attack Model Performance

4.4.2 Impact of Node Mobility

Until now, we have analyzed our attack performance on static networks. In this section we will analyze our improved attack on a mobile network using a kinematic model from [93] known as the Tidal Mobility Model, which captures the chaotic stirring in tidal areas. We approximate this mobility similar to [94] and is as follows:

\[
\begin{align*}
V_x &= k_1 \lambda \upsilon \sin(k_2 x) \cos(k_3 y) + k_1 \lambda \cos(2k_1 t) + k_4 \\
V_y &= -k_1 \lambda \upsilon \cos(k_2 x) \sin(k_3 y)
\end{align*}
\]  

where \(V_x\) is the speed in the \(x\) axis and \(V_y\) is the speed in the \(y\) axis. Additionally, \(k_1, k_2, k_3, \lambda,\) and \(\upsilon\) are variables closely related to environmental factors such as tides and bathymetry. These values change in different environments. As well, \(k_4\) and \(k_5\) are random variables. We assume \(k_1\) and \(k_2\) are random variables which are subject to normal distribution with \(\pi\) as the mean and \((0.1\pi)^2\) as the standard deviation; \(k_3\) is subject to the normal distribution with \(2\pi\) as the mean and \((0.2\pi)^2\) as the standard deviation; \(\lambda\) is subject to the normal distribution with \(1\) as the mean and \(0.01^2\) as the standard deviation; \(\upsilon\) is subject to the normal distribution with \(0.2\) as the mean and \((0.01)^2\) as the standard deviation; and \(k_4\) and \(k_5\) are subject to the
normal distribution with 0.1 as the mean and \((0.01)^2\) as the standard deviation. The effect of mobility in the 2D case is shown in Figure 21 where the red node is the starting location and the green node is the end location after 6000 seconds. This model was originally produced for 2D networks and therefore we adopt a 3D version of this model.

![Figure 21: One Example of Node Mobility over 6000 seconds](image)

We use network 4, the same network used in previous experiments. The simulation settings and parameters are the same except for the value of \(\delta\), which has increased to shorten holding times for mobile networks. We assume the senders are anchored to the bottom of the sea floor, otherwise, network connectivity is largely affected. For clarity, the settings are a 1000 meter \(\times\) 1000 meter \(\times\) 1000 meter area deployment area with 35 randomly deployed network nodes, 3 surface nodes, a transmission range of 200 meters, \(\delta = 25\), a send depth = 900 meters and a threshold = 10 meters. Given the mobility of the nodes, if a node reaches the surface it is randomly redeployed back into the network and if a node hits a side boundary it is placed on the opposite side boundary at the same depth location. This ensures 35 network nodes in the deployment area and helps to maintain network connectivity. We assume the network has been operating for some time before the attacker node enters the area. Once the attacker finds its optimal location, it will then launch its attack.

We gather attack performance over the course of 110 minutes. The results are displayed in Figure 22. The top figure shows the attacker efficiency over time, where anything lower than 100% means the attacker is not stopping all the messages it could have stopped, except when
the efficiency is at 0%, which means that the attacker is not hearing any messages in the area. The bottom figure shows the number of packets that were killed by the attacker and the number of packets received at the surface. We note that network connectivity is lost between the time period of minutes 5-8, 42-44, 48-49, 51-52, 82-84 and 87-91. As the figure shows, mobility impacts the attacker performance. Looking at the attacker efficiency, when it begins to slope down, such as between the time period of minutes 33-40, it shows that nodes are slowly moving out of the attackers transmission range and the attack is slowly worsening, until all forwarding nodes move out of the area and no more nodes in the attacker’s area are forwarding data or exist. The same applies for nodes moving into the attackers area, such as between the time period of minutes 50-75 when the attack is getting better and better each minute. In some cases network connectivity is slightly affected, such as between the time period of minutes 100-110. The attacker is killing roughly 50% of the packets being sent but no packets are being received at the surface, implying that the other 50% are lost due to connectivity issues along the way.

<table>
<thead>
<tr>
<th>Move Time (s)</th>
<th>Total Time (s)</th>
<th>Positions Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>4629.996</td>
<td>4959.996</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 7: Improved Attack Model Performance on a Mobile Network
We can observe from this figure based on the time periods that there appears to be two general routing paths that the nodes move between. The search time can be seen in Table 7. The search time is large because the attacker tried to find a convergent path but was affected by the mobility of nodes. Additionally, the position that was determined to be the best was a much earlier sampled location and therefore roughly 35% of the total time was the attacker moving back to that position. It is clear that for mobile networks, in order to obtain more effective performance the attack model needs to be tuned accordingly. One potential avenue is mobility prediction similar to what is used in localization techniques. We leave this notion for future work.

4.5 Summary

In this chapter we have studied the vulnerabilities of geographic routing protocols in UANs. We have proposed a preliminary spoofing based attack model for underwater geographic routing protocols, using DBR as a case study. We provided detailed simulation analysis on attack performance using various network topologies and studied the performance of different spoofed depths. Additionally, we introduced an improved attack model for static networks that locates the best position in a given topology to launch an attack while maintaining energy requirements. Our attack is shown to be powerful and minimal in terms of traces left behind, because no data needs to be manipulated or compromised. Our improved attack was further tested against mobile networks and performs well but could be improved by considering movement predictions.
Chapter 5

A Resilient Pressure Routing Scheme

Routing data in UANs is a primary concern, especially since sensor nodes can be deployed over large spatial areas and can be passively moving due to ocean currents. As discussed in the previous chapter, routing methods are often categorized into three types: reactive (or on-demand) routing, proactive routing and geographic routing. Traditional routing protocols that use on-demand or proactive routing have been shown to be costly and inefficient given the long-delay characteristics of the underwater acoustic channel and the dynamic nature of the environment [61]. Route discovery and maintenance leads to a higher collision probability in UANs and heavily impacts energy constraints due to the “power hungry” operations of acoustic communication. Geographic routing, which requires no state information (i.e. route discovery/maintenance), forwards data based on location information alone and is becoming a popular technique for use in deep water applications. A sub-type of geographic routing has emerged for UANs known as pressure routing.

Pressure routing was initially introduced in [87] and then later expanded upon in [90, 95, 96]. The idea is to exploit the nature of underwater networks in that data forwarding is often
in a *vertical* manner, from the sea floor to the surface. Each sensor node is equipped with an on-board hydraulic pressure gauge that can accurately estimate the depth of the node within an average error of under 1 meter [97]. Given that underwater acoustics is broadcast in nature, the basic concept behind pressure routing is that nodes can broadcast their data along with their current depth and neighboring nodes who are at a higher depth (physically) can then decide to forward the data onward.

Most work to date on routing in UANs has been about trying to discover effective routing methods, such as how to route data from $A$ to $B$ in the most optimal way. However, in recent studies [57,61–64] it has been shown that existing underwater routing methods are insecure and can easily be tampered with or attacked by a malicious adversary. Specifically, our work in the previous chapter [62] showed how existing geographic routing and pressure routing protocols are vulnerable to being attacked by an insider spoofing attack. It is shown that entire network traffic can be stopped at specific locations in which it can never reach the surface. This is accomplished by having an adversary act like a legitimate node with a fake depth. These existing vulnerabilities motivate the work presented in this chapter.

In this chapter, we propose **RPR**, a Resilient Pressure Routing protocol that aims to maintain routing in the presence of malicious adversaries, such as those in the previous chapter [62]. RPR utilizes cryptographic mechanisms, implicit acknowledgments (I-ACKs), geographic constraints, and randomization to achieve its resilient survival of packet delivery. With this in mind we note that there is a trade-off between performance and resilience. This is commonly seen in secure/resilient or fault-tolerant designs as this trade-off is inherent between these two characteristics [98].
5.1 Protocol Design

Our proposed protocol, RPR, is modeled after DBR [87] and follows a similar opportunistic forwarding strategy [99]. RPR utilizes cryptographic mechanisms, implicit acknowledgments, retransmissions, geographic constraints with a sliding window feature and randomization to achieve robust packet delivery service in the presence of attackers.

The flow of our protocol for both a sender and receiver can be seen in Figure 23. The protocol works as follows: upon deployment, the network enters a period known as the Discovery Phase in which it will set a threshold window as wide as possible. It will remain in this phase for a predetermined number of transmission rounds, set by the network operator, before dynamically adjusting this threshold window. This period is used to learn about potential neighboring nodes which can be used for threshold tuning. After the discovery phase, the
nodes enter the Selection Phase. In this phase, each node will dynamically calculate a threshold window at each hop based on neighbor information. Regardless of the phase that a node is in, the process for sending and receiving remains the same.

Before sending a packet, each node will create the packet header. This includes the sender ID, the forwarder’s ID, the packet sequence number, the node’s depth, a lower bound threshold, $T_{\text{min}}$, an upper bound threshold, $T_{\text{max}}$, and finally the lifetime of the packet. The sender, $S$, will then broadcast this packet out and enter a waiting phase. Any node in range of this broadcast will receive this packet and then check whether its depth, $\text{depth}[n_i]$, is $T_{\text{min}} \leq \text{depth}[n_i] \leq T_{\text{max}}$. If this constraint is met then this node is a candidate to forward the packet and checks a second constraint, which is that it has not yet heard this packet previously (i.e., it is not a duplicate) by comparing the sender ID and packet sequence number in the header. If any of these constraints fail, the receiver will drop this packet and not forward. If both constraints are met, the node will enter a holding time, in which it will wait a period of time before forwarding the packet. If during this time period, the node overhears this packet transmitted by another node that has a better depth, namely $\text{depth}[n_{i+1}] < \text{depth}[n_i]$, it will drop this packet. If the node does not overhear this packet transmitted at the end of its holding time, it will then enter the sender phase, generate a new packet header, broadcast the packet then enter the waiting phase.

Once this packet is forwarded, the original sender, $S$, which is in its waiting phase, will overhear this transmission and use this as an I-ACK that the packet was properly forwarded. At this point, the initial sender is done and can sleep until its next packet. If an I-ACK is not received, we have two cases to check. First, if this is the first time the node has sent or forwarded a packet and it is in the discovery phase, we assume this node is in a local void zone, meaning no forwarder exists to move this packet closer to the surface, and it should no longer participate in any future packet forwarding. If this is not the case when the I-ACK
is not received, we assume that channel conditions have caused the packet to become lost or corrupted and allow the sender, \( S \), to retransmit this packet. A retransmission counter is set before deployment to determine how many times a node is allowed to retransmit a packet before it is dropped by the sender.

We assume that a cryptographic infrastructure exists within the network to provide the following functions: 1) a node can check if an incoming packet is from a legitimate node; 2) the header is encrypted, but all the legitimate nodes in the area can decrypt the header; and 3) the payload is encrypted such that only the intended receiver (e.g., gateway nodes) can decrypt the payload. Many schemes can be designed to achieve these goals. In this work, we design a simple scheme and assume the underwater nodes can afford public-key algorithms since they have more computational power [100]. More advanced schemes (e.g., those based on group signature) can be designed in the future. In this simple scheme, each node has a unique ID, a pair of keys (public key \( PK_i \) and secret key \( SK_i \)), and a certificate for the key pair generated by a trusted party (TP). In addition, all the nodes share a network wide secret key (NSK), which will be used to encrypt information to be shared among multiple nodes. All the gateway nodes share a pair of keys (GPK, GSK) for receiving packets. The original sender encrypts the payload with GPK (the public key of the receiver). Every forwarder encrypts the packet header with NSK, and signs the packet with its public key. When a node receives a packet, it decrypts the header and checks if the packet is signed by a legitimate node. The scheme enforces that only nodes knowing the shared secret (NSK) can inspect a packet and that only the packets with a proper signature are accepted.

5.1.1 Packet Header

The packet header is encrypted with the NSK and consists of six fields: Sender ID, Forwarder ID, Packet Sequence Number, Depth, Thresholds and Packet Lifetime. “Sender ID” is
a unique identifier used to indicate original sender of the packet. “Forwarder ID” is a unique identifier used to indicate the current forwarder of the packet. “Packet Sequence Number” is a unique identifier assigned to the packet by the original source node. “Depth” is the depth information of the most recent forwarder, which is updated at each node in the network prior to the packet being transmitted. “Thresholds” contains the value of $T_{min}$ and $T_{max}$, which specify geographic constraints for forwarding. Finally, “Packet Lifetime” is the time that the packet has been alive since it was generated at the initial source node. This metric is calculated in the following way: when the packet is created or received by a node, it is at timestamp $t_0$, and when the packet is forwarded it is at timestamp $t_1$. Each node will subtract $t_0$ from $t_1$ then add this result to the value in the packet header and re-embed this information into the packet header then immediately broadcast the packet. This metric will be not be entirely accurate since the propagation time at each node can not be calculated without localization. However, in Section 5.2 we show this does not impact the performance of our approach.

### 5.1.2 Holding Time

Existing approaches for calculating the holding time of a packet in pressure routing protocols are based on linear equations and are deterministic [87, 90]. A malicious adversary can exploit knowledge of these linear and deterministic equations to observe network transmissions and even uncooperatively localize with the network without sending a single packet. One such attack was shown to work against DBR in [101]. For this reason, we propose a random based holding time for our approach. Random based approaches are shown to be effective countermeasures in routing protocols to limit insider attacks [66]. We generate a random holding time based on a uniform distribution between two parameters, $r_1$ and $r_2$. This approach will increase our protocol’s resilience to security attacks but comes with an increase in average end-to-end (e2e) delay. However, in order to improve the average e2e delay we make use of the
packet lifetime, $P_t$, embedded into the packet header. When each node calculates its holding time, they will also calculate what we refer to as, the average e2e delay ($AVG_{E2E}$). This is calculated by the following:

$$AVG_{E2E} = \frac{depth[z]}{V_0} + \sum_{n=0}^{n_{max}} (0, 1, \ldots n_{max}) \times N_{hops}$$

where $N_{hops}$ is defined as:

$$N_{hops} = \frac{depth[z]}{TR}$$

where $depth[z]$ is the depth of the deployment area in meters (we assume all nodes know this), $V_0$ is the propagation speed of sound in water in meters, $n_{max}$ is set to $r_2$ and $TR$ is the transmission range. Each node will then use the result of this calculation to check if the packet lifetime in the packet header is greater than or equal to the average end-to-end delay calculation, namely, if $P_t \geq AVG_{E2E}$. If this constraint returns as true, the node assumes that the packet has been taking too long to reach the surface. In this case, we set the node’s holding time to one of two cases which are as follows:

$$H_n = \begin{cases} \frac{H_0}{2}, & \text{if } depth[n] > 2TR. \\ \frac{H_0}{3}, & \text{otherwise.} \end{cases}$$

where $H_n$ is the holding time of the node in question and $depth[n]$ is the depth of the same node. This process expedites packet forwarding to reduce the average end-to-end delay of packet delivery to the surface. If this constraint returns false, nodes will function normally and not change their holding times. This approach will help to maintain an acceptable average e2e delay.
5.1.3 Sliding Window Thresholding

The use of geographic constraints is beneficial to allow for dynamic candidate selection. This helps to limit the potential ranges a malicious attacker could use to spoof and reduces the chances of a malicious attacker gaining control of the traffic flow [66]. Existing pressure routing protocols make use of geographic constraints but implicitly. In DBR, a lower bound threshold is specified globally, before deployment, that enforces the depth difference a node must be from a sender in order to be considered a potential candidate. The downfall of this lower bounding is that once it is set, it cannot be changed and could hurt network connectivity if the network is not deployed properly or has nodes moving passively or actively. Further, as pointed out in [64], a malicious adversary can spoof a wide range of depths to inhibit traffic flow. In HydroCast, potential forwarders use their depth information to evaluate their priority level for being a candidate. To put it simply, a node closer to the destination has a higher priority of being a candidate. In this method, receivers have complete control of deciding to forward or not, which implies a malicious adversary does not need to meet any specific requirement to gain access to the network flow.

In order to introduce dynamic candidate selection we propose the use of sliding window based thresholding. This involves the use of a lower and upper bound threshold, $T_{\text{min}}$ and $T_{\text{max}}$, that can be set by each sender or forwarder in the network by embedding this information into the packet header. These thresholds are initially set globally by the network operator before deployment and can then be changed by each sender or forwarder in the network. We enforce three rules when it comes to setting thresholds:

- These thresholds must BOTH be defined, they can not be left empty;

- $T_{\text{min}}$ may be less than 0 to help avoid void zones but should be at least 0 if no void zone is present and less than that of $T_{\text{max}}$; and
\( T_{\text{max}} \) must be less than or equal to the transmission range of the node.

These rules help to minimize the potential attack approaches from a malicious adversary. For example, a node should not be able to set an upper bound outside of its transmission range because this is a contradiction. Further, these rules help to minimize a malicious adversary trying to exploit this thresholding mechanism because they must provide valid thresholds. If a node observes a packet that does not meet these requirements, it will simply ignore it. Our proposed approach makes use of two phases in order to set the sliding window threshold: a discovery phase and a selection phase.

5.1.3.1 Discovery Phase

In our proposed approach, the network is deployed with global thresholds set to the following: \( T_{\text{min}} = 0 \) and \( T_{\text{max}} = TR \), where \( TR \) is the transmission range. The network will then send packets using this threshold window for a predefined number of transmission rounds, known as \( \text{Tran}_{\text{Disc}} \). We call this period the Discovery Phase. During this phase, the network allows all nodes in the global threshold window a chance to forward packets. Each time a node forwards, all neighbors will overhear this broadcast and store the depth information into a Neighbor Table. After each forward by a node, it will update a counter. When this counter reaches the value of \( \text{Tran}_{\text{Disc}} \) the discovery phase is over and the node will enter the Selection Phase. It should be noted that \( \text{Tran}_{\text{Disc}} \) is user defined and should be set based on the density of the network. Denser networks should set this value higher in order to provide more neighbor history during the discovery phase. This will allow for more threshold windows to use during the selection phase.
5.1.3.2 Selection Phase

During the discovery phase, each node in the network was recording the depth information of neighbors who forwarded into the neighbor table. Once the selection phase has started, each node will randomly choose two depths from this neighbor table for $T_{\text{min}}$ and $T_{\text{max}}$. This selection will be done each time a node sends or forwards a packet. If any new nodes are discovered during this process, they will be added to the neighbor table and be included in the selection process. The benefits of this process is randomized candidate selection in ranges which nodes are known to exist. In this way, a window is not set in which no node exists.

5.1.4 Retransmissions

Since the underwater channel is not a stable environment, packet loss is highly dynamic. For this reason, we have added a retransmission mechanism into our proposed protocol to increase packet delivery. In existing protocols, such as DBR, when a packet is lost there is no recovery mechanism and therefore performance in unstable environments will be low. Further, to reduce unnecessary transmissions in the case of elongated poor channel conditions or unexpected void zones, we make use of a lifetime retransmission counter and timer. This parameter is user defined and specifies how many packets a node can drop before it should sleep for some period of time or indefinitely. This will help to reduce performance impacts by nodes which are in bad deployment areas.

5.1.5 Discussion

The design of RPR provides it with various mechanisms to remain secure and resilient. Only nodes that know the network wide secret key (NSK) and a legitimate ID (and associated keys and certificates) are allowed to participate in the forwarding process. This prevents attacks
from an adversary that simply enters the deployment area. The sliding window thresholding prevents nodes outside the current threshold from forwarding. If a node outside the current threshold is seen forwarding, all nodes disregard this transmission because it violates protocol rules. The thresholds also help to detect the cases when a compromised node fakes its depth information. For example, a compromised node could be seen trying to forward packets from two different depths. This action will cause all nodes to ignore this node’s actions in the future. Further, if the nodes are equipped with modems that can report Angle of Arrival (AOA) of the incoming signals, they can roughly check, with AOA, if the forwarding nodes are located at the reported depth.

5.2 Experimental Results

We have implemented our proposed protocol in a Java based simulation tool known as GeoSim. This simulation tool was designed to test geographic routing protocols and contains an underwater channel implementation. We refer readers to [91] for more information on simulation architecture, implementation and fidelity details.

For our simulation results we compared our proposed approach, RPR, against that of DBR and Random DBR. Random DBR is the DBR protocol but with pure random hold times such that we can showcase how DBR performs with some minor randomization. It is not an existing protocol. Our simulation settings are as follows: six randomly generated networks with densities of 50, 100, 200, 300, 400 and 500 nodes respectively in a 2000 meter x 2000 meter x 2000 meter deployment area with each node having a transmission range of 400 meters. All nodes in the network are static and do not move. We specify a send depth of 1800 meters. This means that only nodes at or greater than this depth will generate packets, any node less than this depth will only forward packets. We run each simulation 15 times at a length of 600 seconds for an average result. In DBR we set the $\delta$ parameter in the holding time calculation to
12 which corresponds a holding time to be between 0 and 13 seconds. Therefore, in RPR and Random DBR, we set \( r_1 \) to 0 and \( r_2 \) to 13, which allows for a random hold time between 0 and 13, to ensure accurate comparison between the protocols. In RPR, the discovery phase is set to last for 5 transmission rounds and each node is allowed 0 re-transmissions before dropping the packet.

We evaluate these protocols using 4 metrics: packet delivery ratio, average end-to-end delay, average number of transmissions and finally attack performance of a malicious adversary. We define these metrics as follows: Packet Delivery Ratio (PDR) is defined as the ratio of the number of unique or distinct packets received successfully at a sink node to the total number of unique or distinct packets generated. Average End-to-End Delay is defined as the average time taken by a packet to travel from a source node to any of the sink nodes. Average Number of Transmissions Per Unique Packet is defined as the total number of transmissions in the network divided by the number of unique packets sent in the network.

![Figure 24: Packet Delivery Ratio](Image)

In Figure 24 we compare the PDR of each approach. For larger networks all three approaches perform equally at 100%. For smaller networks, such as a density of 50 nodes, RPR performs best and increases the PDR by almost 3 times that of DBR. The reason for this is because the dynamic threshold window allows for more routing paths to be discovered. In Figure
In Figure 25 we compare the average end-to-end delay of a packet. Again, RPR performs best with a significant decrease in time delay.

In Figure 26 we evaluate the average number of transmissions per unique packet. RPR has a significantly higher number of transmissions than the other two approaches. However, this is not necessarily a negative impact. The reason for the increase in transmissions is because more nodes are participating in forwarding the packet to the surface. This is an added increase in reliability and resilience because an implicit load balancing is being performed and the packet is traveling through dynamic paths to the surface. We show this exact feature in Figure 27. In this figure, we show a sample network path (transposed to a 2D graph) of DBR and RPR in the
50 node network. DBR nodes are in blue, RPR nodes are in pink and shared nodes are pink in the middle with a blue outline. The nodes shown here are every node that forwarded one unique packet during the simulation. As we can see, in plain DBR, there exists only one path to the surface. However, in RPR, 2 paths exist, which increases the probability that the packet will reach the surface and the number of transmissions in the network. Network topology plays a large role in the performance of each simulation and the attacker.

![Figure 27: Branching Benefits](image)

### 5.2.1 Attack Model

The attack model proposed is a malicious attacker who can move and compromise network nodes [66]. Since the network traffic is encrypted and signed, the attacker must gain access to an authenticated node in order to launch any attack. In this work, we assume the attacker only compromises one node in the network. Once the attacker has compromised a node, it can then use this node’s ID (and keys) to fake or spoof the node’s depth information to gain control of the network traffic. This will inhibit packets from being forwarded to the surface. An example spoofing attack, which we used, is shown in [64]. The attacker will passively listen to network transmissions between two nodes and obtain their depth information. The attacker can then calculate an appropriate depth to spoof and during the next transmission round, will
embed this depth into the legitimate packet and forward it out. If done correctly, all nodes who hear this transmission will drop the packet they hold and the network traffic can be blocked completely. In the future work, we will investigate the case of multiple compromised nodes.

For our experiment, we randomly select a node in the network on a routing path to become compromised. The simulation settings are the same as the previous experiments. The results can be seen in Figure 28. It can be seen that our proposed approach is able to reduce the effects of the malicious attacker by at least 80% in most cases. In the networks with 50 and 100 nodes, the attacker is able to do more significant damage because the number of network paths is limited which reduces the chances of breaking away from the compromised node. For example, in both the networks there exist a few locations where only one node exists to forward the packet onward and dynamic candidate selection fails because of this reason. In such cases the attacker is able to do significant damage.

![Figure 28: Attack Performance](image)

**5.3 Summary**

In this chapter, we have introduced a resilient pressure routing protocol that uses cryptographic mechanisms, implicit acknowledgments, geographic constraints and randomization to
achieve packet delivery in the presence of malicious adversaries. Further, the simulation results presented show that the protocol also maintains good performance when compared with existing pressure routing protocols by reducing the trade-off that often comes with resilient protocol designs. In some cases, the performance of our proposed approach is better than that of existing approaches.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

Underwater Acoustic Networks (UANs) are an emerging technology with capabilities to open our water world to increased human exploration and protection. Our planet is covered by roughly 71% of water and remains vastly unexplored. As humans we wish to explore the hidden depths of oceans to discover new organisms, increase off-shore structures for energy and resource harvesting and monitor remote locations for natural and anthropogenic climate impacts. UANs offer the potential to increase aquatic applications by being able to cover large bodies of water in a distributed fashion.

Significant progress has been achieved by addressing design issues of UANs, such as system integration, communication techniques and networking protocols. However, focus on providing security in these networks has been limited. Malicious adversaries may wish to tamper with these systems by attaching unauthorized nodes or disrupt a subset or potentially all functions of the network. This knowledge has fueled the motivation for the work in this dissertation.
In this dissertation, we have focused on Denial-of-Service (DoS) attacks and countermeasures for underwater acoustic networks and covered three major research thrusts, namely: 1) physical layer vulnerability; 2) network layer vulnerability; and 3) countermeasures. Our work is summarized as follows.

- **Physical Layer Jamming Attacks.** We investigated the vulnerabilities of the physical layer to denial-of-service jamming attacks. Jamming attacks were launched on UANs in an experimental setting by using a developed jamming system and signal. Attacks were tested against three commercially available acoustic modems. Additionally, weaknesses in the signal structure of each acoustic modem were studied through observation of how the network responded to jamming attacks. To the best of our knowledge, this was the first work in literature to study jamming effects in real-world settings on commercially available acoustic modems.

- **Network Layer Spoofing Attacks.** We performed a vulnerability analysis of the network layer in UANs with specific focus on geographic based routing protocols. An attack model was developed to mimic real protocol behavior and demonstrated the impact of location based spoofing attacks. The attack model was improved by a proposed placement algorithm that seeks to discover bottlenecks in network topologies. Insecure heuristics used in geographic based routing protocol designs were identified. To the best of our knowledge, this was the first work in literature to study spoofing attacks on underwater networks and underwater geographic routing.

- **Resilient Routing Countermeasures.** In order to address the concerns of denial-of-service attacks on both the physical and network layers of UANs, we proposed a Resilient Pressure Routing scheme, known as RPR. This protocol maintained routing services in the presence of malicious adversaries. RPR utilizes cryptographic mechanisms,
implicit acknowledgments (I-ACKS), geographic constraints and randomization to achieve its resilient packet delivery.

In addition to the work presented in this dissertation, we have completed work on other important areas in underwater networks, that are beyond the scope of this dissertation. These works include improving underwater network simulation methods and tools, designing and building experimental hardware systems for large-scale testbed deployments, time synchronization and localization, medium access control (MAC) protocol development and application specific network protocol development [23, 73, 102–109].

### 6.2 Future Work

The work completed in this dissertation provides an avenue for various future works. Two main topics fall into this future work, namely, studying the impact of collaborative attackers and developing more countermeasures for DoS attacks.

- **Collaborative Attacks.** The work in this dissertation covered DoS attacks that involved the use of a single attacker. The effects of multiple attackers that collaborate to perform more sophisticated attacks is an interesting topic to pursue. Additionally, multiple attackers expands upon the possible avenues for attacks. For instance, a malicious adversary could observe a network transmission and then use gateway buoys, that are equipped with RF radios, to send a malicious signal further down the network, essentially beating the legitimate transmission to the destination. This problem should provide interesting insight into future security concerns. Future work for better attack performance of these collaborative attackers could also be the development of a technique for more discrete sending times to mimic real transmissions or making use of mobility prediction in the case of mobile nodes.
- **Countermeasures.** Future work on countermeasures for underwater security threats is also a direction of interest. As practical systems and application specific network protocols are developed, new countermeasures for these systems will be required. Mechanisms such as spatial retreat, to avoid or run from attackers, should be investigated. Additionally, coding strategies to produce redundant data in the presence of interference is another avenue for network defense.

- **Additional Security Threats.** The work in this dissertation has provided insight into other areas in underwater communication and networking that have potential to be exploited. Examples of such areas are time synchronization and localization. Both of these services are critical to the success of underwater networks. In underwater networks, many applications rely heavily on time synchronization and localization to provide features such as reliable data transfer and node positioning. A malicious attacker could pose as a severe threat to such networks if these issues are not addressed.
Bibliography


