Design of Underwater Mobile Sensor Networks
for Real-time Aquatic Applications

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Seongwon Han

2014
Design of Underwater Mobile Sensor Networks for Real-time Aquatic Applications

by

Seongwon Han

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2014

Professor Mario Gerla, Chair

Underwater sensor networking is generally regarded as an emerging technology to conduct oceanic exploration and research in an automated and effective manner. As underwater operations become more sophisticated, there is an increasing demand for real-time aquatic applications such as real-time video streaming. However, real-time video streaming requires high bandwidth as well as low latency. Amongst the resources, bandwidth is the most critical limitation. To help overcome this obstacle, we first propose an innovative MAC protocol called Multi-session FAMA (M-FAMA). M-FAMA leverages passively-acquired local information (i.e., neighboring nodes’ propagation delay maps and expected transmission schedules) to launch multiple simultaneous sessions. M-FAMA’s greedy behavior is controlled by a Bandwidth Balancing algorithm that guarantees max-min fairness across multiple contending sources. In addition to this, we propose a hybrid solution that combines acoustic and optical communications. Optics provides good quality real-time video. Acoustic maintains a “thin” channel for network topology and transmission control, and for still frame video delivery when the optical channel fails. In particular, we enable optical communications by acoustic-
assisted alignment and use acoustic communications as a back up when the optical signal is interrupted. The main contribution is to enable reliable, real-time video streaming without underwater optical cables. Another important contribution is the smooth transition between the acoustic and optical video delivery mode, using popular image processing algorithms to compress the video before transmitting it on the acoustic channel.
The dissertation of Seongwon Han is approved.

Gregory J. Pottie

Jack W. Carlyle

D. Stott Parker

Mario Gerla, Committee Chair

University of California, Los Angeles
2014
To my parents

and to my brother
# TABLE OF CONTENTS

1 Introduction ......................................................... 1

   1.1 An Overview on Underwater Sensor Network ................. 2
       1.1.1 Mobile Underwater Networks and Resource Constraints ... 2
       1.1.2 Review of Underwater MAC Protocols ...................... 4

   1.2 Underwater Optical-Acoustic Hybrid Network ............... 4

2 M-FAMA: A Multi-session MAC Protocol for Reliable Underwater Acoustic Streams .................................................. 6

   2.1 Backgrounds .................................................... 6

   2.2 M-FAMA: Motivations and Basic Principles .................... 9

   2.3 Towards Enabling Multiple Transmission Sessions .......... 12

   2.4 Multi-session FAMA design .................................... 16
       2.4.1 Delay Map Management and Guard Time .................... 17
       2.4.2 Delay-map Assisted Packet Scheduling .................... 19
       2.4.3 Schedule Recovery ........................................... 21
       2.4.4 Enabling Multiple Sessions ................................. 22
       2.4.5 Backoff, Recovery, and Maintenance ...................... 26
       2.4.6 Bandwidth balancing ........................................ 27

   2.5 Simulation & Evaluation ....................................... 29
       2.5.1 Simulation Setup ............................................. 29
       2.5.2 Simulation Results .......................................... 32

   2.6 Related Work ................................................ 42
3 Evaluation of Underwater Optical-Acoustic Hybrid Network .......... 45
   3.1 Backgrounds .................................................. 45
   3.2 Acoustic Communications ................................... 46
      3.2.1 Overview of Acoustic Communications ................. 46
      3.2.2 Using Acoustic Communications in Networking ........... 47
      3.2.3 Acoustic Signal Attenuation .......................... 48
   3.3 Optical Communications .................................... 49
      3.3.1 Overview of Optical Communications .................... 49
      3.3.2 Optical Signal Attenuation ............................ 50
      3.3.3 Using Optical Communications in Networking ............. 51
   3.4 Acoustic vs. Optical Discussion ............................ 52
   3.5 Hybrid Solution ............................................. 55
      3.5.1 Objectives of Hybrid Solution .......................... 55
      3.5.2 Acoustic Source Localization ........................... 56
      3.5.3 Hybrid Transmission ................................... 59
   3.6 Simulation .................................................... 60
      3.6.1 Simulation Setup ....................................... 60
      3.6.2 Simulation Results .................................... 61
   3.7 Discussion ................................................... 63

4 Real-time Video Streaming from Mobile Underwater Sensors ............. 64
   4.1 The Hybrid Solution ......................................... 64
      4.1.1 Design .................................................. 65
      4.1.2 Scenarios for The Hybrid Solution ...................... 66
   4.2 Image Processing and Video Compression ....................... 69
4.2.1 Simple Image Processing Techniques .......................... 70
4.2.2 Sobel Image Processing ........................................... 72
4.2.3 Gaussian Blurring .................................................... 75
4.2.4 Canny Edge Detection ............................................. 78
4.2.5 Other Methods ....................................................... 81
4.3 Evaluation ............................................................... 82
  4.3.1 Implementation Details ......................................... 82
  4.3.2 Data Rate and Latency .......................................... 83

5 Conclusions .............................................................. 87

References ................................................................. 89
LIST OF FIGURES

2.1 SEA-Swarm Architecture ........................................... 7
2.2 Topology ................................................................. 10
2.3 Slotted FAMA ............................................................. 10
2.4 Multi-session FAMA ..................................................... 10
2.5 Enabling Multiple Transmission Sessions .......................... 12
2.6 Star Topology (4-Senders 1-Sink) .................................. 14
2.7 1-Sender 4-Sinks: throughput with fixed data size (512\textit{bytes}) 15
2.8 4-Senders 1-Sink: throughput with fixed data size (512\textit{bytes}) 15
2.9 Node C’s delay map update after overhearing node B’s RTS .... 19
2.10 An example of a concurrent transmission schedule .............. 21
2.11 M-FAMA \textit{conservative} ............................................... 26
2.12 M-FAMA \textit{aggressive} .................................................. 26
2.13 Simulation Topologies .................................................. 30
2.14 4Senders-1Sink: throughput as a function of offered load with tx range of 750m ......................................................... 33
2.15 4Senders-1Sink: throughput as a function of offered load with tx range of 1.5km ......................................................... 33
2.16 4Senders-1Sink: M-FAMA’s throughput convergence under homogeneous traffic .............................................. 34
2.17 4Senders-1Sink: M-FAMA’s throughput convergence under heterogeneous traffic .......................................... 34
2.18 4Senders-1Sink: Bandwidth Balancing .............................. 35
2.19 1Sender-4Sinks: throughput as a function of offered load with tx range of 750m ................................................................. 36
2.20 1Sender-4Sinks: throughput as a function of offered load with tx range of 1.5km) ................................................................. 36
2.21 Sea swarm (tree): throughput as a function of offered load .......... 37
2.22 Random with MCM (0.3m/s): throughput as a function of offered load . 38
2.23 Throughput amongst different MCM mobility speed ..................... 38
2.24 Jain’s Fairness Index for the five protocols ................................. 40
2.25 Guard Time sensitivity to a MCM mobility speed (0.3m/s) .......... 41
2.26 Energy Consumption in the random scenario with fixed data size (128 bytes) 41

3.1 Absorption Coefficient Versus Frequency ................................. 47
3.2 Inclination angle \( \beta \) and transmitter light-beam diverge angle \( \theta \) ..... 51
3.3 Optical Intensity over Distance in Different Water Types ............... 52
3.4 Acoustic and Optical Modem Bitrates ..................................... 53
3.5 Field of View of Optical Transmissions .................................. 54
3.6 Example of Hybrid Solution .................................................. 55
3.7 Time Difference of Arrival .................................................... 57
3.8 Mobile Source to Sink Setup .................................................. 60
3.9 Throughput as a function of offered load ................................... 62
3.10 Energy Consumption with varying data size ............................. 62

4.1 Optical/Acoustic alignment between AUVs ............................... 65
4.2 Bottom video exploration ..................................................... 67
4.3 Shallow water inter-submarine video communications .................. 68
4.4 Real-time video streaming via Acoustic/Optical .......................... 70
4.5 Image in grayscale .................................................. 72
4.6 Sobel without the adaptive algorithm .............................. 76
4.7 Sobel with the adaptive algorithm .................................. 76
4.8 Gaussian and Sobel without the adaptive algorithm .............. 78
4.9 Gaussian and Sobel with the adaptive algorithm ................. 78
4.10 Canny edge detection ................................................. 79
4.11 Results of image processing on full-size images(2592x1944) .......... 81
4.12 AquaSeNT OFDM modems .......................................... 82
4.13 Average file sizes for different image resolutions .................. 83
4.14 Average time interval for video frame delivery (128x96) ............. 84
# List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Existing Acoustic Modems</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>Attenuation Coefficient of Different Water Conditions</td>
<td>51</td>
</tr>
<tr>
<td>3.3</td>
<td>Acoustic vs. Optical</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Horizontal Sobel 5x5 mask [40]</td>
<td>73</td>
</tr>
<tr>
<td>4.2</td>
<td>Vertical Sobel 5x5 mask [40]</td>
<td>73</td>
</tr>
<tr>
<td>4.3</td>
<td>5 x 5 Gaussian matrix [4]</td>
<td>77</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I am indebted to many people for help with this dissertation. First and foremost I would like to thank my advisor, Professor Mario Gerla, for the continued patience, encouragement, and support. I have learned so much under his guidance on both academia and life in general, and am forever grateful for his dedication. I would like to thank the members of my dissertation committee, Professor D. Stott Parker, Professor Jack W. Carlyle, and Professor Gregory J. Pottie, for their insightful comments, feedback, and suggestions on this work. I must also thank all past and present members and visiting scholars of the Network Research Lab at University of California, Los Angeles for providing me with useful feedback, discussions, and insights. I don’t plan on ever forgetting the precious memories I have of this place. Furthermore, a number of graduate students at UCLA made valuable contributions to this work. I would like to thank everyone who has collaborated with me for all my research projects including Youngtae Noh, Sungwon Yang and Jihyoung Kim. My research has also greatly benefitted by working with many others including Richard Liang, Yung-Ju Cheng and Roy Chen. Last but not least, I would like to thank my parents and my brother for their unconditional love and support. Without all of them above, this work would not have been possible. I thank all of you from the bottom of my heart.
VITA

Education

2009 M.S., in Computer Science,
University of California, Los Angeles, USA

2006 B.S., in Computer Science,
Ajou University, Republic of Korea

PUBLICATIONS

CONFERENCE


JOURNAL


POSTER


CHAPTER 1

Introduction

Although ocean covers more than 70 percent of the Earth surface, a vast majority of it, about 95 percent, remains unexplored according to the National Oceanic and Atmospheric Administration in the United States. Since the ocean can be thousands of feet deep and is difficult for human divers to explore, researchers have turned to underwater sensor networks to gather information in an efficient and automated manner. While traditional sensors provide tabular data (i.e., salinity, temperature, and pressure), recently the need for still images and even real-time video streaming has emerged in the research community. Autonomous Underwater Vehicles (AUVs) have the ability to meet the video demands posed by applications such as ocean bottom monitoring, oil spill detection and search for minerals. If there was a way to provide real-time video streaming from AUVs to support ships, the above mentioned tasks could be carried out much more efficiently in interactive mode.

Mobile underwater networking is a developing technology for monitoring and exploring the Earth’s oceans. For effective underwater exploration, multimedia communications such as sonar images and low resolution videos are becoming increasingly important. Unlike terrestrial RF communication, underwater networks rely on acoustic waves as a means of communication. Unfortunately, acoustic waves incur long propagation delays that typically lead to low throughput especially in protocols that require receiver feedback such as multimedia stream delivery. On the positive side, the long propagation delay permits multiple packets to be “pipelined” concurrently in the underwater channel, improving the overall throughput and enabling applications that require
sustained bandwidth. To enable session multiplexing and pipelining, we propose the Multi-session MAC Protocol for Reliable Underwater Acoustic Streams (M-FAMA) shown in Chapter 2.

1.1 An Overview on Underwater Sensor Network

We review the mobile underwater networks, types of mobile sensors, their constraints (e.g., communication characteristics and energy consumption) and then thoroughly examine underwater MAC protocols.

1.1.1 Mobile Underwater Networks and Resource Constraints

The design of oceanic networks for monitoring and scientific exploratory purposes can be largely classified into two categories: (1) static sensors tethered at the seabed or buoys on the ocean surface with external power sources (e.g., NEPTUNE [45]), and (2) mobile sensors such as AUVs and underwater floats (e.g., SeaWeb [72], ARGO [38], UCSD Drogues [47]). Static sensors are typically used for long term, pre-planned missions such as seismic activity monitoring, whereas battery-powered mobile sensors are used for short-term missions such as oil and chemical spill monitoring. The key benefit of mobile sensing is that mobility permits more flexible underwater exploration with wide area coverage at reasonable cost. AUVs can follow planned trajectories such as a sequence of tracklines, waypoints, and depth excursions [15], while floats have restricted mobility as they move along with water current (e.g., ARGO [38], UCSD Drogues [47]). Given that the cost effective coverage is one of the primary concerns of mobile sensors, such networks must employ low-cost, energy-efficient mobile nodes, and thus, resource constraints must be carefully examined [74].
1.1.1.1 Mobile sensor types

The most common AUV configuration is a torpedo-like vehicle (e.g., REMUS, IVER2) with a streamlined body with propeller and control surfaces at the stern [15]. The speed of such AUVs in the range of 0.5 to 5m/s, and most vehicles operate at around 1.5m/s. Another configuration is a glider (e.g., Seagliders [74]) that uses small changes in its buoyancy in conjunction with wings to make up-and-down, sawtooth-like movements. Although gliders have such restricted mobility patterns, due to the energy efficiency, they can provide data collection on temporal and spatial scales that would be costly if traditional shipboard methods are used. Unlike AUVs, underwater floats like UCSD Drouges and ARGO [38] mainly use a buoyancy controller for depth adjustment and passively move along with the water current.

1.1.1.2 Resource constraints of mobile sensors

We review the resource constraints of mobile underwater sensors, namely acoustic communications and energy consumption. Communications in the underwater acoustic channel are with two innate characteristics: low bandwidth and large propagation delay. The available bandwidth of the acoustic channel is limited and strongly depends on both range and frequency. As surveyed by Kilfoyle et. al. [51], existing systems have highly variable link capacity, and the attainable range and rate product can hardly exceed 40km-kbps. The signal propagation speed in the acoustic channel is $1.5 \times 10^3$ m/sec, which is five orders of magnitude lower than radio propagation speed $3 \times 10^8$ m/sec in the air. This huge propagation delay has great impact on network protocol design. Also, it is important to note that underwater acoustic modems consume significant amount of energy when compared with terrestrial radios; e.g., WHOI Micromodem-1 has the active/receive state with power consumption of 158mW and the transmission state with full power consumption up to 48W [35].
1.1.2 Review of Underwater MAC Protocols

In multi-hop wireless networks, it is important to efficiently utilize limited network resources and to provide fair access for competing data flows. It has been proven that CSMA provides reasonable performance and fairness [16]. Since CSMA does not require strict scheduling, it can support node mobility, which is also a major challenge in mobile underwater networks. However, the handshaking mechanism of CSMA leads to a severely degraded system throughput due to the presence of long propagation delay of acoustic signals in mobile underwater networks, which is a well-known problem. Moreover, carrier sensing may fail to detect an ongoing transmission due to the propagation delay, which impairs the performance of CSMA protocols [69]. A more detailed review can be found in Section 2.6.

1.2 Underwater Optical-Acoustic Hybrid Network

Traditional underwater modems have used acoustic communications to transmit data through the water using acoustic waves. Acoustic waves travel underwater over distances of several kilometers and do not require direct line-of-sight between the sender and the receiver. However, acoustic communications have disadvantages, namely: the long propagation delay of sound waves compared to electromagnetic or light waves, limited bandwidth and the ease of detection and eavesdropping by the enemy (a critical issue in tactical operations).

Optical communications have received significant attention recently and are the subject of ongoing research [28, 29, 32, 66, 30]. Farr et al. [32] presented insightful optical communication scenarios and demonstrated video monitoring up to 15 meters. Doniec et al. [29] developed AquaOptical II, a bidirectional underwater optical communication system capable of transmitting a few Mbps with the effective range over 15 meters. They further elaborated the previous system mainly focusing on video streaming and successfully transmitted over a range of 25 meters [30]. Generally, optical communica-
tions have a higher bandwidth capacity and consume much less energy than acoustics. The speed of light through water is faster than the speed of sound, which results in a lower latency in optical communications. However, light waves have a larger attenuation. At most, optical modems can transmit at distances of about 100 meters [32], in excellent water conditions. Due to the nature of optical communications, a clear line-of-sight is required between the sender and receiver, although there are efforts to overcome this obstacle [11, 10]. This requires for the optical modems to be aligned in order to provide reliable data transmission. It has been reported that optical modems can have a field of view of up to 120 degrees. However, the transmission distance decreases as the transmission angle increases [42].

In Chapter 3 and 4, we propose a hybrid solution for real-time video streaming from sensors to a monitoring center (e.g., surface ship). The primary objective is to realize real-time underwater video streaming over both the acoustic and optical modes (where the acoustic mode is the backup mode). However, in case of optical channel failure, the acoustic channel has low bandwidth and long propagation delay, and is not quite adequate to take over the support of the real time application (e.g. remote search of the ocean floor). A possible solution is to dramatically reduce the size and rate of video frames, so that they can fit the acoustic channel and yet allow application interactivity. To this end, we propose to use image processing techniques to compress the video frames and reduce the size to fit the acoustic channel. We use the acoustic modems to transmit ACKs and other control messages to prevent collisions between AUVs.
CHAPTER 2

M-FAMA: A Multi-session MAC Protocol for Reliable Underwater Acoustic Streams

2.1 Backgrounds

Although oceans cover two-thirds of the Earth’s surface, human exploration and understanding of these frontiers has historically been limited by technical barriers. Recent work suggests that Underwater Acoustic Sensor Networks (UW-ASNs) are effective tools for exploring and observing the ocean [6, 52, 84]. An example is the SEA Swarm (Sensor Equipped Aquatic Swarm) architecture, where a large number of sensors are deployed as a group that moves with the water current [56, 88] (see Figure 2.1). A swarm of sensor nodes is escorted by surface relay buoys, which are equipped with acoustic, RF, and satellite interfaces. Each sensor monitors local underwater activities and acoustically reports critical multimedia data to any one of the surface stations over multiple hops if necessary. The data are then relayed via radio channels from the buoys to a central monitoring station.

Swarm mobility presents new technical hurdles—especially in the context of an acoustic communications channel [20]. Despite technological advances in acoustic communications, several challenges remain, including: limited bandwidth, long propagation delay (1.5km/s: five orders of magnitude slower than radio frequency) [83], relatively high transmission energy cost (with typical reception vs. transmission energy ratio of 1:125 [5]). Many underwater MAC protocols have recently been proposed to address such limitations. Most of these protocols are based on Carrier Sense
Multiple Access (CSMA) or Code Division Multiple Access (CDMA). Only a few use Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA). TDMA necessitates a network-wide time consensus, which results in a large number of control packet exchanges and requires a lengthy synchronization process. This implies that all nodes must remain synchronized, regardless of node failures or node movements, in order to maintain reliable transmission schedules. More importantly, TDMA-based methods are not suitable for resource-constrained underwater mobile sensor networks, because nodes must periodically perform expensive scheduling operations [67, 79, 70, 46, 54]. Likewise, FDMA is an inherently inefficient protocol for UW-ASNs; only a subset of the available frequency/bandwidth can be used due to the prevalent fading in underwater environments [73, 13, 12].

Since CSMA-based protocols easily support the required network dynamics (i.e., node mobility, failure, joining, and leaving), interest in CSMA-based protocols has increased recently over CDMA [54]. Long propagation delays cause significant performance degradation in protocols using a ready-to-send/clear-to-send (RTS/CTS) mechanism. Newer CSMA-based protocols attempt to address this challenge, by enabling channel reuse through concurrent transmission sessions [39, 85, 22, 24, 65]. Here, channel reuse is typically categorized as either temporal or spatial reuse; i.e., temporal
reuse occurs when multiple outstanding packets can be scheduled without collisions (either from single sender or multiple senders), and spatial reuse occurs when multiple neighboring (exposed) terminals transmit at the same time. However, to the best of our knowledge, none of the existing protocols fully exploit the channel reuse properties underwater. The channel reuse in recent protocols such as RIPT [24] and DOTS [64] is limited to the receiver side. While each receiver supports multiple sessions from different neighboring senders, there is no support for a sender to initiate multiple sessions to the other nodes (also known as pipelining). APCAP attempts to enable multiple sessions at the sender side by transmitting packets out-of-order [39], but it does not detail scheduling strategies for out-of-order packet delivery. In fact, full packet pipelining support is challenging in that explicit flow/congestion control must be implemented at the MAC layer in a distributed fashion—a simple back-off scheme is not sufficient. So far none of the existing protocols considered this issue.

In this chapter, we propose a new underwater MAC protocol called Multi-session Floor Acquisition Multiple Access (M-FAMA) that permits senders to initiate multiple concurrent sessions\(^1\) to other receivers. Sources avoid collisions by calculating their neighbors’ transmission schedules and propagation delays from passively overheard message transmissions. By adding a small guard time to calculated transmission schedules, M-FAMA protects against collisions arising from node mobility (as depicted in Figure 2.1). M-FAMA was inspired by the DOTS protocol and makes three important contributions beyond DOTS: 1) it allows multiple outgoing sessions from each source and multiple (pipelined) packets on each session; 2) it applies a localized distributed algorithm (called bandwidth balancing) to maintain max-min fairness between sessions within the same collision domain; 3) it shows significant gains with respect to DOTS when applied to a representative underwater monitoring and surveillance scenario. The remainder of the chapter defines the M-FAMA protocol and reports extensive simula-

---

\(^1\)The term *session* conventionally refers to opening, closing, and managing a communications dialogue between end-user application processes. In this paper, we use it to describe the time during which an RTS-CTS-DATA-ACK sequence is exchanged between a sender and its intended receiver.
tion experiments comparing M-FAMA with existing underwater MAC protocols.

2.2 M-FAMA: Motivations and Basic Principles

An RTS/CTS exchange can be used to reduce the chance of collision interference due to the hidden terminal problem [50]. However, this solution does not prevent collisions entirely—especially in a high-latency environment like the Underwater Acoustic scenarios. Fullmer found that imposing wait times on RTS/CTS transmissions can reduce collisions in cases of channel contention [34]. Moreover, he identified the following two conditions for collision-free transmission when using an RTS/CTS mechanism:

- **RTS wait time**: The time from RTS transmission to receiver issue of CTS response – should be greater than the maximum propagation delay (the time for a transmitted frame to reach its maximum transmission range)

- **CTS wait time**: The time a sender waits to receive a CTS reply after transmitting its RTS – should be greater than the RTS transmission time plus double the maximum propagation delay plus the hardware transmit-to-receive transition time.

Based on this work, Molins et al. proposed Slotted-FAMA [61] for underwater networks where node communications are slot-synchronized, and any packet exchanges (including RTS/CTS/ACK control packets) can only happen at the beginning of a slot. While the protocol experiences fewer collisions, guard times within slots and fixed slot size exacerbate the delays between communications.

Propagation delays increase collisions and reduce the channel throughput when using an RTS/CTS mechanism. Fortunately, large propagation delay also creates a new opportunity to achieve higher throughput by reusing the channel with interleaved sessions. Note that collisions only occur at the receiver side, not the sender side [17]. A collision occurs when two or more signals arrive simultaneously at a receiver, which is unable to decode the overlapped signals. In light of this definition, we relax the
constraint imposed by most MAC protocols that the sender be protected from interference. Opportunistic concurrent transmissions, when the transmitter is “exposed”, can improve the throughput significantly.

Figure 2.2 shows a network topology useful for illustrating the benefits of channel reuse in M-FAMA. In this case, node $A$ and node $C$ are hidden from each other, but node $B$ is within the transmission range of $A$ and $C$. As depicted in Figure 2.3, Slotted FAMA restricts channel access to only one sender-receiver pair (e.g., node $B$ and $A$) during the slot time. Slot time is determined by the maximum propagation delay, which is 0.5 second for the $750m$ communication range in an UW-ASN. With no channel reuse and a $16kbps$ data rate acoustic connection, sending a $128byte$ (or $1024bits$) data packet during a RTS-CTS-DATA-ACK (i.e., session) represents a worst-case channel utilization of $3.2\%$.

Multi-session FAMA leverages the RTS/CTS exchange for learning propagation times to and between neighbors. With loose clock synchronization among terminals
and a known transmission and propagation time for each control and DATA packet (assumed of fixed size), each terminal can calculate its neighbors’ transmission and reception schedules by promiscuously overhearing the neighbor’s transmissions. Using the knowledge of neighbors’ schedules, a node can schedule collision-free transmissions of its own.

Figure 2.4 shows how M-FAMA leverages this knowledge to actively initiate four sessions for two receivers; in the same amount of time, S-FAMA (Figure 2.3) is only able to transfer one RTS-CTS-DATA-ACK sequence to a single receiver. In the case depicted in Figure 2.4, node B first transmits an RTS destined for node A. While the RTS packet is still propagating, node B then transmits another RTS destined for node C. When node A receives its RTS, it waits until time (packet transmission time + maximum propagation delay) and then replies with a CTS. Meanwhile, node C has also received its RTS, and replies by transmitting CTS after waiting the appropriate amount of time. Node B receives CTS messages from nodes A and C sequentially, and then sends the necessary DATA messages consecutively. Note that node B actively initiates second sessions for both nodes A and C by sending another RTS to the destination before receiving an ACK from nodes A or C for the previous session’s DATA transmissions. The reader will notice that the number of simultaneous sessions from the central node to the peripheral nodes can be increased as the ratio \( \{\text{propagation time/\textit{transmission time}} \} \) increases.

To motivate the use of M-FAMA in more general topologies, consider the application depicted in Figure 2.1. Several underwater sensors are crawling the bottom of the sea, in part carried by the underwater currents, mapping the habitat, say, with video cameras. The video is low resolution, possibly a sequence of still images. It is stored at the underwater source and delivered as a multimedia file to sonobuoys; from these it is forwarded to the support ship via radio. Swarm nodes, sonobuoys and ship can move with currents. The continuous motion prevents the use of tethers (i.e., fibers and/or cables) that connect sensors to a support ship. Instead, untethered node-to-node
and node-to-buoy communications are used. The dynamic underwater sink tree that connects various sea bottom crawlers to one sonobuoy must multiplex several streams from several different sources. More precisely, at each intermediate repeater, multiple streams (from different sources) must be forwarded upwards. This causes an intermediate node in one tree to be exposed to interference from other trees. We will show that M-FAMA spatial multiplexing feature can efficiently handle these problems. Moreover, M-FAMA multi-session pipelining will be very effective with links experiencing large propagation/transmission delay ratios. The experiment will demonstrate the throughput gains and video improvements yielded by M-FAMA over existing protocols.

2.3 Towards Enabling Multiple Transmission Sessions

Since M-FAMA was inspired by the DOTS [64] protocol, we discuss our preliminary study on enabling multiple transmission sessions in DOTS in this section. Here, the term session refers to opening, closing, and managing a communications dialogue between end-user application processes (i.e., a sequence of RTS-CTS-DATA-ACK packet exchanges between a sender and its intended receiver). In the original DOTS, a receiver opportunistically has permitted multi-session reception whenever an incoming RTS does not conflict with its current ongoing transmission session. However, a sender can have only one outstanding packet (i.e., single transmission session but multiple reception sessions). Given that there could be opportunities of having multiple outstanding packet transmissions, we investigate a mechanism of scheduling multi-session
transmission (called MDOTS). We allow each node to manage multiple independent sessions, and thus, there could be multiple outstanding packets within a session period (pipelined).

To illustrate the advantages of MDOTS over DOTS, consider a line topology \((A - B - C)\) where node \(B\) can reach both \(A\) and \(C\), but they are hidden from one another. In this case, DOTS is only able to transfer one session (i.e., RTS-CTS-DATA-ACK sequence) to a single receiver. As depicted in Figure 2.5, in the same amount of time, MDOTS can actively initiate two different sessions (one to node \(A\) and the other to \(C\)); node \(B\) first transmits an RTS destined for node \(A\). While the RTS packet is still propagating, node \(B\) waits for a random period of time and then transmits another RTS destined for node \(C\). When node \(A\) receives its RTS, it waits until time has passed (i.e., total packet transmission time + maximum propagation delay) and then replies with a CTS. Meanwhile, node \(C\) has also received its RTS, and replies with a CTS after waiting the appropriate amount of time. Node \(B\) receives CTS messages from both nodes \(A\) and \(C\) sequentially and then sends the DATA packets respectively. Note that node \(B\) can initiate multiple sessions to each destination node (either \(A\) or \(C\)) by sending another RTS to the destination before receiving an ACK from nodes \(A\) or \(C\) for the previous session’s DATA transmissions.

To support this, the MAC layer is extended to maintain a buffer that queues multiple packets received from the Network layer. It allows us to maintain multiple concurrent packet transmissions. MDOTS can easily support out of packet delivery in case of head of line blocking. Consider a case where node \(B\) has two packets in a buffer (the first destined to node \(A\) and the second destined to node \(C\)). Assuming that node \(B\) cannot transmit a packet to node \(A\) due to expected collision, it can immediately start a new session with node \(C\). The scheduling process is the same as DOTS. When scheduling multiple packet transmissions, a sender must track the number of outstanding transmission sessions and simultaneously check expected collisions with the delay map. To avoid a single node capturing (or monopolizing) the whole channel, we simply limit the
maximum number of transmission sessions on the fly per node. In our study, we vary the maximum number from 2, 4, 8, and 16 (denoted as MDOTS-2, -4, -8, and -16). To understand the performance benefits of MDOTS, we perform preliminary simulations in two representative topologies namely the topology of 4 Senders and 1 Sink (Figure 2.6), and the topology of 1 Sender and 4 Sinks. We expect that the topology of 1 Sender and 4 Sinks is more favorable to M-DOTS variants because the sender does not compete with other neighbors to access channel but can maintain the maximum number of outstanding sessions to the sinks. In the simulations, we set the distance between Sender and Sink as 750m, the data rate is set to 50kbps and the size of packets as 512bytes.

*1 Sender and 4 Sinks:* Figure 2.7 shows the throughput of the five protocol in the topology of 1 Sender and 4 Sinks (shown in Figure 2.13(b)) with data packet size of 512byte and transmission range of 750m. In this scenario, all MDOTS protocols outperform original DOTS protocol. Amongst the MDOTS variants, MDOTS-16 shows the best performance. The 4- topology is similar in layout to the topology of 4 Senders and 1 Sink (Figure 2.6), but the roles of Sender and Sink are switched. Instead of four outer nodes all trying to start sessions with a central node, one central node tries to initiate sessions with the 4 Sinks. This configuration works very well with aggressive versions of MDOTS (e.g., MDOTS-16). In this topology, a central node is a participant of every communication, so it always has full and current knowledge of all ongoing communication sessions. The delay map and collision avoidance of MDOTS proto-
cols allow the central sender to leverage its full knowledge and schedule transmission sessions without any collisions, leading to better throughput. The results show that MDOTS-16 achieves the saturated throughput performance.

4 Senders and 1 Sink: Figure 2.8 shows the throughput of the five protocols in 4-Senders 1-Sink topology (shown in Figure 2.6) when data packet size is 512 bytes and
transmission range is 750m. For this case, MDOTS-2 shows the best throughput performance. However, MDOTS protocols with more than 2 sessions show degraded throughput than original DOTS protocol. With this topology, four senders are competing with each other. None of the senders are close enough to directly overhear each other’s RTS message. A sender will never hear a RTS from another sender, which means that senders cannot find out ongoing sessions between the sink and another sender until it overhears the CTS from the sink. Since the senders do not estimate the degree of congestion, aggressive protocols like MDOTS-4, MDOTS-8, and MDOTS-16 will tend to be overly aggressive in sending RTS messages trying to capture the sink’s attention. The sink will be overwhelmed with many RTS messages from the four senders. This partly explains the lower performance results of MDOTS-8 and MDOTS-16. To alleviate and provide reliable throughput performance, we need a mechanism to accurately measure contending senders in a contention domain and allow promising new sessions.

Although a preliminary evaluation showed that MDOTS significantly outperforms the original DOTS, we had to completely redesign MDOTS. Since MDOTS is a highly aggressive protocol, it performs very poor in terms of the fairness. Moreover, the throughput is unacceptable when there are multiple contenders sharing the same channel as shown in Figure 2.8. To fix this, M-FAMA has come into the world after we investigated a mechanism of dynamically adjusting the number of sessions for fair bandwidth sharing (and permitting variable packet size and packet pipelining). The following section provides a detailed description of how M-FAMA performs scheduling and maintenance, and presents two variants (conservative and aggressive) of the M-FAMA protocol.

2.4 Multi-session FAMA design

As previously mentioned, M-FAMA nodes monitor MAC-level state information to avoid collisions. This information includes transmission and reception schedules af-
fecting one-hop neighbors, and a delay map indicating propagation delays between the
node and its one-hop neighbors, and between one- and two-hop neighbors. To allow
state information to be assembled entirely through passive channel observation, we en-
hance the MAC frame headers with supplementary data (discussed in Section 2.4.1).
Based on the type of MAC frame overheard (e.g., RTS, CTS, DATA, ACK) and the
delay map information, a node is able to infer when its one-hop neighbors will be re-
ceiving transmissions. This information allows the node to avoid opening a session that
would collide with a neighbor’s message reception. Similarly, combining the node’s
knowledge of its current sessions with the delay map information allows it to calculate
the times when it will be receiving packets. Thus, the node can avoid opening sessions
that would conflict with its own receptions. To avoid scheduling errors caused by node
mobility or clock skew, we incorporate a motion-dependent guard time, which will be
further discussed in Section 2.4.1. Whenever a node has a frame to send, it compares
the transmission against neighboring and local reception times to detect potential col-
lisions (Section 2.4.2). If no conflicts are detected, the node begins its transmission;
otherwise, it backs off the communication. Since acoustic fading/scattering may in-
terfere with overheard transmissions, collisions cannot be entirely eliminated through
this mechanism. M-FAMA provides an optimized recovery scheme (Section 2.4.5).
Further, given that fairness is not guaranteed (e.g., some sources with high data rate
could capture an unfair fraction of the channel), we enforce fairness with a “bandwidth
balancing” policy across all sources (Section 2.4.6).

2.4.1 Delay Map Management and Guard Time

For a node to build the delay map through passive listening to the channel, each frame
header must contain the following information:

- *source address*: the sender of the observed MAC frame
- *destination address*: the intended destination for the observed MAC frame
• transmission timestamp: the time at which the observed MAC frame was sent
• src-dest delay: the estimated propagation delay between the source and the destination

By inspecting overheard frames from neighboring transmissions, an M-FAMA terminal is thus able to construct the delay map of the propagation delays with its one-hop neighbors, and between its one- and two-hop neighbors. Combining the delay information with type of the message overheard, the node can predict future transmissions and receptions in the channel. Whenever the node’s MAC layer seeks to transmit a frame, it calculates the transmission and reception times for all the messages (RTS, CTS, DATA, and ACK) required for successful transmission to the destination terminal. If any of these communications would collide with any neighboring or local receptions, the node refrains from transmitting, backing off its communications until no further collisions can occur.

Noh et al. [65] previously demonstrated that time synchronization can be efficiently achieved and maintained in underwater acoustic networks. With clock synchronization across sensor nodes, the value of the timestamp provides timing information for each frame; each node can calculate the propagation delay to a neighbor by subtracting the transmission timestamp of the MAC frame from the reception time of the MAC frame. Using this information, the overhearing node can deduce the expected times when it will be overhearing future communications for the session, and when its neighbors will be receiving transmissions of their own.

This process is depicted in Figure 2.9, where node C overhears an RTS message from node B for an intended communication with node A. Node C calculates B’s expected CTS reception time as the RTS timestamp + maximum propagation delay + estimated src-dest delay. Node C can also estimate when it will overhear B’s data transmission, which is node B’s RTS timestamp + 2 * maximum propagation time + propagation delay between node B and C. Finally, node C determines B’s expected ACK
reception time as node B’s RTS timestamp + 2 * maximum propagation time + hardware receive-to-transmit transition time + expected src-dest delay. This information is incorporated into C’s delay map and neighboring transmission schedules, allowing C to avoid transmissions that would lead to collisions at its neighbors, or where C would need to receive data at the same time a neighboring message would be overheard.

To cope with node mobility caused by the ocean currents, M-FAMA introduces a guard time in its calculations. The basic idea is to estimate the displacement of the mobile node between two subsequent control packet advertisements and use a corrective term, called guard time, to account for such error. Assuming that each node will announce itself after an interval proportional to the max acoustic signal RTT between any two nodes, the displacement will be equal to the distance covered in the above interval. Thus, each node calculates this guard time as 2*(average movement distance / speed of sound in water). Note that the multiplier (2) is used to account for the case where the sender and the receiver are moving in opposite directions. This guard time is then added to the end time for the frame reception in the delay map, providing a protection against collisions resulting from node mobility.

2.4.2 Delay-map Assisted Packet Scheduling

Whenever a node initiates a communication or overhears RTS or CTS transmission from its neighbors, the node calculates timing information for all future transmissions and receptions in the sequence. The node creates entries in its delay map for each of the
calculated times. Before sending an RTS for a new session, the node inspects its delay map to verify that all the components associated with this session (i.e., its own RTS and DATA; CTS and ACK from receiver) will not interfere with neighbors’ activities. Moreover it must verify that impending RTS and DATA packets from other nodes will not interfere with its own receiver reception. If these conditions are met, the node proceeds with the communication.

Based on the delay map, a node decides whether or not it can transmit a packet without possible interference with a neighbor node’s packet reception. To illustrate the advantage of M-FAMA, consider an exposed terminal topology (y-x u-v) where two nodes, x and u, are exposed to each other. As depicted in Figure 2.10, Node x first transmits an RTS destined for node y. Node y replies with a CTS after waiting for the appropriate amount of time (i.e., total packet transmission time + maximum propagation delay). While node y is waiting for CTS transmission, node u also receives this RTS and has data to send. Considering that a collision only occurs in receiver side, it can begin its own transmission to node v concurrently if the following two conditions hold:

- **Neighboring non-interference:** Its current transmission (RTS) and future transmission (DATA) must not interfere with neighbors’ ongoing and prospective receptions (node u’s prospective RTS and DATA transmissions should not interfere with node x’s CTS and ACK receptions).

- **Prospective non-interference:** Its future receptions (CTS and ACK) must not be interfered with by neighbors’ prospective transmissions (node u’s prospective CTS and ACK receptions should not be interfered with by node x’s prospective DATA transmission).
Figure 2.10: An example of a concurrent transmission schedule

2.4.3 Schedule Recovery

A node may miss its neighbors’ RTS/CTS/DATA packets due to the half-duplex nature of the acoustic modem or the lossy nature of an acoustic channel. Under such circumstances, a node may begin its transmission sequence with an incomplete delay map, which may cause packet collision. Since each transmission decision is made locally, collision-free scheduling is not guaranteed.

Similar to the previous work [64], in M-FAMA, schedule recovery happens at both sender and receiver sides. At the sender side, when an RTS or a DATA frame is sent, a timer is set to the duration by which the corresponding CTS or ACK frame is received. Once this timer expires, the sender realizes that its transmission has been unsuccessful. In either case (i.e., no CTS or ACK reception), the sender will back off and issue a new RTS. M-FAMA takes a conservative approach of sending a new RTS for the missing ACK to lower the potential damage; i.e., due to an incomplete delay map we cannot guarantee safe retransmission of a large packet at that moment. At the receiver side, a packet loss can be detected in a similar fashion when the DATA frame does not arrive before a timer expires. Once the timer expires, the receiver can reset its state either to send frames (if it has any) or to receive future frames.

A node will update its delay map whenever it receives or overhears any packets from its neighbors. Due to packet loss, however, when updating a node’s local delay map, scheduled Tx/Rx packets at the node may find schedule conflicts with those of neighboring nodes. Conflicts may happen in the following cases; i.e., (1) a node overhears
a CTS packet from its neighboring node, but it has failed to overhear the correspond-
ing RTS packet as packet loss has happened or its location is far apart (i.e., located in
the same contention domain with the receiver but out of the contention domain of
the sender); (2) a node overhears a DATA packet from its neighboring node, but it has
failed to overhear the corresponding RTS packet due to packet loss or RTS-CTS packet
pair (common contention case). In general, we cannot salvage scheduled Rx packets
since a node cannot reschedule Tx packets at the remote nodes under the current pro-
tocol model. If there are two or more Rx schedules suffered from a conflict, schedule
recovery with multiple senders will be prioritized based on the initial RTS timestamps
in a local delay map. Unlike scheduled RX packets, we find that Tx packets at a node
can be easily re-scheduled to avoid collision; then, the updated delay map will be used
to re-schedule deferred Tx packets.

2.4.4 Enabling Multiple Sessions

The M-FAMA MAC layer maintains a buffer where it queues packets received from the
Network layer. This allows us to avoid Head-of-Line blocking (HOL) without violat-
ing protocol layering by directly accessing the network-layer queue to find unblocked
prospective sessions with other destinations. From the sender’s perspective, this MAC-
layer queuing permits a new session to be created whenever a packet received from the
Network layer is allowed to be transmitted. New sessions can be established sequen-
tially to the same receiver—this is the feature that we generally call pipelining. They
are all called sessions in this context. When the sender session is created, the node
compares transmission and reception schedules for the new session against the existing
communication schedules. If a collision is anticipated, the state’s timing information
is reset, and the node will reattempt the transmission session after a backoff period.
Otherwise, the node will send the RTS packet for this session and wait to receive a CTS
packet. When the CTS packet arrives, the node sends a DATA packet and waits for an
ACK packet to arrive. The ACK packet signals completion of the session, meaning that
the session has terminated.

For the receiver, a new session is created whenever the node receives an RTS. When the receiver session is created, the node compares transmission and reception schedules for the new session against the existing communications schedules. If a collision is anticipated, the session is closed and the sender will need to send another RTS to attempt the session again. Else, the receiver returns the CTS packet for this session and waits to receive the DATA packet from the sender. When the DATA packet arrives, the node extracts the Network-layer packet and sends it up to the Network layer. The receiver sends an ACK packet and terminates the session. Algorithms for the sender and receiver state transitions are shown in Algorithm 1 and Algorithm 2. These algorithms show the session state transition logic, but for simplicity, do not include the details of delay map management, collision detection, or timer design.

There are two M-FAMA variants, Conservative and Aggressive. They differ in terms of when senders are permitted to open new sessions. Figure 2.11 and Figure 2.12 show how multiple sessions are maintained in these two M-FAMA variants. The M-FAMA Conservative mode penalizes pipelining in favor of spatial multiplexing. Namely, the node is not allowed to start the next session for the same destination until it transmits the first session’s DATA packet. However, it can freely open new sessions with different destinations, taking advantage of spatial reuse. For example, in Figure 2.11, the sending node B cannot open a new session for the node A, but it can open a new session with a different destination, e.g. node C. Since the second session for the same destination is allowed only after transmission of the DATA packet for the first session, the sending node will never have more than two active sessions per destination. This feature is meant to improve fairness, by preventing one receiver from monopolizing the sender, while the other potential receivers are starved. The conservative mode is well suited to networks with rich fan out and relatively short propagation delays.

In contrast, M-FAMA Aggressive permits to pipeline more sessions to the same destination. The next transmission is scheduled after the transmission of the previous
Algorithm 1 Algorithm for the sender

1: procedure packet_to_send (message m, receiver rx)
2: a new session for a receiver (Sr_x)
3: if collision detected(Sr_x) then
4:  Sr_x, status:=backoff
5: else
6:  if received RTS from Network layer then
7:     insert Sr_x entry in delay map
8:  Sr_x, status:=SENDING_RTS
9:  send RTS message
10:  if M-FAMA mode == Aggressive then
11:     call Network_layer_has_packet_to_send to initiate a new session
12: end if
13: set expiration timer for CTS
14:  Sr_x, status:=WAITING_FOR_CTS
15: else if received CTS for Sr_x then
16:  if M-FAMA mode == Aggressive then
17:     call Network_layer_has_packet_to_send to initiate a new session
18: end if
19:  wait until maximum propagation delay for the CTS
20:  send DATA packet
21:  set expiration timer for ACK
22:  Sr_x, status:=WAITING_FOR_ACK
23:  call Network_layer_has_packet_to_send to initiate a new session
24: else if received ACK for this session then
25:  session complete, delete Sr_x
26: end if
27: end if
28: end procedure
Algorithm 2 Algorithm for the receiver

1: procedure packet_to_receive (message m, transmitter tx)
2: a new session for a transmitter (S_{tx})
3: if collision detected(S_{tx}) then
4:   delete S_{tx} return
5: else
6:   insert S_{tx} entry in delay map
7:   if received RTS for this node then
8:     send CTS packet
9:     set expiration timer for DATA
10:    S_{tx}, status:=WAITING_FOR_DATA
11:  else if received DATA for this node then
12:     send up DATA to the Network layer
13:    send ACK packet
14:    session complete, delete S_{tx}
15: end if
16: end if
17: end procedure
RTS or reception of the CTS to this destination. In addition, new sessions are allowed immediately after the transmission of a previous session’s DATA packet to this destination. This allows a sender to open as many sessions per destination as the propagation delay permits. We see this in Figure 2.12, where B opens a connection with A and C, and then opens a second connection with both nodes without waiting for a CTS reply. M-FAMA Aggressive is designed to provide higher throughput in cases of low channel contention and high propagation delays (where pipelining is essential).

2.4.5 Backoff, Recovery, and Maintenance

Since each node checks against its delay map before starting a new session, new sessions cannot collide with existing sessions (provided that all RTS/CTS messages were overheard successfully). Control packet collisions can occur, for instance when RTS messages from different senders arrive at the same node. However, as the RTS packet size is very small, the probability of such a collision is very small. M-FAMA uses
a Binary Exponential Backoff (BEB) algorithm to recover from collisions. Given the long propagation delays and high error rates in our network, exponential growth in the Contention Window (CW) from a pure BEB scheme leads to an overly large window size. Therefore, we use a modified BEB scheme: each node starts a timer after its RTS transmission, and counts the overheard RTS packets from other nodes until the timer expires before receiving its own CTS. This count provides a heuristic estimate of the number of other nodes contending for the channel. We denote this number as the Observed Contender Count (OCC), similar to the counter in the backoff scheme proposed in T-Lohi [82]. Therefore, our backoff algorithm can be expressed as follows:

\[
\begin{align*}
CW &= \min(2 \times CW, CW_{\text{min}}2^{OCC}) & \text{← upon collision} \\
CW &= CW_{\text{min}} & \text{← upon success}
\end{align*}
\]  

(2.1)

The receiver upon receiving an RTS, will wait until a time equal to (RTS transmission timestamp + maximum propagation time) before sending a CTS response with the earliest packet creation time. This approach basically eliminates all spatial unfairness.

Each M-FAMA node uses a refresh and expiration mechanism to account for backed-off or canceled neighbor transmissions, and stale delay information for its one-hop neighbors. Whenever a new transmission is overheard, the node searches its state information for entries with matching source and destination fields. In the case of duplicate entries, the node keeps the entry with the latest timestamp. An expiration timer is set for each entry added to the delay map and scheduled transmissions. When the timer expires, the item is removed, to limit the delay map and transmission schedule overhead.

2.4.6 Bandwidth balancing

M-FAMA relies on the delay map to minimize collisions. At the same time, it greedily maximizes throughput within the given delay map constraints (especially if the aggressive multiplexing approach is used). However, fairness is not guaranteed; it is quite possible that some sources with high data rate capture an unfair fraction of the channel.
In this section we show how fairness can be enforced using a Bandwidth Balancing (BB) policy that was inspired by the DQDB fairness algorithm reported in [41] and by the follow-up paper that extended it to distributed bottleneck flow control for wireless networks [87]. To illustrate the BB approach, it will be useful to refer to Figure 2.13(a) example: multiple sources and single sink. It is clear that each source should maintain an adequate number of sessions (and/or a number of pipelined packets on each session) in order to maximize aggregate throughput based on total load conditions. However, it is possible that some sources, because of their position or their traffic characteristics may transmit more than their fair share. The throughput of each source must be controlled so as to yield max-min fairness. For M-FAMA we follow the BB model reported in [87]. The main departure from [87] is to control not the fraction of bandwidth used by each sender in a contention domain, but the number of multiple sessions and pipelined packets issued by each source. Following the BB algorithm, in M-FAMA each source measures over a proper history window the residual (i.e., unused) bandwidth in the acoustic channel. If some sources are hidden from others, a central sink as depicted in Figure 8(a) can by default hear everyone and can thus propagate the residual bandwidth information. The residual capacity \( R_i \) at node \( i \) is expressed as:

\[
R_i = \frac{T_{idle}}{T_p} C_i
\]  

(2.2)

\( C_i \) is the channel capacity at node \( i \) (same for all nodes in our case), \( T_p \) denotes the last measuring period and \( T_{idle} \) is the measured idle time. Large \( T_p \) allows accurate channel view but also long response time affecting the source node’s ability to react to network changes. In this study, \( T_p \) was set to \( 10 \times \) maximum propagation delay. The new allowed data rate \( \gamma_i' \) at node \( i \) is then expressed as:

\[
\gamma_i' = \alpha \left( R_i/OCC + \gamma_i \right)
\]  

(2.3)
Here, $\gamma_i$ is the current sending rate. Each node only takes a fraction $\alpha$ of the available bandwidth. The coefficient $\alpha$ varies between $[0, 1]$. If $\alpha$ is small, a large amount of bandwidth is wasted but the network converges fast to a fair operating point. If $\alpha$ is close to 1, a small amount of bandwidth is wasted but the network convergence time increases. Besides the BB bandwidth constraint, another constraint on the number of packets/sessions is posed in M-FAMA by the overheard control packets and corresponding delay maps. For example, a source that currently has only one session and is allowed an extra session by BB, can start the new session only if that session will not interfere with scheduled transmissions from other sources. The BB control scheme proved to be very effective to manage congestion and fairness and was implemented in the M-FAMA simulator model. All the M-FAMA experiments incorporate BB, except otherwise specified.

2.5 Simulation & Evaluation

2.5.1 Simulation Setup

For acoustic communications, the channel model described in [81] and [59] is implemented in the QualNet physical layer. As in [81, 19], we use Rayleigh fading to model small-scale fading. Unless otherwise mentioned, the data rate is set to 16$kbps$ as in [89, 57]. We use two different transmission ranges of 750$m$ and 1500$m$. We measure throughput consumption per node as a function of the offered load per node. The load is varied from a single frame generated every 30$sec$ up to a single frame every 0.25$sec$. In some scenarios the offered load exceeds the available capacity. In such cases, the stream rate is adaptively reduced causing a reduction in video resolution or a decrease in the number of frames per second. Each simulation experiment was run for 1$hour$.

We start by examining M-FAMA’s performance in the four topologies (see Figure 2.13). The 4-sender/1-sink star topology as in Figure 2.13(a) features four sender nodes competing to send their data to the center sink node. This topology is representative
Figure 2.13: Simulation Topologies
of a sea swarm engaged in scouting an underwater region and reporting multimedia information to a U/W command post. In the 1-sender/4-sink star topology (see Figure 2.13(a)), the sender and receiver roles are reversed. This topology is representative of a reliable multimedia file transmission from a single source (say diver or AUV) to nearby divers or ships or other agents. The 4 sink configuration provides an opportunity to examine temporal reuse in action, since the source staggers the transmissions over time. The SEA Swarm (tree) topology represents a sensor data gathering scenario in which a swarm of sensors deliver data to the closest sonobuoy on the surface (with a static routing table). The final scenario is a dynamically varying topology in which the nodes are randomly placed in a 3D cube and are carried by underwater currents. The random deployment features ten fully-connected nodes, moving within the 3D cube based on the ocean current model called MCM [20].

We compare M-FAMA with well-known underwater CSMA protocols, namely Slotted FAMA (S-FAMA) [61], DACAP [68], and DOTS [64]. S-FAMA is a synchronized underwater MAC protocol that eliminates the need for excessively long control packets via time slotting. DACAP is a non-synchronized CSMA protocol that allows each node pair to use different RTS/CTS handshake intervals depending on distance between nodes. To cope with possible collision, DACAP requires both the sender and receiver nodes to send warning messages when they detect possible collision, thus deferring pending data reception/transmission. DOTS is a synchronized CSMA protocol that harnesses both temporal and spatial reuse to improve throughput. Like M-FAMA, DOTS relies on overheard neighboring node transmissions, but it lacks the support of multiple sessions from the source. The Bandwidth Balancing gain parameter for M-FAMA is set to $\alpha$ as 0.8.

Two M-FAMA variants are used in the simulation. As previously discussed, the variants differ in the allowed number of per-destination sessions for each sender node. Results for these variants are labeled as M-FAMA (Con) and M-FAMA (Agg), where Con is M-FAMA's Conservative mode and Agg is Aggressive mode. We measure the
throughput of each flow at the receiver as a function of the offered load at each source
as in [68, 64]:

\[
\text{Offered Load} = \frac{\text{# of generated data frames} \times \text{Data size}}{\text{Simulation Duration} \times \text{Data rate}}
\] (2.4)

\[
\text{Throughput} = \frac{\text{# of rx data frames} \times \text{Data size}}{\text{Simulation Duration} \times \text{Data rate}}
\] (2.5)

2.5.2 Simulation Results

2.5.2.1 4-Sender/1-Sink Topology

Examining Figure 2.14 and Figure 2.15 we note that for range of 750m (Figure 2.14),
M-FAMA (Con) outperforms all the other protocols by large margins. The superior
performance of M-FAMA (Con) over M-FAMA (Agg) results from the fact that four
senders are competing for the channel over a short range, leading to congestion. M-
FAMA (Agg) attempts to open multiple sessions and suffers from a high RTS collision
rate. When the range is extended to 1500m as in Figure 2.15 the load is reduced because
of the higher round trip delay and M-FAMA (Agg) shows the best throughput outper-
forming M-FAMA (Con) by 10%. M-FAMA (Agg) takes advantage of the increased
propagation delay, by pipelining a larger number of sessions.

Recalling that M-FAMA is equipped with BB control, we demonstrate in Figure
2.16 and Figure 2.17 how the BB technique works in asymmetric traffic situations that
are prone to unfairness. The topology is the same as in Figure 2.15. However, in Figure
2.16, the sources start at different times. A and B start at time 0s while D and E start at
time 90s. As explained before, since packet size is fixed, the sender increases/decreases
its rate by controlling the number of consecutive session it opens and thus packets it
sends. In our example, senders A and B, upon the arrival of packets from D and E,
follow the BB instructions by gradually decreasing the number of outstanding sessions.
The Bandwidth Balancing algorithm converges after 120s. At equilibrium, all senders have roughly the same relative throughput of 0.15 as already noted in Figure 2.15. In the second experiment, two sources send a multimedia stream, and the remaining two sources send low rate sensor data (e.g., position, temperature, etc.). Max Min Fairness
Figure 2.16: 4Senders-1Sink: M-FAMA’s throughput convergence under homogeneous traffic

Figure 2.17: 4Senders-1Sink: M-FAMA’s throughput convergence under heterogeneous traffic

requires that the sensor sources transmit their full input rates, while the multimedia streams share the left over bandwidth. Figure 2.17 shows exactly this behavior. The aggregate throughput is about 0.6 i.e., the same as in Figure 2.15 where 4 multimedia sources are active. Figure 2.18 confirms that BB is essential to provide fairness
and maintain stability. The two M-FAMA version (Con) and (Agg) are tested with and without BB. The versions without BB collapse and can achieve only $1/3$ of the throughput of the BB version. Moreover, BB does not introducing significant signaling overhead. Recall that the tradeoff between convergence time and bandwidth efficiency is controlled by the gain parameter $\alpha$. If the applications require faster convergence, this can be done by reducing the gain factor, at the expense of bandwidth efficiency.

### 2.5.2.2 1-Sender/4-Sink Topology

Figure 2.19 and Figure 2.20 show the results. For both transmission range of $750m$ (Figure 2.19) and $1.5km$ (Figure 2.20), M-FAMA (Con) and M-FAMA (Agg) dominate the other protocols. For the load of 4 packets/sec, M-FAMA (Con) and M-FAMA (Agg) outperform the rest by 200–500%. This topology is the ideal scenario for demonstrating M-FAMA’s ability for spatial and temporal reuse. Note that the central node has complete knowledge of all ongoing communication sessions (thus avoiding any collisions). As the number of session increases, the aggressive scheduling allows for higher throughput. If we compare this scenario with the previous scenario (4-sender/1-sink),
we note that in the 4 to 1 scenario the outside nodes have limited knowledge of the other ongoing sessions. With this limited knowledge, they are more likely to initiate sessions that will result in RTS collisions at the central node. In contrast, in the 1 to 4 scenario, the center knows everything. This explains the stronger performance of M-FAMA in
2.5.2.3 Sea Swarm (Tree) Topology

Figure 2.21 shows the results for Sea Swarm depicted in Figure 2.13(c). For simplicity, the topology is static, and all flows traverse from the bottom to the top (sonobouys). Now every packet is generated in the bottom nodes, and two hops are needed to deliver the packets to the surface. Conceptually, this scenario is similar to a multiple source/single sink scenario replicated over two hops. Not surprisingly the results are similar to those of the 4sender-1sink topology. The middle nodes suffer from congestion—there are three incoming flows, and a middle node is also in charge of transmitting data to the sonobouy. As expected, M-FAMA (Con) outperforms all the other protocols because of the heavy load and the relatively short range (1000m). Fair bandwidth sharing is also an important issue. Note that, in this topology, perfect bandwidth sharing is impossible due to high traffic concentration in the middle nodes. M-FAMA and DOTS exhibit about 0.8 of Jain’s fairness index [48]. The other protocols fare worse, S-FAMA with index = 0.5 and DACAP with index = 0.4 respectively.
2.5.2.4 Random Topology with Meandering Current Mobility (MCM)

The effects of random topologies and node mobility are examined in Figure 2.22 and Figure 2.23. Ten nodes are randomly deployed in a 3D cube with dimensions (866 m * 866 m * 866 m). Given the node transmission range, this topology enables full connectivity among all nodes. Each node follows a jet stream path generated by the MCM.
model [20]. The main jet stream speed of each node is set to 0.3m/s. The transmission range is set to 1.5km. Five sender-receiver node pairs are actively engaged in data communications, exchanging 128 byte data packets. Figure 2.22 shows that M-FAMA (Con) outperforms M-FAMA (Agg) by 15%, DOTS by 35%, DACAP by 100% and S-FAMA by 200%. M-FAMA (Con)’s superior performance over M-FAMA (Agg) is due to the relative high load caused by ten nodes in a relatively small cube, with low propagation delays.

2.5.2.5 Fairness

MAC protocols with backoff schemes based on incomplete information about network congestion may exhibit spatial unfairness (a form of channel capture), as described in Syed et al. [82]. We already addressed unfairness in M-FAMA and showed how to overcome it with the BB algorithm. We now examine the fairness of other protocols as well, using the benchmark topology in Figure 2.22 (random with MCM). To measure fairness, we use the Jain Fairness Index [48].

\[
\text{Fairness Index} = \frac{(\sum x_i)^2}{(n \cdot \sum x_i^2)} \quad (2.6)
\]

The results are shown in Figure 2.24. We already knew that M-FAMA + BB is fair. Figure 2.24 confirms it. We also discover that DOTS exhibits a high fairness index (0.9 and above), This is explained by the fact that DOTS in this case works in a Round Robin mode. Each source transmits one packet when it gets its turn. S-FAMA and DACAP on the other hand have low Jain Index and are subject to severe unfairness and capture. The capture and unstable behavior is also revealed by the large fairness index variance, indicating that different sources manage to capture the channel at different load conditions. Moreover, these results indicate that the problem of unfairness and capture in underwater networks is severe. Techniques like Bandwidth Balancing must be used to reestablish stability and fairness.
2.5.2.6 Guard Time and Energy

We also explored the impact of nodal speed and M-FAMA’s guard times, and the energy consumption of various protocols. The results are shown in Figure 2.25 and Figure 2.26. We provide a brief summary in the following. First, to understand the impact of speed, we varied the maximum speed of nodes (i.e., AUVs) from 0.3m/s to 3m/s, and found that mobility does not cause any significant throughput changes to M-FAMA under the scenarios considered. Second, we evaluated M-FAMA’s performance by varying the guard time in a mobile scenario (with 0.3m/s). Recall that if the guard time is too short, packet collisions significantly reduce throughput. If the guard time is too long, the lower temporal/spatial reuse reduces throughput. We found that the 1ms, 2ms, and 4ms guard time intervals achieve nearly identical throughput, while the 8ms and 16ms times have lower throughput due to lower utilization. In terms of energy consumption, we found that M-FAMA (Con) consumes more energy than S-FAMA, DACAP and DOTS because it delivers far more control frames than these three protocols as shown in Figure 2.26. However, the per-DATA energy for M-FAMA (Con) is significantly lower than the other protocols. In the dense random scenario, M-FAMA (Agg) consumes...
Figure 2.25: Guard Time sensitivity to a MCM mobility speed (0.3m/s)

more energy than M-FAMA (Con) even though it delivers fewer frames because the high node concentration leads to greater channel contention.
2.6 Related Work

Yackoski et al. [86] proposed UW-FLASHR, a variant TDMA protocol that can achieve higher channel utilization than existing TDMA protocols. Hsu et al. [46] proposed ST-MAC, an underwater TDMA protocol that operates by constructing a Spatial-Temporal Conflict Graph (ST-CG) to describe the conflict delays among transmission links, and reduces the ST-CS model to a new vertex coloring problem. A heuristic, called the Traffic-based One-step Trial Approach (TOTA), is then proposed to solve the coloring problem. Kredo et al. [54] proposed a TDMA-like protocol called STUMP that uses propagation delay information and prioritizes conflicting packet transmissions based on certain metrics (e.g., random ordering and uplink delay ordering). To reduce a TDMA protocol’s high dependency on topology information, Diamant et al. [27] proposed a spatial-reuse TDMA scheduling protocol with a broadcast scheduling algorithm called as robust broadcast scheduling problem (R-BSP). R-BSP adapts a combination of an underlying skeleton schedule (obtained from a topology-transparent schedule) and a topology-dependent schedule, which ultimately provides additional spatial reuse in case of reliable topology information. Ma et al. [60] proposed an efficient scheduling algorithm with constant approximation ratios to the optimum solutions for both unified and weighted traffic load scenarios. This work identifies the spatio-temporal link scheduling problem in UW-ASNs, which is significantly different from terrestrial wireless networks by a new conflict graph (more accurate than slotted spatio-temporal conflict graph in [46]) and provides interference-aware spatio-temporal link scheduling algorithms. However, TDMA-like protocols are not well suited to mobile, resource-constrained sensor networks due to poor failure resilience, computation effort, and requirement for network-wide time consensus. Diamant et al. [26] proposed a distributed collision avoidance handshake-based scheduling protocol that makes use of joint temporal and spatial reuse and will be referred to as the joint time and spatial reuse handshake protocol. This protocol improved existing solutions by considering spatial-temporal reuse but their applications are limited to stationary networks. Hence-
forth, we follow with a short discussion of related works in terrestrial networks and its applicability to mobile underwater CSMA protocols.

CSMA-like protocols (or reservation-based protocols) have also been proposed to exploit temporal reuse in several ways. Given that channel reservation (i.e., RTS/CTS) takes a long time, Guo et al. proposed Adaptive Propagation-delay-tolerant Collision Avoidance Protocol (APCAP). This scheme allows a node to transmit packets out-of-order [39], but does not detail scheduling strategies for out-of-order packet delivery. To reduce the control overhead (e.g., reservation, acknowledgement), R-MAC [85] delivers a burst of packets combined with delayed ACKs, thereby improving the channel throughput. Chen et al. proposed Ordered CSMA where the nodes transmit their data packets in a fixed order as in a Daisy Chain [22]. Given the fact that two sequential signals traveling in the same direction will not collide, each station transmits immediately after receiving a data frame from the previous station sequentially, instead of waiting for a period of maximum propagation delay. Yet, ordered CSMA is not appropriate for large-scale, mobile and multi-hop networks because generating an efficient node ordering requires knowledge of relative positions of all nodes in the network and a large number of packet exchanges. Moreover, it will be costly to maintain the order as nodes move and dynamically enter/exit the transmit chain.

Chirdchoo et al. [24] proposed a receiver-initiated reservation protocol called Receiver-Initiated Packet Train (RIPT) where after inviting neighbors with packets to transfer, the receiver accepts their transmission and builds a transmission schedule for the neighbors based on propagation delays. In RIPT, the receivers need to periodically initiate the packet transfers; under varying traffic demands, it is non-trivial to determine when to initiate packet transmissions. Chirdchoo et al. [25] proposed another reservation based protocol, MACA-MN, which increases the channel utilization by enabling multiple packet trains to the neighbors. MACA-MN allows each sender to send packet trains to the multiple neighbors by transmitting RTS with embedded information of the number of DATA packets for multiple intended neighbors and the inter-node propagation delay.
from the sender to its intended receivers. Kredo et al. [55] recently examined a range of TDMA-based channel scheduling methods (e.g., node/group/link/slot levels) to determine the best balance between performance and coordination overheads in underwater networks.

Ng et al. [63] proposed reverse opportunistic packet appending (ROPA), a sender-initiated handshaking based protocol where a sender solicits its one-hop neighbors to opportunistically append their packets to the original outstanding packet (packet trains) to increase channel utilization. Further, Ng et al. [62] presented Bidirectional-Concurrent MAC (BiC-MAC) which further enhances channel utilization of ROPA; a sender-receiver node pair can exchange multiple rounds of bidirectional packet transmissions for every handshake. Noh et al. proposed the DOTS protocol [65] which harnesses both temporal and spatial reuse to improve channel utilization. However, DOTS’ ability of temporal and spatial reuse is limited to the receiver side. There is no support for a sender to open concurrent sessions to the same destination. Unlike existing underwater CSMA solutions, M-FAMA neither requires an additional phase for reservation scheduling nor restricts transmission schedules to a specific order. M-FAMA is a sender-initiated protocol that relies solely on passively-overheard neighboring transmissions to make intelligent local scheduling of multiple concurrent sessions.
CHAPTER 3

Evaluation of Underwater Optical-Acoustic Hybrid Network

3.1 Backgrounds

According to statistics collected by National Oceanic and Atmospheric Administration, the ocean covers more than 70 percent of the Earth’s surface, yet over 95 percent of the underwater world remains unexplored. Researchers have studied various alternatives by building underwater networks to facilitate monitoring and exploration of underwater ecosystem. The deployment of underwater sensor networks provides valuable data, including water temperature, salinity, fish population, the ebb and flow of the tide; the information helps in addressing issues such as the effect of human activities on underwater ecosystem and the impact of pollutants on the marine environment. Another promising application of underwater networks is to launch unmanned robotics to record real-time videos of unexplored underwater environments, and send back the live video stream via underwater network. These applications require a high bandwidth, scalable and energy-efficient network; thus, researchers have studied the feasibility of acoustic and optical communication in underwater environment.

When using acoustic communications, devices can transmit at longer ranges compared to optical communications, but operate under the constraints of limited bandwidth and high energy consumption for transmissions and receptions. On the other hand, optical communications can provide higher bandwidth with lower energy consumption, but suffer from highly limited communication range (i.e., lesser than 50m). The rest of the
chapter is organized as follows. In Sections 3.2 and 3.3, we present our survey on underwater acoustic and optical communications, respectively. In Section 3.4, we discuss the tradeoff between acoustic and optical communications and further provide further preliminary simulation results. In Section 3.5, we propose a hybrid solution; combining acoustic and optical communications in order to obtain high enough bandwidth for video transmission and reduce energy consumption. In Section 3.6, we conduct extensive simulations to validate the performance of the proposed solution. Finally, we discuss the solution in Section 3.7.

3.2 Acoustic Communications

This section covers the basic properties of acoustic communications, an evaluation of using acoustic network, and the attenuation of acoustic signals.

3.2.1 Overview of Acoustic Communications

Existing radio wave communications (e.g., Wi-Fi, Zigbee) are unsuitable for underwater communications because water severely absorbs electromagnetic waves and causes radio wave signal strength to drop dramatically within centimeters of the transmitter. Thus, acoustic communication is commonly adopted in underwater communications for research and commercial uses. Underwater acoustic networking is popular due to several reasons: first of all, acoustic signals can be propagated over long distances in magnitudes of several kilometers, providing a notably large effective-range for transmission. Moreover, acoustic signals are broadcasted sound waves so that they have a wide field-of-view, often spread omnidirectionally. In the event of an obstacle is present in the line-of-sight between sender and receiver, sound waves may simply travel through non-absorbing materials, or go around the obstacle via a wide field-of-view. Therefore, acoustic communication does not strictly require line-of-sight.

Acoustic communications, however, have several drawbacks: the speed of sound
is relatively slower than electro-magnetic waves, resulting in a slow propagation delay between sender and receiver (around 1513.74 meter/s). Such propagation delay slows down the data rate; thus, acoustic communications result in highly limited bandwidth. We may potentially increase the frequency at which the acoustic signal is broadcasted, but increasing the frequency will result in larger attenuation and higher energy consumption, which will be discussed in section 3.2.3.

### 3.2.2 Using Acoustic Communications in Networking

When using acoustic signals for networking, we have sender nodes capable of broadcasting signals to receiver nodes, so the network characteristics are very similar to existing wireless networks like Wi-Fi. Acoustic networking thus will have similar issues as terrestrial wireless networks, such as the hidden terminal effect, interference, and signal collisions. Moreover, since the speed of sound in water is much slower than the speed of electromagnetic waves in air, interference issues are even worse.

Qadri and Shah [71] have evaluated the performance of applying existing routing protocols (DSR, AODV, DSDV, and OLSR) in underwater acoustic sensor net-
works. They conclude that DSR is not suitable because of low packet delivery ratio and throughput, and OLSR is not suitable due to its high energy-consumption. AODV and DSDV have better performance but different trade-offs. AODV is suitable for denser network of less traffic, while DSDV is suitable for high traffic of regular network.

### 3.2.3 Acoustic Signal Attenuation

When evaluating the general performance of wireless communications, one important factor to consider is the attenuation of signals under different conditions. According to the work by Stefanov et al. in [80], the attenuation of acoustic signals can be modeled by the following equation:

$$A(d, f) = d^k a(f)^d$$

where $A(d, f)$ represents the amount of attenuation over distance $d$ and frequency $f$, and the normalizing constant $A_0$ and spreading factor $k = 1.5$ are fixed values. According to this equation, the amount of attenuation $A$ increases as distance $d$ increases. This equation shows that the further the receiver node is away from the sender node, the more difficult it is for a signal can be transmitted to the receiver. We observe that the absorption coefficient $A(d)$ described by the Thorp’s formula [59] shows the directly proportional relationship between absorption coefficient and frequency. In other words, acoustic signals attenuate faster at higher frequency. Table 3.1 shows two commercial acoustic modems’ data from [21], and we observe that while the EvoLogics modem is operating at 48-78 kHz, it only has an effective range up to 6 km, whereas the Aquatec modem has a much higher range up to 20 km, but operating at lower frequency of 8-16 kHz.

A node can theoretically transfer more data per second by using a higher frequency. However, higher frequencies not only yield higher attenuation, as discussed before, but also require higher power consumption. In Table 3.1, we observe that the EvoLogics
modem has a higher data rate, but it consumes much more power over transmission compared to the Aquatec modem.

<table>
<thead>
<tr>
<th>Modem Name</th>
<th>AquaModem</th>
<th>S2C series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Aquatec</td>
<td>EvoLogics</td>
</tr>
<tr>
<td>Frequency band</td>
<td>8-16 kHz</td>
<td>48-78 kHz</td>
</tr>
<tr>
<td>Data rate</td>
<td>300-2000 bps</td>
<td>Up to 20 kbps</td>
</tr>
<tr>
<td>Transmission Power</td>
<td>20 W</td>
<td>100 W</td>
</tr>
<tr>
<td>Range</td>
<td>Up to 20 km</td>
<td>Up to 6 km</td>
</tr>
</tbody>
</table>

Table 3.1: Existing Acoustic Modems

3.3 Optical Communications

This section covers an overview of optical communications, an evaluation of using optical networks, and the attenuation model for optical signals.

3.3.1 Overview of Optical Communications

From acoustic modem specifications in Table 3.1 and [21], we observe that acoustic communication supports a limited data rate, up to 20 kbps. A video streaming service requires much larger bandwidth than 20 kbps, so we consider the use of optical communications as a communication medium. Optical communications are currently experimental in underwater networks, and existing research includes [8, 29, 31] and [10]. Optical communications generally benefit from much higher bandwidth at lower energy consumption rate, as well as a lower propagation delay because the speed of light is much faster than the speed of sound.

Despite higher throughput at lower power, optical communications suffer from larger attenuation over distance, an issue that will be addressed in section 3.3.2. Moreover, optical communication has a much narrower field of view and requires line-of-sight between sender and receiver, which will be further discussed in section 3.3.3.
3.3.2 Optical Signal Attenuation

Optical signals have more restricted range due to higher attenuation. Anguita et al. [9] modeled the power of optical signals at receiver node in the following formula:

\[ P = \frac{2P_0A_r \cos \beta}{\pi L^2 (1 - \cos \theta)} + 2A_t \cdot e^{-cd} \]  \hspace{1cm} (3.2)

area of receiver \((A_r)\), inclination angle to receiver \((\beta)\), distance to receiver \((L)\),
transmitter light beam diverge angle \((\theta)\), area of transmitter \((A_t)\), attenuation coefficient \((c)\), and distance to sender \((d)\). The relationships of \(\beta\) and \(\theta\) are illustrated in Figure 3.2.

The inclination angle \(\beta\) denotes how far off a receiver node B is from the center of sender node A’s signal, and the transmitter light beam diverge angle \(\theta\) denotes one half of the field-of-view of sender A’s signal. According to Equation 3.2, the power decreases as \(\beta\) increases up to 90 degrees. In other words, the signal attenuation increases as the receiver node B moves away from the center of the light beam. Thus, a line-of-sight (i.e., alignment between the sender and receiver) is a significant factor to maximizing the receiving power. Also, the receiving power attenuates over a larger transmitter light beam diverge angle \(\theta\). Therefore, we can observe that a larger field-of-view also results in higher attenuation. In conclusion, for optimized receiving power, the optical communications requires both a narrower field-of-view and direct line-of-sight.

Equation 3.2 can be transformed to the following simpler model presented by Giles and Bankman in [36]:

\[ I = I_0 e^{-cd} \]  \hspace{1cm} (3.3)

In Equation 3.3, the transmitted intensity \(I_0\) attenuates over distance \(d\). Different water types have a different attenuation coefficient \(c\), as shown in Table 3.2 from [36].
Figure 3.2: Inclination angle $\beta$ and transmitter light-beam diverge angle $\theta$

Simulation shows that water with a higher attenuation coefficient suffers from quicker attenuation over shorter distance. In Figure 3.3, we observe that turbid harbor water with $c = 2.19$ has an effective range of less than 5 meters. In normal oceans, we have an effective range less than 20 meters. In optimal pure seawater, we finally have a possible range of up to 100 meters under optical communications. We conclude that optical communications suffer from large attenuation, with an effective range of less than 100 meters.

<table>
<thead>
<tr>
<th>Water Type</th>
<th>Attenuation Coefficient ($m^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Seawater</td>
<td>0.043</td>
</tr>
<tr>
<td>Clean Ocean</td>
<td>0.141</td>
</tr>
<tr>
<td>Coastal Ocean</td>
<td>0.398</td>
</tr>
<tr>
<td>Turbid Harbor</td>
<td>2.190</td>
</tr>
</tbody>
</table>

Table 3.2: Attenuation Coefficient of Different Water Conditions

### 3.3.3 Using Optical Communications in Networking

Arnon et al. [11] proposed a novel approach to overcome the environment without a line-of-sight by using a reflective communication link. However, a line-of-sight (i.e., alignment between the sender and receiver) is generally a strict requirement for optical communications. Thus, sender and receiver nodes need to establish a direct link before initiating data transmission. In other words, the optical modems on sender and receiver nodes need to be aligned before any packets can be transferred. Despite the fact that
optical communications is performed without wires, this point-to-point link topology makes optical networking similar to wired networking. Such transmission is characterized as more targeted and localized. Unlike acoustic signals, optical signals are not omnidirectional. Optical communications thus benefit from less interference issues and negligible propagation delay.

### 3.4 Acoustic vs. Optical Discussion

<table>
<thead>
<tr>
<th>Telemetry Method</th>
<th>Range</th>
<th>Data Rate</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic</td>
<td>Several km</td>
<td>1 kbps</td>
<td>100 bits/Joule</td>
</tr>
<tr>
<td>Optical</td>
<td>100 meters</td>
<td>1 Mbps</td>
<td>30,000 bits/Joule</td>
</tr>
</tbody>
</table>

Table 3.3: Acoustic vs. Optical

In order to determine the correct transmission medium to use and balance the trade-offs between optical and acoustic communications, the differences between the two mediums must be explored.
The primary difference between the two communication methods is the speed of propagation. When in water, the propagation speed of sound is roughly 1500 m/s. The propagation speed of light in water is around \(2.26 \times 10^8\) m/sec, slightly slower than the \(3 \times 10^8\) m/sec of air. In other words, the propagation speed of light is five orders of magnitude slower than the propagation speed of sound.

The tradeoff between transmission range and bitrate must also be considered when comparing the two methods of transmissions. Figure 3.4 is a chart showing the different acoustic and optical modems currently available. The optical modems are concentrated in the top left corner, meaning that their bitrates are orders of magnitude higher than the acoustic modems, and are measured in terms of megabits. However, their ranges are also much more limited than the acoustic modems, and are measured in terms of meters. The acoustic modems are spread out along the bottom half of the graph, with
a wide range of bitrates and distances. The bitrates for the modems are measured from bits to kilobits, and the ranges are measured on the order of magnitudes.

In Table 3.3 below, we see the conclusions drawn previously summarized by Farr, et al. [33]. The power efficiency of optical communications is significantly higher than the power efficiency of acoustic communications.

Another differing aspect between optical and acoustic communications is the field of view required by the modems. Acoustic communications can be made to be omnidirectional, and do not require direct line of sight between sender and receiver as the waves are able to make their way around obstacles. However, optical communications require direct line of sight between sender and receiver.

In Figure 3.5, the experiments done by Schill, et al. [76] demonstrate the possible fields of view of an optical transmitter. From their results, it is shown that optical transmitters can transmit up to about 120 degrees. However, as the receiver is placed further and further off from the center of the beam, the effective distance decreases. This effect is explained by the equation for the power of optical communications defined in Section 3.3.3, where the power of the transmission is dependent both on the inclination angle to the receiver and the light beam divergence angle. By forcing the transmitter...
to have a wide field of view, both of these angles are set to larger values, resulting in lower power.

3.5 Hybrid Solution

3.5.1 Objectives of Hybrid Solution

As seen in the previous sections, the acoustic and optical communications have their tradeoffs in terms of power, range, and bitrate. These tradeoffs must be balanced on a case-by-case basis in order to better tradeoff between acoustic and optical communications. Section 3.4 showed that optical communications had a higher bitrate and lower energy consumption, but a much shorter range compared to acoustic. Acoustic communications, on the other hand, had a slower bitrate and higher power consumption, but also a much longer range. In order to properly take advantage of the benefits of both solutions, a hybrid solution is necessary.

In Figure 3.6, a possible example of a hybrid solution is shown. In the hybrid solution, an Autonomous Underwater Vehicle (AUV) is equipped with both acoustic and optical modems. The AUV contains three acoustic receivers, an acoustic transmitter, an optical transmitter, and an optical receiver. The optical transmitter and receiver can be used to communicate with other nodes when within optical communication range and aligned. The acoustic receivers and transmitters will serve a dual purpose of transmit-
ting small bits of information over long ranges, and helping with the alignment of the optical communication components.

3.5.2 Acoustic Source Localization

Node alignment is essential in optical communications to achieve expected bandwidth performance. In this section, we present an acoustic source localization technique that maneuvers the time-difference-of-arrival (TDoA) calculation. Combination use of TDoA with a depth sensor then allows a node to locate the relative three-dimensional positions of other nodes, which will be used for the alignment between the sender and receiver.

3.5.2.1 Time-Difference-of-Arrival

Work by Liu et al. [58] presents a high-level explanation of acoustic source localization via the time-difference-of-arrival measurements. Assuming there is a receiver node B that wants to locate the position of sender node A, the TDoA technique requires a minimum of three acoustic receivers located at different points on the receiver B. Then, sender node A will broadcast an acoustic signal, which will arrive at the three acoustic receivers at different time. Using the time-difference-of-arrival of two points, we can anticipate all the possible points of the sound source in the form of a hyperbola. Then, we choose a different pair of nodes, and draw out another hyperbola. The location of the sound source can then be estimated by calculating the intersection of two hyperbolas, as shown in Figure 3.7.

3.5.2.2 Estimating relative 3D position

For three-dimensional underwater space, we design our triangular plane formed by three acoustic receivers to be orthogonal to z-axis. Each node will contain a depth sensor, such as a pressure sensor, to correctly calculate the absolute z position (i.e.,
depth) from water surface (avg. error $< 1\, \text{m}$ [49]). The z-position data is transferred from sender A to receiver B, so that the receiver can calculate the relative z-position. Then, the receiver uses the z-position data to project the speed of sound in xy plane, and uses TDoA to estimate x-position and y-position. Once all relative x-, y-, and z-positions are calculated, the node B can use this position data for sender-receiver alignment and routing.

3.5.2.3 Selecting a Node to Communicate With

Using the previously described alignment methods, an AUV can align its optical receiver with the sender’s transmitter using an onboard acoustic transmitter. However, if there is more than one source node present, it must be able to identify each node among the nodes. This requires a different method of alignment, and a preliminary version of this alignment protocol is described in the following.

We use the TDoA to triangulate the position of the other node. This assumes that the nodes are equipped with at least 3 acoustic antennas and a depth sensor, and that the nodes are within line-of-sight.

On a successful connection request, Node A requests a connection to Node B using
the Communication_Request packet. Node A waits for Node B to generate its own Communication_Request. This continues while the nodes move closer and closer into alignment, at which point the periodic communication request stops.

If Node B is busy, Node A determines this when it receives a Node_Busy message from Node B, which is currently involved with another transmission to Node C. Node A will then wait for the specific time requested by Node B, while Node C continues the transmission and discards the Node_Busy message that it also receives. Once Node A is done waiting, after a random interval, Node A sends another Communication_Request and begins the process again.

If Node B is unreachable, then Node A determines this when a predetermined period of time expires. Node A then resends the Communication_Request message with an updated timestamp and waits for the same amount of time. After 4 attempts at retransmission, Node A declares Node B unreachable and stops attempting to communicate with Node B.

There are several considerations to be taken when implementing this alignment algorithm. We account for the multipath effect by adding a timestamp to each packet. Since the multipath effect will result in a delay for other messages, the packet sent over the most direct path will arrive at the destination first. The timestamp messages with previous sequence number are discarded once the first message arrives with newer sequence number. Nodes must deal with synchronization issues, and must wait a random interval (i.e., random jitter) before submitting a Communication_Request instead. To achieve this alignment, acoustic conversations are highly necessary due to long communication ranges and omnidirectional spreads so that optical modem can initiate its communications after the alignment between the sender and receiver.
3.5.3 Hybrid Transmission

To properly utilize both types of communication possible, the following algorithm is proposed when node A wants to transmit to node B:

1. Node A sends an acoustic invite to Node B
2. Node B uses the information to triangulate the position of Node A and turns to align to Node A
3. Node B sends an acoustic response to Node A
4. Node A uses the acoustic response from Node B to triangulate the position of Node B and align to Node B
5. If Node A is currently out of optical communication range, it proceeds to move into range to transmit while using the acoustic modem to transmit data
6. Once Node A is in optical communication range, it either switches to using the optical modem exclusively for transmissions, or uses a combination of both the acoustic and optical modems to transmit data.

In the algorithm above, a node uses an acoustic modem when the intended receiver is located out of optical communication range. If the distance is close enough for optical communication range, then it will use optical communications after alignment. In the cases where the distance between the two nodes is long (i.e., out of optical communication range), or the amount of data that needs to be sent is small, then using acoustic communications will be optimal as it does not require alignment.
3.6 Simulation

3.6.1 Simulation Setup

The existing acoustic and proposed hybrid approaches are evaluated via QualNet simulator using actual modem properties. The setup is as follows:

The optical modem setting used is the AquaOptical II modem from Doniec, et al. [29]. The maximum range was set to be 50 meters and the maximum bitrate is 2.28 Mbps. For the acoustic modem setting, we used the S2CR 18/34 modem from Evologics with a range of 3.5 kilometers and a maximum bitrate of 13.9 kbps. The specifications of this modem were obtained from the Evologics web site [37]. Unless otherwise stated, the transmission power is 105 dB re $\mu$ Pa.

In the simulations, four source nodes were placed in a cube as shown in Figure 3.8. The dimensions of the cube were 1 km on each edge, with the source nodes at the bottom corners, equally distant from the sink node. The sink node was placed at the top of the cube like a surface buoy. The sources generated packets based on CBR
with predefined packet generation rates, and M-FAMA [44] was used for an underwater MAC protocol. For the optical communications, an AUV travels to each source node at a speed of 3m/s, and stops 50 meters away for 600 seconds at each source node to receive the data from the node.

The data packet size used by the acoustic modem was set to be 1.75kB, and the data packet size used by the optical modem was set to be 50kB. Unless otherwise specified, the average value of 50 runs with the 95% confidence interval is reported.

### 3.6.2 Simulation Results

In Figure 3.9, the average throughput of hybrid (i.e. combining acoustic and optical modems) and acoustic only approaches are plotted against the offered load. For the throughput of acoustic modem only case, it shows that the acoustic channel gradually saturated as the offered load increases. For the hybrid case, acoustic modem is used for long distance and optical modem is opportunistically used for short distance after alignment assisted by acoustic communications. To our surprise, the result reports that the proposed hybrid solution outperforms average throughput of the acoustic modem only case up to more than twenty times. The careful reader may notice that there is no huge gain in terms of the throughput between optical only and hybrid solution. However, it is noteworthy that the acoustic modem not only delivers data but also enables the optical communication by being guidance of AUV’s optical modem alignment. Less obviously but of equally importance, the acoustic modem can be used in cases of challenged environments for the optical communications namely long range communications, non-line-of-sight, and poor water quality. In reality, the acoustic modem is still essential part for the underwater acoustic sensor networks.

The proposed hybrid approach also has a significant advantage in terms of the energy efficiency. In Figure 3.10, the energy consumptions of the modems were plotted against the amount of data received. It is shown that there is no significant energy
consumption gap between the acoustic only and hybrid cases for the small amount of received data. However, the gap becomes distinct as the received data increase. Note that equipping a node with both modems provided it with the freedom to select the optimal transmission method with the lowest transmission time, but that using a combination of optical and acoustic transmissions while two nodes are within optical trans-
mission range yielded negligible time savings gains that were probably not worth the extra power expended to utilize both modems.

### 3.7 Discussion

In this chapter, we explored the properties of both underwater acoustic and optical communications. From our simulations, we determined that acoustic communications were well suited for transmitting small amounts of data over long distances, or for aligning nodes to prepare for optical communications. We proposed the concept of a hybrid system where a node is equipped with both acoustic and optical modems. From our extensive simulations, the proposed hybrid solution outperforms the case in which only the acoustic modem is used from both throughput and energy consumption perspectives. The performance gain from the acoustic modem is seemingly negligible in some cases. However, the importance of acoustic communications is still not to be disregarded. As seen in Table 3.2 and Figure 3.3, the attenuation of the optical modem depends greatly on the water conditions. In the case of a turbid harbor, the attenuation may be so severe that transmitting data via optical modem would be virtually impossible. In such cases, despite the high energy consumption and slow data rates, the acoustic modem would still be necessary to ensure that the data is still transmitted.
CHAPTER 4

Real-time Video Streaming from Mobile Underwater Sensors

4.1 The Hybrid Solution

We propose the hybrid solution illustrated in Figure 4.1 in order to take advantage of the benefits of both the acoustic and optical communications media. Our primary objective of acoustics is to provide real-time video monitoring between AUVs and Base Station at all times even if no optical link is available. Moreover, the acoustic channel carries back up video frames when the optical channel fails. We propose image processing techniques such as Canny edge detection or Sobel to reduce image size (like vector graphics) to 90 percent smaller than a grayscale original image. This way, we can expect at most 3-5 frames per second (good enough for real-time video monitoring). A low resolution greenscale image is also transferred via Acoustic channel (either pre-defined interval or specific image which is requested by user). On the other hand, high definition video is always transmitted via optical communications. Naturally, the system should withstand mobility (submarines, drifters and base all drift with the currents). As in Figure 4.2, a hybrid acousto-optic mesh can be dynamically configured and can maintain the equivalent of high speed optical links between its nodes. The control is completely supported by acoustic and the high speed data travels on optical links. The rest of the chapter is organized as follows. Section 4.1 describes the hybrid solution proposed in this chapter and presents representative scenarios. Section 4.2 examines image processing algorithms that can reduce the bandwidth required for
real-time video streaming. In Section 4.3, we briefly discuss the implementation of the acoustic system and evaluate the overall performance on tested images.

4.1.1 Design

Figure 4.1 shows a mobile AUV unit composed of both an acoustic and an optical modem. Also three additional acoustic receivers are included to improve transmission redundancy as well as to provide additional data for localizing another transmitter and aligning the optical modems. In order to carry this out, we use time-difference-of-arrival to determine the relative position of the sender in three dimensions, similar to the technique described by Liu et al. [58]. We also use the information from the depth sensor to determine the current node’s position on the z-axis, and transmit it to other nodes as necessary. During the alignment process, we continue sending data through acoustic communications, until the optical modems are aligned properly. Once the optical modems are aligned, we send data using optical communication, to take advantage of the increased bandwidth, but continue to use acoustic communication for sending control signals such as acknowledgements (ACKs).

Alignment takes place as follows: the sending node sends a request to the receiving
node over the acoustic medium. The receiving node responds to the request with the
acoustic transmitter as well, and both sender and receiver use the trilateration informa-
tion to move towards each other and to get into optical alignment. Once the nodes can
communicate over the optical modem and are within range of each other, they can begin
to use the optical medium for data transmissions. In the event that there are multiple
source nodes present that are transmitting, it is more difficult to determine where the
node that is sending the data is located, and to align the potential receiving optical sen-
sor properly. Therefore, previous work [43] includes a preliminary solution to address
this issue, which involves the proper node responding to the communication request
packet that the sending node transmits to nearby receivers.

4.1.2 Scenarios for The Hybrid Solution

This section shows a number of possible scenarios which can take advantage of the
proposed hybrid solution.

4.1.2.1 Bottom video exploration

One important reason for using the optical channel underwater is to exploit its high
bandwidth (up to several Mbps) for interactive video. The classic application is the de-
ployment of an U/W robot equipped with video camera at great depths from a surface
vessel for exploratory or recovery operations. While at low depths the robot can be
guided via a cable that carries data and possibly also power; at great depths the guiding
cable is not practical, it may get entangled. A proposed solution is to drop a “base
station” to the bottom from the surface ship. The base station is stationary and is con-
nected by cable to the surface ship. Multiple robots can roam from the base station in
several directions, they carry video cameras that are monitored by operators on the ship
and are used to remotely guide the robots. If the robots are at more than 50 m distance
from the base station, they will not be able to communicate directly. One interesting
scenario is the autonomous deployment of a mesh network that supports one or more video streams from robots to base station and ship. The key idea is to create an optical tree from robots to base station. The data and commands in the reverse direction (base station to robot) are carried via acoustic channels. The short distance, say < 200m, guarantees low delays (< 200ms) even for acoustic propagation and does not compromise real time interactivity. The challenge is to develop a totally distributed algorithm that takes the relay nodes from base station to the field and positions/orients them in the proper way (note, the nodes can localize themselves relative to the base station).

4.1.2.2 Shallow water inter-submarine video communications

Another application of the hybrid concept is the establishment of high speed video connections among a team of mini-submarines participating in a scouting expedition. As depicted in Figure 4.3, each submarine sends its video to all others. In this case, the acoustic modems are used to position the submarines and to align their lasers. The submarines provide the optical multihop mesh. In principle, each submarine carries two or more optical transmitters and several optical receivers, so that a mesh can be
maintained at all times. As a difference from the sea bottom robot operation, the submarines are equipped with independent, separately oriented lasers, so that they can beam two separate neighbors thus creating a mesh. Moreover, the links are full duplex. There are several challenges: first, the acoustic positioning; second the maintenance of a fully connected acoustic mesh among submarines; third, the alignment of the optical transmitters and receivers and the establishment of a connected, multihop optical mesh; finally, the support of video many to manycast.

4.1.2.3 Mixed pure and murky environments

We can use optical transmissions only in clear waters. In murky water we must use acoustic instead. Data rate must be dramatically scaled down from Mbps to a few hundreds bps (from motion video to still frames). In U/W operations near the surface or in shallow waters, paths conditions may change continuously and intermittently (from clear path to murky path). This is where the hybrid solution shines. We switch back and forth from optic to acoustic channel. If the switchover frequency is very high, it may be more convenient to use the two paths (optical and acoustic) in parallel with combined
still frame and video traffic. Network Coding may be considered in this case, to allow the exploitation of both streams in parallel, without having to switch path/technology on a continuous basis. Naturally, the challenges of merging two Network Coded streams from channels of such disparate speed and propagation delay are not indifferent.

### 4.1.2.4 Covert AUV to Ship communications

Suppose a ship deploys several AUVs in a 1000m radius to detect possible attackers or to scout the terrain with cameras. Enemy submarines may be listening and the operation must be covert in the sense that the AUVs should not be detected by submarines when they transmit. A possible strategy works as follows. The ship trails an underwater acoustic/optical mini base station. The base station periodically sends a ping via acoustic communications. When a friendly AUV in the area recognizes the ping, it uses it to align its transmitter to the base station. This enables the establishment of an optical link that cannot be detected by the enemy. In fact, since the ping is repeated periodically, the enemy does not know if there is indeed an AUV in the area.

### 4.2 Image Processing and Video Compression

Figure 4.4 shows an example of a device and how the video would stream. The concept is to compress the video file using image processing techniques. The video quality may be coarse but with every key frames having a greenscale that makes it acceptable until the optical link becomes available again. To test the feasibility of this concept, we implemented and evaluated a few image processing algorithms and examined the file size of the resulting images.

We used a series JPEG images of underwater scenes to conduct our tests. Some level of compression, such as the compression that the JPEG file format uses, is necessary in order to achieve the bandwidth savings. This compression is considered “lossy”, however, in that each time the file is saved and compressed, some data is lost, even if
there were no other changes performed to the image. These JPEG images are uncompressed by our programs before they are used; the uncompressed versions are composed of units known as pixels, arranged in a multidimensional array. Each pixel has a set of three RGB values from 0 to 255; these represent the red, green, and blue values for the color that the pixel should display. There is no differentiation between a grayscale image and a full-color image in the JPEG file format [3]. Our results comparing the average file size reductions from each technique are located in Figure 4.11.

4.2.1 Simple Image Processing Techniques

Averaging the RGB values for each pixel, and setting the new RGB values for each pixel to this average converts a full-color JPEG image to grayscale, with the appropriate shade of gray that corresponds to the color. This simple processing algorithm results in a file size reduction to 33 percent of the original size on average. By replicating the same value for all three RGB components of a pixel, we take advantage of the compression scheme and thus are able to reduce the file size greatly.

The blue and green colors travel the longest in the water because of their shortest
wavelengths. Therefore, the images taken in the water are always dominated by blue-green color. In fact, red, orange and yellow colors disappear in relatively shallow depth in the water due to their wavelength [75]. Therefore, underwater image enhancing techniques mainly focus on skewed color compensation caused by light scattering and color change [23]. Accordingly, we can use a variant of this technique that we call “greenscaling” to only set the green values of each pixel to this new value, and setting the other red and blue values to zero, resulting in an image with a green hue, but where the prominent features of the image can still be distinguished. This results in a slightly more optimal solution with an average resulting file size of 24 percent of the original, as the value for all the red and blue pixels is the same (zero), thus making the compression even more effective. However, these images still retain the same quality as the grayscale images, and only have an additional green hue. Both the grayscale and greenscale algorithms run in linear time, and are computationally efficient.

Following this, we run a simplistic edge detection algorithm that emphasizes sharp differences in RGB values, traversing the image horizontally. Using the grayscale image as a starting point, we used the mathematical differences in the average of the RGB values to determine where the edges are in the image. We found that this simplistic algorithm reduced the average size to 17 percent of the original. However, this was not a significant improvement over the greenscale algorithm because of the complexity of the resulting images since our algorithm found too many edges in the images. Attempts to remove the edges representing smaller numerical differences did not significantly improve the results. In addition to this, three out of the five resulting images produced pixels that were too lightly colored to detect, making the entire image appear to be nearly white.

Therefore, we decided to investigate more complex edge detection algorithms in order to detect a smaller number of edges with a higher resulting contrast, and to produce a less complex picture (and a smaller image file) as a result.
4.2.2 Sobel Image Processing

We first examine the Sobel edge detection algorithm for use in our proposed solution [40, 78]. There is ongoing research that would allow for Sobel edge detection to be performed in real time by a FPGA, which could easily be incorporated into an AUV, and would reduce the computational cost of performing this algorithm by converting the software computation into a hardware set of operations [53].

Edge detection algorithms are generally classified as based on either gradient or Laplacian methods. The Sobel algorithm [40] is a gradient algorithm, which means that it looks for the maximum and minimum of the first derivative of the values in the image. The primary component of the Sobel algorithm is the Sobel operator, which we use on a 5x5 block of pixels. While using the default 3x3 operator does provide true edge detection, we found that in pictures with noise the file size was not decreased enough for the Sobel algorithm to be effective. Therefore, we use a larger operator to find the larger edges in the image rather than noise.

We calculate the gradient for a particular pixel twice, in order to determine the
derivative by both row and column. Following this, we calculate the magnitude of the gradient vector to get one gradient to perform our Sobel calculations. Instead of handling the different RGB components, we start with the grayscale image and apply the result to the red, green, and blue components of the resulting pixels. Pixels on the same vertical or horizontal axis are factored into the calculations, but are not given as much weight as the pixels directly next to the current one. Pixels to the diagonal of the pixel in question are also factored into the calculations, but are given an even lesser weight compared to the ones that are both directly next to the current pixel and also those on the same direction as the one being analyzed. We illustrate this calculation below, displaying the coefficients that are multiplied by the values of the neighboring pixels, and representing the current pixel as the center:

<table>
<thead>
<tr>
<th>-5</th>
<th>-4</th>
<th>0</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8</td>
<td>-10</td>
<td>0</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>-10</td>
<td>-20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>-8</td>
<td>-10</td>
<td>0</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>-5</td>
<td>-4</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.1: Horizontal Sobel 5x5 mask [40]

<table>
<thead>
<tr>
<th>5</th>
<th>8</th>
<th>10</th>
<th>8</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-4</td>
<td>-10</td>
<td>-20</td>
<td>-10</td>
<td>-4</td>
</tr>
<tr>
<td>-5</td>
<td>-8</td>
<td>-10</td>
<td>-8</td>
<td>-5</td>
</tr>
</tbody>
</table>

Table 4.2: Vertical Sobel 5x5 mask [40]

Following this, we divide by the sum of the coefficients, in order to obtain a weighted average.

For our specific implementation, the pixels on the border of the image are set to white, as they cannot have a proper gradient calculated; this also decreases the final size of the file, while losing little data. We invert the result of the gradient and set a
pixel with a small gradient to white, and one with a large gradient to black. In order to
determine the combined magnitude of the horizontal and vertical gradients, we use the
following formula:

$$|G| = \sqrt{G_x^2 + G_y^2}$$ (4.1)

where $G_x$ and $G_y$ are the horizontal and vertical components.

The result of this algorithm generates an image that reflects all the gradients of the
original image, regardless of how large or small the gradients are. We find that there is a
significant file size reduction in most cases, at roughly 27 percent of the original size of
the image. In most cases, this new technique outperforms grayscale and greenscaling,
and provides usable images where the primary features can be distinguished in all of
the five cases, as opposed to the primitive edge detection algorithm. However, this is
not the most efficient solution, and thus we make an adjustment since we assume that
there are many small gradients that are contributing to the file size.

We discard the gradients that are small, and set those pixels to white to decrease
the file size and to emphasize the major edges. Our determination is made based on
the RGB value of a pixel: lighter values are numerically greater than the threshold, and
darker values are lesser than the threshold. Past attempts to do this with a universal
threshold were unsuccessful, as if the threshold was too low, then some of the images
would be white or almost entirely white and would be completely useless. But if the
threshold was too high, then other images would have very few pixels removed, and
have a much larger file size than necessary. Thus, it is necessary to have an adaptive
threshold for multiple types of images, including those images with lots of edges and
those that do not have them.

To incorporate an adaptive threshold, we begin with a numerical threshold of RGB
value = 250, which already eliminates a small portion of the grayscale spectrum. Then,
we evaluate the percentage of pixels that represent edges, compared to all of the pixels
of the image. If the percentage of pixels representing edges is greater than 25 percent, then we decrease the threshold by 10 and reevaluate the percentage until less than 25 percent of the pixels represent the edges. In this way, lighter edges that are not essential to the understanding of what the image represents are removed. The results can be seen in Figure 4.6 and 4.7.

However, even the 5x5 Sobel operator struggles to handle JPEG images that have a lot of noise, since these edges are detected in the image, resulting in a larger file size due to the unnecessary edges. The adaptive method described above fails to distinguish between darker edges that are essential to the image and those that are a result of noise. Therefore, there is room for improvement in this edge detection method.

### 4.2.3 Gaussian Blurring

To counteract the effects of noise in the image on edge detection, we implement the technique of Gaussian blurring [14]. The objective of the blurring is to remove noise from the image by calculating a weighted average of the current pixel as well as the neighboring pixels, and replacing the current pixel with the result. This will proceed to remove sudden color changes that are generally isolated and that are not essential to the understanding of the image by using the average to lessen the effects of an outlier. We use a 5x5 Gaussian matrix, as calculated using the method described at [4] that uses integration over the following Gaussian equation rather than approximation:

$$G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$  \hspace{1cm} (4.2)

The coefficients are rounded to be integers for ease of calculation, and the standard deviation is set at the value of 1. Following this, we divide by the sum of the coefficients in order to obtain a weighted average. Below is the coefficient matrix, with the weights for the neighboring pixels listed below, and with the center square representing the current pixel:
Preliminary tests suggested that a 3x3 operator would not be sufficient, so we chose a 5x5 matrix as a compromise between functionality, risk of making the edges too wide, and concerns about the additional computational time required. We do not calculate the Gaussian blur for the outermost two rows and columns on each edge of the image,
Table 4.3: 5 x 5 Gaussian matrix [4]

<table>
<thead>
<tr>
<th>1</th>
<th>4</th>
<th>7</th>
<th>4</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>16</td>
<td>26</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td>41</td>
<td>26</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>26</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

since the matrix would extend outside the image, but we do use those pixels in our other calculations as necessary.

Following Gaussian blurring, we perform the Sobel algorithm as described above. This results in a file size reduction to 19 percent of the original size, which is slightly more efficient than the Sobel-only solutions, and is arguably better than the primitive edge detection in that the quality of the images is much better, considering the difference of only 2 percent in the file sizes.

We also implement the adaptive cutoff method following the Gaussian blurring and Sobel algorithm to further reduce the file size. With this, we are able to reduce the file size to an average of 15 percent of the original size. In addition, this method produced images that contained the essential edges for all five of our test cases. We believe that the adaptive method works better in this case than in the Sobel-only case because the grayscale values have a higher probability of being edges rather than noise, since the pixels representing noise have been reduced. We show the differences in Figure 4.8 and 4.9.

Therefore, we believe this method is the best among the ones studied so far because of the smallest average resulting file size, combined with the usability of all of the output images.
4.2.4 Canny Edge Detection

We examined a few other edge detection methods for comparison. The first method we evaluated was Canny edge detection [18, 28]. This method builds on Gaussian blurring as well as Sobel edge detection, and adds the additional processing step of finding the
direction of the gradient, using the formula and the Sobel gradients calculated in both the horizontal and vertical directions:

\[
\theta = \tan^{-1} \left( \frac{\Delta y}{\Delta x} \right)
\]  

(4.3)

Following Gaussian, we use the technique described in [2] to determine the numerical difference in the direction of the gradient. We use the degree measure of the direction angle found and round to the nearest 45-degree increment. The next step is to determine if a pixel is part of an image. We examine the two neighboring pixels on either side of the current pixel in the direction that we found in the previous step. If the current pixel’s magnitude of the gradient is larger than that of both of these neighboring pixels, then we consider it an edge and mark the pixel as black, or (0,0,0) in RGB space. Otherwise, we decide that this is not an edge, and we mark it as white, or (255,255,255) in RGB space.

Then, we use a simplified form of a hysteresis threshold to remove the minor edges. A hysteresis threshold has two numerical values, a higher value and a lower value. Any
pixel with a gradient magnitude below the lower threshold is considered to be a minor edge and is removed from the output image, and a magnitude above the higher threshold is considered to be a major edge. A pixel with a gradient magnitude in between the thresholds could be considered part of a minor or major edge, depending on whether this pixel is an essential part of another edge that is considered major. For our purposes, we look at the neighboring pixels and ensure that one of the neighboring pixels is also above the lower threshold to determine if it is a major edge. This choice preserves quality over file size, though it is possible to choose other methods of handling values in between the thresholds. More advanced methods recursively examine the neighbors of the neighboring pixels of the current pixel if the magnitudes of all the neighboring pixels fall in between the higher and lower thresholds.

Using this method with a lower threshold of 15 and a higher threshold of 30, we attain the average of 24 percent of the original file size. However, the average is not a fair representation of this method, because the median is 12 percent, and there is an outlier of 75 percent that skews the average. With the exception of the outlier, the file sizes are approximately the same, or in some cases slightly smaller, than those generated with the Gaussian and Sobel algorithms with the adaptive method as described in the previous section, as can be seen in Figure 4.11. It is possible that with further threshold and/or algorithm adjustments the issue of the outlier would be resolved, as there are many extra edges in the image that can be seen in Figure 4.10.

For simplicity, in our evaluations we will use the Sobel and Gaussian algorithms with the adaptive method, since this method consistently provides nearly optimal results, and because the Canny method typically generates files of approximately the same size. However, the Canny edge detection method clearly shows promise for future exploration.
4.2.5 Other Methods

Using median filtering, where the median of the adjacent pixels and the current pixel is calculated and then used to replace the value of the current pixel, is another option to correct image noise. However, in order to calculate the median of the matrix of pixels, a sorting algorithm must be used [77]. This would not be ideal for the application that we propose, where there are constraints on both computational time (since this is real-time video) as well as the power available to the underwater networking devices.

The other main category of edge detection methods is Laplacian methods, which uses the Gaussian blurring algorithm as the first step to remove noise from the image. Following this, the Laplacian operator uses the second derivative of the image to determine where the edges are. However, this makes the algorithm more vulnerable to the effects of noise, a problem that is prevalent in many remote sensing applications [7]. Therefore, we decided to focus on the other two methods in our design evaluation.
4.3 Evaluation

To evaluate our solution, we perform an experiment with a system that takes pictures, applies image processing to them, and then transmits them between computers using acoustic communications.

4.3.1 Implementation Details

To measure sustainability of acoustic communication, we perform data communication between two AquaSeNT acoustic modems [1] as depicted in Figure 4.12. The application allows for a set number of JPEG images to be transmitted over acoustic communication in a small fish tank with a base roughly 2 feet by 1 foot in size. The acoustic modem has a bandwidth of 14-20 kHz, as well as a data rate of 3200 bps. This provides an upper limit on the transmission rate and makes it difficult to transfer larger images with the bounded latency of real-time video streaming.

The framework does not use any familiar network protocols since the modem does not include such support. The only features supported are file size verification as well
Figure 4.13: Average file sizes for different image resolutions

as partial acknowledgment transmissions. In order to perform the transmissions, the image files were resized to be small enough to transmit over the acoustic medium. Before each transmission, both the sending and receiving buffers are flushed before 608 bytes are transmitted at a time. The program inserts delays into the transmission time in order to ensure that the receiver has enough time to receive and process the signals.

The images are captured through a webcam using the libav \texttt{avconv} package. The pictures are displayed upon successful transmission of the file to the receiving program using the GNU \texttt{display} utility.

4.3.2 Data Rate and Latency

The above system is estimated to have a transmission rate of 83.29 bytes per second, or 12.294 seconds per kilobyte (sec/kB), when a baud rate of 9600 bps is used. Although the transmission rate can go up to 400 bytes per second according to the specifications [1], we are conducting the test in a small water tank so the modems are inevitably affected by severe multipath fading due to tank wall reflections. Using this information
combined with the size of the typical image file, we can determine what effects image processing will have on transmission time and consequently, the ability to transmit real-time video.

For our purposes, we run our primary algorithms on 64 pixels by 48 pixels, 128x96, 320x240, and 640x480 size versions of our test images; the results are shown in Figure 4.13. While the file size varies depending on the contents of the original image, there are clear trends that reflect the choice of image processing method. As the bandwidth of an acoustic modem is very constrained, it is not feasible to send full-size images (taken by ordinary digital cameras); sending an unprocessed image would take a few hours in the worst case. Using our proposed Gaussian and Sobel method will reduce the transmission time up to twenty minutes. It is a large improvement but it is still unfeasible for real-time streaming.

We note that as the dimensions of the image decrease, the effectiveness of the grayscale, greenscale, and primitive edge detection algorithms increase, while the performance of the Sobel and Gaussian algorithms suffers. In 640x480 case, the primitive edge detection outperforms all the other algorithms by a small margin but the image
quality is often not sufficient. However, the Sobel and Gaussian algorithm combination consistently outperforms the remaining methods both in the image size reduction and the image quality, and outperforms the primitive edge detection method in the 64x48, 128x96 and 320x240 cases.

We also analyze the estimated time needed to transmit the images over the acoustic medium. In the 640x480 case, it takes about 15 minutes to transmit the unprocessed JPEG file from the sender to the receiver. Converting the image to grayscale reduces the latency to about 5 minutes, while Sobel reduces the latency to 3.5 minutes and adding Gaussian blurring results in 2 minutes. While this is a large improvement, this still does not support real-time video transmission. However, we note that with image sizes of 128x96 and 64x48, we are able to transmit the unprocessed images in 113 seconds and 81 seconds, respectively. With the Sobel and Gaussian combination, an image can be transmitted every 16 seconds at a resolution of 128x96, and every 10 seconds at a resolution of 64x48. While not real-time video grade, this rate still allows the user to interact in real time with the environment and, for instance to determine what the sensors are detecting. It is possible that differences in the baud rate, the distance between the sender and receiver, or in the specific model of acoustic modem may improve the latency of the data transmissions. For example, one possible way to get more frames per second is to use multiple bands for MIMO (Multi-Input Multi-Output) communication for faster overall transmission rate. Another way is to use the interframe prediction-based video compression techniques such as H.264, MPEG-2, Ogg, and Theora. We do not consider this in this paper but assume that if we only transmit the differences of continuous images, we can still send more images. Further improvements in our image processing techniques will continue to reduce latency by decreasing the file size that needs to be transmitted, while retaining the important details of the images.

To evaluate our proposed solution with existing solutions, we also perform simulation and measure elapsed time interval to deliver an image frame with different dis-
tances via QualNet. We set data rate to 9600 bps, fix packet size to 512 bytes and select image resolution = 128x96 pixels. We use recently proposed M-FAMA MAC protocol [44] and measure the latency. As shown in Figure 4.14, we note the general behavior, namely the delivery time of all solutions is proportional to distance. Required latency means the time required to deliver one packet from the AUV to destination to support real-time streaming. Gaussian and Sobel outperforms other existing solutions according to average time interval for image frame delivery. It is noteworthy that original image for 250m distance requires 20 seconds but our proposed Gaussian and Sobel only require 2.7 seconds.
CHAPTER 5

Conclusions

In this dissertation, we have proposed a new MAC protocol called M-FAMA designed for underwater acoustic streams. It allows a source to open multiple sessions to different receivers achieving temporal/spatial reuse and yet avoiding collisions by careful accounting of neighbors’ transmission schedules. It supports packet pipelining on the same link, with significant throughput improvement when only one node pair is active. M-FAMA is a greedy protocol that attempts to maximize throughput at the expense of fairness. To correct this tendency a fully distributed, low overhead Bandwidth Balancing control was introduced that guarantees stability and fairness in arbitrary topologies and traffic patterns. To further prevent congestion, a conservative version of M-FAMA was introduced, that restricts pipelining, with promising results in dense, heavy loaded networks with low propagation delays. Extensive simulation experiments have shown that M-FAMA outperforms the most popular underwater MAC protocols in representative streaming scenarios. Future work will include the study of a dynamic M-FAMA protocol that switches to conservative mode in specific load and topology conditions, and; the extension of M-FAMA to multicast applications and opportunistic forwarding.

We also investigated a hybrid solution involving both acoustic and optical communications as a step towards real-time video transmission for the underwater mobile sensor networks. The optical channel is “fragile” and intermittent (it can easily break up because of modems misalignment, mobility, water impurity etc). In the proposed hybrid system, when the optical channel quits, the acoustic channel takes over. To this end, the video signal must be compressed to such a level that the acoustic channel can...
deliver the video (or images) at a rate that allows real time interaction. We have examined the use of image processing as a compression technique in order to reduce the size of the data files. We have evaluated the feasibility of several image processing methods, and have shown that this is a viable technique that makes a significant impact on latency and bandwidth. We have proposed two possible solutions that reduce the images between 10 and 20 percent of the original file size and estimated the latency of the data transfer. Future work includes analyzing these methods to conform to the computational time and power constraints found in typical underwater networking devices. It is possible that further modifications to the files can be made in order to minimize both file size and computational time, by analyzing the critical points of the algorithm to eliminate bottlenecks and to even parallelize portions of the algorithm. In addition, work needs to be done regarding the extension of these techniques to video transmission such as H.264 or Theora (as opposed to JPEG files), as well as the evaluation of additional methods of image compression and processing.
REFERENCES


[37] EvoLogics GmbH. S2cr 18/34 underwater acoustic modem.


