Stochastic Medium Access Control for Underwater Acoustic Sensor Networks

A Dissertation Presented
by

Yu Han

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Engineering

Northeastern University
Boston, Massachusetts

June 2016
To my family.
Contents

List of Figures iv
List of Tables vi
Acknowledgments vii
Abstract of the Dissertation ix

1 Introduction 1
  1.1 Contributions .................................. 2
  1.2 Thesis Organization .................................. 5

2 Related Work 6
  2.1 Underwater MAC Protocol Design ......................... 6
  2.2 Randomized Power Control for MAC ...................... 7
  2.3 Selected Underwater MAC protocols for Comparison ........ 8

3 A Propagation-Delay-Aware Stochastic MAC Protocol 10
  3.1 Motivation .................................. 10
  3.2 The DAP-MAC Protocol Design ......................... 11
    3.2.1 The Protocol Overview .................................. 11
    3.2.2 The Utility-Optimization Framework .................. 12
    3.2.3 Neighbor Discovery and Compatibility Relation Update 14
    3.2.4 Data Transmission Strategy .......................... 15
    3.2.5 Mobility Support .................................. 16
  3.3 Theoretical Analysis .................................. 18
    3.3.1 Statistics on Compatibility Factor .................. 18
    3.3.2 Optimal Network Throughput .......................... 20
  3.4 Performance Evaluation .................................. 22
    3.4.1 Simulation Settings .................................. 22
    3.4.2 Simulation Results .................................. 24
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A Traffic-Adaptive Receiver-Synchronized MAC Protocol</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>4.1 Motivation</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>4.2 The TARS Protocol Design</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>4.2.1 The Protocol Overview</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>4.2.2 Receiver-Synchronized Transmission Scheme</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>4.2.3 The Utility-Optimization Framework</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>4.2.4 Q-value Estimation and Maintenance</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>4.2.5 Acknowledgment Sending Mechanism</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>4.2.6 Mobility Support</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>4.3 Performance Evaluation</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>4.3.1 Simulation Settings</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>4.3.2 Simulation Results</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>A Stochastic MAC Protocol with Randomized Power Control</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>5.1 Motivation</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>5.2 The SMARP Protocol Design</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>5.2.1 The Protocol Overview</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>5.2.2 Randomized Power Control</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>5.2.3 The Stochastic Transmission Framework</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>5.2.4 A Case Study</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>5.3 Performance Evaluation</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>5.3.1 Simulation Settings</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>5.3.2 Simulation Results</td>
<td>65</td>
</tr>
<tr>
<td>6</td>
<td>Testbed-based MAC Protocol Performance Evaluation</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>6.1 The Testbed and Protocol Implementation</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>6.2 Performance Evaluation</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>6.2.1 Experimental Settings</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>6.2.2 Experimental Results</td>
<td>77</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>81</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>An example showing multi-sending, where nodes A and B send packets to node C at the beginning of a slot and the two packets arrive at node C apart without collision due to the different sender-receiver distances.</td>
</tr>
<tr>
<td>3.2</td>
<td>An example star network with the receiver’s compatible graph. (a) The 5-sender 1-receiver star network topology. (b) The compatible graph of receiver c.</td>
</tr>
<tr>
<td>3.3</td>
<td>An example transmission timeline of DAP-MAC, where two senders A and B send packets to receiver C in slot $n-1$ and the receiver C replies an ACK packet with a piggybacked UPDATE packet in slot $n$.</td>
</tr>
<tr>
<td>3.4</td>
<td>$E[F]$ changes with the number of senders ($N$), the maximum transmission range ($d_{\text{max}}$) and the distance threshold ($\Delta$), under a typical set of underwater parameters ($N = 10$, $d_{\text{max}} = 750$ m and $\Delta = 160$ m).</td>
</tr>
<tr>
<td>3.5</td>
<td>Theoretical comparison of DAP-MAC and S-ALOHA in terms of channel access probability and packet success rate for the network shown in Figure 3.2a.</td>
</tr>
<tr>
<td>3.6</td>
<td>Star network: total number of successfully received packets and among them the number of packets by single and concurrent transmissions under the saturated condition ($\lambda = 1$ pkt/s).</td>
</tr>
<tr>
<td>3.7</td>
<td>Star network: packet sending rate, number of retransmissions per successfully delivered packet, throughput and packet end-to-end delay for the four protocols.</td>
</tr>
<tr>
<td>3.8</td>
<td>Star network: throughput performance for the four protocols with different data packet sizes (64 bytes, 128 bytes and 256 bytes) under the saturated condition ($\lambda = 1$ pkt/s), which is used to show the impact of the packet size to the network throughput.</td>
</tr>
<tr>
<td>3.9</td>
<td>Ad hoc network: three types of compatible graphs for the 25 overlapping star subnetworks, classified by the positions of receivers (corner, side or inner).</td>
</tr>
<tr>
<td>3.10</td>
<td>Ad hoc network: throughput and packet end-to-end delay for the four protocols.</td>
</tr>
<tr>
<td>3.11</td>
<td>Ad hoc network: four networks are partitioned from the original network and the new network consists of the groups of one corner receiver 25, two side receivers 10 and 22, and one inner receiver 7.</td>
</tr>
<tr>
<td>3.12</td>
<td>Ad hoc network: throughput performance between the new network and the original network, which is used to investigate the interference.</td>
</tr>
<tr>
<td>3.13</td>
<td>Mobility testing: throughput and number of retransmissions per successfully delivered packets under different guard time settings: 0 ms, 10 ms, 20 ms and 30 ms.</td>
</tr>
</tbody>
</table>
3.14 Mobility testing: variation of compatibility factor of an observation node with node moving speed of 1.7 m/s.

3.15 Mobility testing: (a) UPDATE sending latency of a node sending UPDATE packets since the compatibility relation changes. (b) Throughput comparison for the four protocols.

4.1 An example on receiver synchronization, where nodes A and B send packets to node C and node A has longer propagation delay than node B.

4.2 The setting of slot size and transmission phase.

4.3 Illustration of the $Q$-value estimation.

4.4 $Q$-value estimation. (a) Estimated $Q$-value; (b) Estimation accuracy.

4.5 Collision reduction by receiver-synchronized (RS) scheme, compared to the transmitter-synchronized (TS) scheme.

4.6 Traffic adaptive sending probability. (a) Optimal node sending probability; (b) Normalized throughput gain compared to the case using $p = 1/N$.

4.7 Star network: packet sending rate, number of retransmissions, throughput and packet delay.

4.8 Ad hoc network: throughput and packet delay.

4.9 Classification of nodes, where nodes are labeled from Class I to Class VI.

4.10 Traffic adaptation under heterogeneous data loads. (a) Node sending probability; (b) Throughput comparison for the homogeneous and heterogeneous data load cases.

4.11 Probability distribution of packet receiving start and end time within a slot.

4.12 Mobility testing: throughput and packet delay.

5.1 Number of allowable power levels with respect to the sender-receiver distances.

5.2 The optimal sending probability and the total packet success rate for the studied case.

5.3 Optimal sending probability comparison among LiSS, TARS, and SMARP.

5.4 Single-packet and capture success rates in the total successfully delivered packets.

5.5 Star network: packet sending rate, number of retransmissions, throughput and packet delay.

5.6 Power consumption per successfully delivered packet.

5.7 Ad hoc network: Node sending probability under saturated data loads.

5.8 Ad hoc network: Success rates by intended packets.

5.9 Ad hoc network: network throughput and packet delay.

5.10 Mobility testing: dynamics on number of power levels.

5.11 Mobility testing: network throughput and packet delay.

6.1 The performance testbed. The modems are connected to computers via serial ports and they communicate with each other through an underwater acoustic channel.

6.2 The Teledyne Benthos SM-975/976 smart modem.

6.3 Experiment setup. Three SM 975 acoustic modems are submerged to form a single-hop network with a line topology, where the distance between two adjacent modems is about 1m.

6.4 Packet delivery ratio.

6.5 Packet end-to-end delay.
List of Tables
Acknowledgments

PhD pursuing is a long journey that requires one’s commitment, endeavor, and courage to conquer all the obstacles and difficulties, while unveils the rare but astonishing scenery in return. As this journey is closing to its end, I feel proud of myself not only for the strong determination of starting this journey four years ago, but also for the efforts I have made and the achievements I have reached so far. This journey would not have been so wonderful without the help and support from the people to whom I would like to express my heartfelt and deepest appreciation.

First, I would like to show my greatest gratitude to my advisor, Prof. Yunsi Fei. I feel fortunate and honored to have her as my advisor, who is very knowledgeable and possesses her personality charm. This dissertation would not have been possible without her advice and assistance. She advised me throughout every stage of my research, from the motivation to the algorithm, and from the results to the writing. She spent great efforts to help me become an independent and strong researcher. Her knowledge, wisdom and passion strengthened my skills and inspired me to explore and solve more challenging problems. Moreover, her encouragement and caring helped me relieve anxiety and live through those difficult days. The knowledge that I learned from her in both technical and personal aspects would benefit my future career and my whole life.

My heartfelt thanks also go to my committee members: Prof. Edmund Yeh, Prof. Stefano Basagni, and Prof. Adam Ding. I took the Data Networks course taught by Prof. Yeh and was also a TA of his course in Fall 2015. His teaching has built a solid knowledge base for my research and his advice has inspired and strengthened both my technical and teaching skills. During my working on the NU MONET project, I received a lot of advice from Prof. Basagni on the underwater experiments and performance evaluation. His broad knowledge and guidance has made the results available, and moreover, amazing! Prof. Ding helped me solve the optimization problem in my research, and enlightened me on seeking the alternative heuristic solutions. His great passion and enthusiasm inspired me a lot.

I would also like to thank my colleagues and friends for their support, encouragement and assistance along this journey: Liwei Zhang, Bingnan Jiang, Pei Luo, Zhen Hang Jiang, Chao Luo, Xin Fang, Yashar M. Aval, Haofan Shi, Shan Lu, Zhongliang Chen, Jierong Cheng, Lili Wang, Ying Mu, and all my friends who cheered me on and helped me get here.

Finally, and most importantly, I would like to thank my family for their unconditional love and endless support. I would not have been the person I am today without the deepest love from my parents, Juhe and Qingjin. Their encouragement, nourished guidance, and unselfish support, has sustained me throughout this journey. Thanks to my husband, Jian, who always loves me and encourages me to keep going on this journey. Thanks to my 3-year-old little girl, Yueqi, for bringing me lots of happiness and comforting me after a long day’s work by hugging me and saying “Mommy, I love you.” I am so fortunate to have all of you with me on this journey. I am hoping that I will
become your solid base of support, just like what you have been doing, to me. I love you all.
Abstract of the Dissertation

Stochastic Medium Access Control for Underwater Acoustic Sensor Networks

by

Yu Han

Doctor of Philosophy in Computer Engineering
Northeastern University, June 2016
Dr. Yunsi Fei, Advisor

Underwater sensor networks (UWSNs), the enabling technology for a broad range of aquatic applications, rely on acoustic communication for long-range transmissions. However, underwater acoustic communication suffers from several adverse features, such as long propagation delay and low bandwidth, making the networking design for UWSNs much more challenging than their terrestrial counterparts. Medium access control (MAC), a crucial component in networking design, coordinates among multiple agents that share the same channel resource and is responsible to schedule packet transmissions efficiently and fairly. However, due to the long propagation delay, the traditional handshaking-based random-access MAC protocols are not channel efficient for UWSNs.

In this dissertation, we address the inefficiency issues in random-access MAC for UWSNs, using a stochastic sending probability-based approach. We propose three handshaking-free underwater MAC solutions targeting the same goal: high network throughput, low packet end-to-end delay, and robustness under dynamics and controlled implementation complexity. The three solutions are based on a common utility-based probability optimization framework, but with different design considerations and objective functions. We first leverage the feature of long propagation delay, often taken as negative, to improve the parallelism between multiple senders. Our proposed protocol, the Delay-Aware Probability-based underwater MAC protocol (DAP-MAC), characterizes the group compatibility relation, a proposed indicator for successful concurrent transmissions, and utilizes this relation in the stochastic optimization framework for the best transmission strategy. The drawback of DAP-MAC is it requires long time slots to accommodate concurrent transmissions. We reduce the slot size and explicitly resolve the unique spatial-temporal uncertainty issue in UWSNs in our proposed protocol, the Traffic-Adaptive Receiver-Synchronized underwater MAC protocol (TARS). In TARS, we consider the data queue status and design the throughput-optimal transmission strategy...
to be traffic adaptive, which is very suitable for mobile and traffic-varying UWSNs. Simulation results show that TARS achieves the highest throughput and lowest packet delay among the typical representative MACs (about 13% ~ 146% higher in throughput and 13% ~ 21% lower in delay than others in a mobile ad hoc network). For further throughput improvement, we consider the capture effect and proactively create power capture at the receivers in our proposed protocol, the Stochastic MAC protocol with Randomized Power control (SMARP). We design a randomized power control scheme based on the non-negligible difference in acoustic propagation attenuation, and include the capture success probability in the stochastic optimization framework. Simulation results show that SMARP yields better throughput and packet delay than TARS and other representative MACs (about 23% ~ 240% higher in throughput and 27% ~ 60% lower in delay than others in a mobile ad hoc network). Finally, toward the development of wireless networking for enabling marine observatory capabilities at the Northeastern University Marine Science Center, we implement the TARS protocol on our testbed. We conduct in-tank underwater experiments and evaluate the performance of the TARS protocol under a three-node single-hop network. Experimental results show that TARS achieves the highest packet delivery ratio with controlled low packet end-to-end delay than other representative handshaking-free underwater MAC protocols, and is very suitable for the actual implementation in UWSNs.
Underwater sensor networks (UWSNs), the enabling technology for a variety of aquatic applications, rely on acoustic communication at the physical layer for long-range transmissions \[1,\] \[2\]. However, underwater acoustic channel differs significantly from the radio-frequency (RF) communication channel used in terrestrial networks. As one of the most difficult communication channels in use \[3\], it suffers from the long and varying propagation delay, limited and unreliable bandwidth, high path loss, severe multipath, etc. All the adverse features pose many challenges in efficient underwater networking design, especially the medium access control (MAC), which resolves the contention among multiple agents sharing the same communication channel.

The sound propagation speed in water is approximately five orders of magnitude lower than the RF signals, and it changes with temperature, depth and salinity \[4\]. Such long and varying propagation delay makes efficient MAC design very challenging. In addition to the inherent temporal uncertainty which has been considered in RF-based terrestrial networks, underwater acoustic communication also features spatial uncertainty \[5\]. This is because when multiple senders are located at different distances to the intended receiver, their packets transmitted at different times may reach the receiver simultaneously and result in packet collisions. While in RF-based terrestrial networks, propagation delay and therefore distance difference is negligible. As a result, MAC protocols that are originally designed for RF-based networks are no longer suitable for UWSNs.

For example, coordination among sensor nodes to share the communication channel usually involves message exchange for channel reservation or consensus. A handshaking process is widely used in delay-negligible terrestrial wireless networks (e.g., IEEE 802.11). It requires the exchange of RTS (Request To Send) and CTS (Clear to Send) control packets prior to data packet transmissions to avoid collisions. However, such handshaking process is very costly for underwater sensor networks,
CHAPTER 1. INTRODUCTION

due to the large overhead caused by the long propagation delay. Further, as the RTS and CTS control packets reserve the channel access exclusively for a pair of nodes, their neighbor nodes hold off transmissions to avoid data collisions after they overhear the RTS or CTS packet. Such exclusive channel access pattern further exacerbates the already large propagation overhead, resulting in low channel utilization and large packet end-to-end delay. Though variations have been made to improve the performance of handshaking-based protocols in UWSNs \[6, 7, 8, 9, 10, 11, 12\], alternative approaches are being explored to further improve the channel utilization.

The relatively light-weight Aloha-type random access approach \[13\] is more attractive for UWSNs because of its smaller overhead without handshaking. However, for this type of approach, the spatial-temporal uncertainty significantly affects the network performance. For example, the time synchronization in slotted Aloha, which has been used in RF-based networks to handle the temporal uncertainty, loses its throughput gain in UWSNs with the additional spatial uncertainty, resulting in the same throughput as pure Aloha \[5\]. Recently, several solutions have been proposed to address this spatial uncertainty in UWSNs. Some work, such as \[5\] and \[14\], uses a guard band or an extra control packet for advance notification to reduce collisions. Other work, such as \[15\] and \[16\], uses a receiver-synchronization approach to reduce collisions by avoiding cross-slot packet receptions. However, all these protocols do not optimize the sending probabilities, and thus achieve low channel utilization under heavy traffic loads, like the Aloha protocol. The work in \[17\] and \[18\] both uses the stochastic approach to determine the best packet transmission strategy. However, the work in \[17\] requires accurate interference estimation, which is not trivial to obtain in time-varying underwater channels, and the work in \[18\] underestimates the number of packet collisions which decreases the network performance. A well-designed MAC protocol with high network throughput and low packet end-to-end delay is very desirable for UWSNs.

1.1 Contributions

In this dissertation, we address the inefficiency issues in random-access MAC for UWSNs, using a general stochastic sending probability-based approach. We propose three handshaking-free stochastic underwater MAC solutions targeting the same goal: high network throughput, low packet end-to-end delay, and robustness under dynamics and controlled implementation complexity. The three solutions are based on a common utility-based probability optimization framework, but with different design considerations and objective functions.

We first envision that the long propagation delay of underwater acoustic communications,
although often taken as negative, can be leveraged to improve the parallelism between multiple senders, i.e., transmitting data simultaneously so as to achieve higher channel utilization. The exclusive channel access requirement in the RF-based terrestrial networks is unnecessary for UWSNs. Motivated by the inefficiency of long handshaking and the feasibility of successful concurrent transmissions in UWSNs, we propose a Delay-Aware Probability-based underwater MAC protocol, DAP-MAC, which eliminates the handshaking process and utilizes concurrent transmissions to significantly improve the network throughput. Compatibility relation among senders based on their distances to the common receiver is derived at run-time. A utility-optimization framework uses this compatibility relation to determine the best transmission strategies for senders: a sender with a higher number of compatible senders has higher channel access probability because of higher packet success capability, and vice versa. Our extensive simulation results demonstrate that DAP-MAC achieves better network throughput than the following typical underwater MAC protocols: $p$-persistent slotted Aloha (S-ALOHA) \cite{ref13}, slotted FAMA (S-FAMA) \cite{ref6} and DOTS \cite{ref7}.

As a handshaking-free protocol, DAP-MAC has been shown to be very effective in improving the network throughput in UWSNs by considering the concurrent transmissions. However, it requires a long time slot (i.e., at least packet transmission time plus maximum propagation delay) to eliminate cross-slot interference and to characterize concurrent transmissions, which limits the throughput improvement. Thus, a stochastic MAC protocol with shorter time slots is more desirable. However, the unique spatial uncertainty in underwater acoustic communication \cite{ref5}, may lead to increased packet collisions, which significantly affects the performance of the random-access MAC. Aiming at addressing the spatial uncertainty issue, we propose a Traffic-Adaptive Receiver-Synchronized underwater MAC protocol, TARS, for higher network throughput. The slot length in TARS is about the packet transmission time. We adjust the packet transmission time (phase) in a slot, which is dependent on the sender-receiver distance, to align packet receptions for collision reduction. Both the sound propagation speed variation and node mobility are considered in setting the optimal transmission phase and the slot size. We also consider the data queue status in the utility-optimization framework, in addition to the packet interferences in a two-hop transmission range. Therefore, the optimal sending probability is traffic adaptive and can be achieved in a distributive manner, which is very suitable for mobile and traffic-varying UWSNs. We compare TARS with the following more advanced underwater MAC protocols: LiSS \cite{ref18} (better than slotted Aloha), M-FAMA \cite{ref8} (better than DOTS), DSH-MAC \cite{ref9} (better than slotted FAMA), and DAP-MAC. Our simulation results show that compared to these representative protocols, TARS is able to achieve better performance with higher network throughput and lower packet end-to-end delay.
CHAPTER 1. INTRODUCTION

Both DAP-MAC and TARS protocols assume that a packet transmission is successful if only one packet is received at each time and thus any overlapped receiving packets are treated as collisions. This is a conservative assumption with the possible existence of capture effect, i.e., a packet with the strongest receiving power captures the receiver and could be successfully decoded even in the presence of interferences by other overlapping packets [19]. Power capture happens only when there is significant difference in receiving powers of overlapping packets, caused by the difference in their channel propagation losses and the transmission power levels. In the third work, we make use of the capture effect in channel access for further throughput improvement. We propose a stochastic MAC protocol with randomized power control for UWSNs, called SMARP. We design a power control scheme by considering the non-negligible difference in acoustic propagation attenuation, and proactively create power captures at the receiver side to improve the network throughput. Fairness is maintained by randomly selecting the transmission power among a set of power levels. We adopt the same utility-optimization framework in TARS, but consider the capture success probability to determine the optimal transmission strategy. Simulation results show that SMARP achieves higher throughput and lower packet delay than TARS and other representative underwater MAC protocols.

Finally, we also make efforts toward the development of wireless networking for enabling marine observatory capabilities at the Northeastern University Marine Science Center, as a part of the NU MONET project [20]. In particular, we contribute to develop an underwater acoustic testbed and implement the TARS protocol on the testbed. We conduct in-tank underwater experiments. We evaluate the performance of TARS under a three-node single-hop network and compare it to two other representative handshaking-free underwater MAC protocols: slotted ALOHA [13] and LiSS [18]. Experimental results show that TARS achieves the highest packet delivery ratio with controlled low packet end-to-end delay than slotted ALOHA and LiSS. Our testbed-based performance evaluation also shows that TARS is very suitable for the actual implementation in UWSNs.

The protocols proposed in this dissertation cover various aspects in stochastic random-access MAC for the unique underwater acoustic communication channel, with a design goal of high network throughput and low packet end-to-end delay. Our major contribution is the intensive in-depth study to explore stochastic random-access MAC and tackle cognizant issues, which have not been investigated much for UWSNs. Moreover, all the proposed protocols are designed with the implementation issues considered. As our testbed-based performance evaluation of the TARS protocol indicates, our proposed protocols can be implemented with controlled complexity in real in-water experiments.
CHAPTER 1. INTRODUCTION

1.2 Thesis Organization

The remainder of this dissertation is organized as follows. Chapter 2 gives the literature review on underwater MAC protocol design and the randomized power control scheme. We then present our proposed protocols: DAP-MAC, TARS and SMARP, in detail in Chapters 3, 4 and 5 respectively. In Chapter 6 we evaluate the performance of the TARS protocol on our developed underwater acoustic testbed. Finally, we conclude the dissertation and highlight our future work in Chapter 7.
Chapter 2

Related Work

In this chapter, we review the MAC protocols that are designed for long-delay underwater sensor networks and the randomized power control for capture effect. Then we give a summary of the reference protocols that we select to compare with our proposed protocols.

2.1 Underwater MAC Protocol Design

In the past decade, there have been significant efforts on the design of underwater MAC protocols to cope with the adverse features of underwater acoustic communication channel [21, 22, 23, 6, 10, 11, 7, 8, 12, 24, 9, 5, 14, 15, 16, 17, 18]. These solutions can be divided into two categories: contention-free and contention-based MAC protocols. TDMA-based protocols, such as UW-FLASHR [21], ST-MAC [22] and STUMP [23], are the typical contention-free MAC protocols for UWSNs. However, they require centralized scheduling, which results in either excessive control overhead or sluggish responses to network dynamics.

Contention-based MAC protocols are more suitable for mobile UWSNs because of their better flexibility and responsiveness to varying traffic loads and changing network topologies. This type of solutions can be further classified into two categories: handshaking-based solutions and light-weight handshaking-free solutions. Most existing handshaking-based MAC solutions are variants of those originally proposed for terrestrial wireless networks [6, 10, 11, 7, 23, 8, 12, 24, 9]. In order to improve the channel utilization, some work schedules parallel transmissions utilizing the long propagation delay, such as APCAP [11], PDAP [10], DOTS [7] and M-FAMA [8], while some work adopts the packet train approach for each handshaking [12]. However, the handshaking procedure inherently exacerbates the already large propagation overhead, resulting in limited improvement on
CHAPTER 2. RELATED WORK

channel utilization. Some other solutions are receiver-initiated, such as RIPT [24] and DSH-MAC [9], which save the one-way handshaking delay. However, the timing of initiating a packet transmission is nontrivial to determine under dynamic networks with varying traffic conditions.

On the other hand, the light-weight random access protocols without handshaking, such as the simple Aloha [13], become attractive for UWSNs for their small overhead and potentially higher channel efficiency. However, due to lack of prior channel coordination and existence of spatial-temporal uncertainty, such type of solutions may suffer from high packet collisions. Several Aloha-based solutions are proposed to reduce packet collisions [5, 14, 15, 16]. A modified slotted Aloha is proposed in [5] by adding a guard band in slot to alleviate collisions, which only works well in short-range networks. In [14], the use of an advance notification control packet in Aloha is proposed to reduce collisions. However, the idle time spent in overhearing is a waste of channel bandwidth which also increases the packet delay. The work in [15] and [16] uses receiver-synchronized approach to reduce collisions in slotted Aloha. They both suffer from low throughput in a heavy-traffic condition.

There are also a few light-weight solutions that use the stochastic approach to find the best transmission strategy that improves the network throughput [17, 18]. The work in [17] requires the estimation of packet interferences, which is nontrivial for mobile and time-varying underwater channels. In [18], a utility-optimal scheduling scheme (LiSS) is proposed to achieve throughput optimization. However, it underestimates packet collisions caused by cross-slot receptions and channel dynamics.

2.2 Randomized Power Control for MAC

Using multiple power levels to improve the throughput of random-access MAC protocols by power capture was first proposed in [26] for the RF-based terrestrial networks. It divides users into two groups with each group assigned a fixed high or low power level. Though this method improves the throughput of slotted Aloha by 1.5 times, it suffers from unfairness in bandwidth sharing. Such issue can be solved by adding randomness to the power selection. The first work on randomized power control [19] considers uniform power selection for slotted Aloha. However, it does not consider the difference in propagation distances and assumes the same number of power levels for every node, which may not be the case in reality. Other work, such as [27, 28, 29, 30], focus on finding the optimal number of power levels or the power density function for slotted Aloha. None of the above work considers a stochastic probability-based random access setting, where the
randomized power control may affect the optimal transmission strategy. In addition, they are all proposed for the RF-based wireless networks.

In underwater acoustic network regime, power control has been considered for maximizing network connectivity [31] and for minimizing energy consumption [32]. It has also been used in the MAC protocols for collision reduction [33] and for concurrent transmission, such as UPC-MAC [34]. They all use handshaking for channel access.

2.3 Selected Underwater MAC protocols for Comparison

We carefully select a number of underwater MAC protocols, as the representative protocols in different design regimes, to compare with our proposed protocols. Theses comparisons aim at disclosing the features of our protocols and demonstrating how our protocols improve the network performance.

We choose the $p$-persistent slotted Aloha [13], slotted FAMA [6] and DOTS [7] to compare with DAP-MAC. Slotted Aloha serves as the handshaking-free baseline to show the advantage of dynamically assigned probabilities with concurrent transmissions considered in DAP-MAC. Slotted FAMA and DOTS are all handshaking-based protocols, where slotted FAMA is the first well-known protocol that adopts handshaking to UWSNs and DOTS considers opportunistic transmissions. We compare these two protocols with DAP-MAC to show how the system performance is impacted by handshaking as well as concurrent transmissions.

In TARS, we use LiSS [18], M-FAMA [8] (conservative mode) and DSH-MAC [9] to compare, which are more advanced underwater MAC protocols. LiSS is a stochastic sending-probability-based protocol, but its optimal sending probability is only determined by the network topology, i.e., the number of neighbors. We use it to compare with TARS to show how TARS’s traffic-adaptive sending probabilities with the receiver synchronization help to improve the network throughput. M-FAMA and DSH-MAC are more sophisticated handshaking-based protocols than DOTS and slotted FAMA. M-FAMA seeks for concurrent transmissions with collision control and DSH-MAC [9] reduces one-way of handshaking for higher efficiency. We compare TARS with these two protocols to demonstrate the advantages of stochastic transmission-based protocols in improving the network throughput and packet end-to-end delay.

We compare our SMARP protocol with UPC-MAC [34], LiSS [18], and TARS. UPC-MAC is a handshaking-based protocol which uses power control to achieve concurrent transmission. We compare SMARP with UPC-MAC to show the impact of handshaking on network performance. As
two stochastic handshaking-free protocols with no power control considered, LiSS and TARS are used to compare with SMARP to show how SMARP’s randomized power control scheme and the optimal sending probability by considering power capture help improve the network performance.
Chapter 3

A Propagation-Delay-Aware Stochastic MAC Protocol

3.1 Motivation

The long propagation delay of underwater acoustic communications makes the exclusive channel access requirement in the RF-based terrestrial networks unnecessary for UWSNs. A two-sender example is shown in Figure 3.1 where nodes A and B are sending packets to node C at the beginning of a slot in a time-slotted system. The distances between A and C and between B and C are $d_{A,C}$ and $d_{B,C}$, respectively. Assuming that packets are at length $L$ bits and the data transmission/decoding rate is $\omega$ bps, if the two distances satisfy the inequality below, the two packets sent by A and B at the same time can arrive at C apart without collision\footnote{1We do not consider any capture, i.e., any overlapping packet receptions are considered as collisions and all the involved packets are discarded.}:

$$\Delta_d = |d_{A,C} - d_{B,C}| = |(t_{p,A} - t_{p,B})\nu| \geq \frac{L}{\omega}\nu,$$  (3.1)

where $t_{p,A}$ and $t_{p,B}$ are the propagation delay of packets from node A to C and B to C, respectively, and $\nu$ is the speed of sound in water (approximately 1500 m/s). With typical UWSN parameters $L = 128$ bytes and $\omega = 10$ kbps, the distance difference threshold for concurrent transmission is $\Delta_{Thresh} = 153.6$ m, which is attainable for distributed long-range underwater sensor networks where the transmission range reaches 2 km. In this case, the requirement of exclusive channel occupation is too conservative for UWSNs, and the long propagation delay can allow multi-sending in a time slot to achieve higher channel utilization and system throughput. For multiple (more than two) senders,
Eq. (3.1) works for each two-sender pair. If a sender’s packet does not overlap with any other packets at the receiver, i.e., satisfying (3.1) for the sender and all other senders that transmit packets in the same time slot, the sender’s packet can be successfully received.

Figure 3.1: An example showing multi-sending, where nodes A and B send packets to node C at the beginning of a slot and the two packets arrive at node C apart without collision due to the different sender-receiver distances.

Motivated by the inefficiency of long handshaking and the feasibility of successful concurrent transmissions in UWSNs, we design a Delay-Aware Probability-based underwater MAC protocol, DAP-MAC. The proposed protocol is similar to an “Aloha” protocol with no handshaking, but each node’s channel access probability is dynamically optimized at run-time in a utility-optimization framework satisfying proportional fairness [35][36][37][38]. Proportional fairness means allocating the channel access to senders in proportional to their packet sending success capabilities. We show theoretically that the optimal channel access probabilities are determined by the compatibility relation of senders (which will be defined later), and the optimal throughput at the receiver can be achieved by a distributed method with small control overhead. The distributive nature of DAP-MAC is suitable for long-delay and power-hungry UWSNs, which enhances the robustness further in mobile UWSNs.

3.2 The DAP-MAC Protocol Design

3.2.1 The Protocol Overview

DAP-MAC is a time-slotted contention-based protocol without handshaking. A packet can only be sent at the beginning of a slot, and a successful data transmission is followed by an ACK
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

return. As shown in [39], time synchronization among distributed nodes is achievable in underwater wireless networks, and we assume nodes follow the same sloting pattern. The slot size $T_s$ is chosen as the maximum propagation delay ($T_{prop}$) plus the DATA packet transmission time ($T_{tx}$).

A node who has a packet to send transmits the packet to the packet’s destination according to its prescribed channel access probability, which is optimized at run-time within a utility-maximization framework. The channel access compatibility relation for a group, characterized by the distance difference between senders and their common receiver, is utilized to achieve parallel transmissions. We define that two nodes are channel access compatible to a common receiver if their packets can be received apart after the nodes transmit packets concurrently. Intuitively, senders with more compatible nodes should be given higher probabilities, and vice versa. By explicitly exploring concurrent transmissions and assigning optimal probabilities for channel access, DAP-MAC is able to efficiently utilize the channel in long-delay UWSNs. Implemented in a distributive manner, DAP-MAC also handles network mobility very well.

3.2.2 The Utility-Optimization Framework

DAP-MAC uniquely takes into account the group compatibility relation and adopts a utility-optimization framework to determine the optimal channel access probabilities. We start from utility optimization for a simple receiver-centered subnetwork, which consists of a common receiver and a set of senders. We will show that this approach requires only a little additional information exchange which is feasible for UWSNs, and it improves the system performance significantly.

For a set of senders $\mathcal{N}^c = \{1, \ldots, N\}$ of a receiver $c$, each sender $i$, $i \in \mathcal{N}^c$, is associated with a channel access probability $p_i^c \in \mathcal{P}_i^c = [0, 1]$, and an utility function $u_i^c : \mathcal{P}_i^c \times i_{i \in \mathcal{N}^c} \mathcal{P}_i^c \rightarrow \mathbb{R}$. The utility function $u_i^c$ can be represented using the throughput (i.e., packet success rate) achieved by sender $i$, which is characterized by the compatible graph in DAP-MAC, as defined below.

**Definition 1 (Compatible Graph)** In an undirected compatible graph $G^c = (V, E)$ with respect to node $c$, $V$ contains all $c$’s neighbor nodes, and for each edge $(u, v) \in E$ nodes $u$ and $v$ are compatible, for $(u', v') \notin E$ nodes $u'$ and $v'$ are not compatible.

We define the compatible set of node $i$, $C_i^c$, consisting of all its compatible nodes to receiver $c$ and itself, and the interference set $I_i^c$ as the complementary set of $C_i^c$, i.e., $I_i^c = \mathcal{N}^c \setminus C_i^c$. Further, we define the Compatibility Factor (CF) of node $i$ as the number of elements in set $C_i^c$, $|C_i^c|$, and the Interference Factor (IF) as $|I_i^c|$.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

As seen in Figure 3.1, two senders are compatible if the difference of their distances to the receiver is larger than a threshold. The threshold is defined as $\Delta_{\text{Thresh}} = (\frac{T_{\text{tx}}}{\nu} + \delta)\nu = (T_{\text{tx}} + \delta)\nu$, where $\delta$ is a small amount of time added to account for node mobility or fluctuation in propagation delay [40, 41]. As an example, Figure 3.2b depicts a compatible graph for a star network (shown in Figure 3.2a) with the receiver $c$ and 5 senders. With a distance threshold of 160 m, nodes 1, 3 and nodes 1, 4 are two compatible pairs. The compatible set of sender 1 with respect to receiver $c$ contains nodes 1, 3 and 4 with $|C_{1}^c| = 3$, and its interference factor $|I_{1}^c| = 2$. For an arbitrary compatible graph, we will show in Section 3.3 that the mean of compatibility factor for a receiver-centered subnetwork varies with the number of senders, maximum transmission range and distance threshold, and maintains a high value for a typical set of underwater network parameters. A high compatibility factor indicates a high chance of successful concurrent transmissions, which is the design basis for DAP-MAC.

Based on the compatibility relation, the packet success rate of sender $i$, as the throughput indicator, is given by:

$$u_i^c = p_i^c \prod_{j \in I_i^c} (1 - p_j^c),$$

which is the probability that when node $i$ transmits a packet with probability $p_i^c$, all nodes in node $i$’s interference set do not send packets in the same slot. Under the proportional fairness constraint [35], the optimal channel access probability vector $p^c = \{p_1^c, p_2^c, \ldots, p_N^c\}$ can be obtained by maximizing the objective function $U^c(\mathbf{p}) = \sum_{i \in \mathcal{N}^c} \log(u_i^c)$, with $p_i^c \in [0, 1], \forall i \in \mathcal{N}^c$. We have the following theorem.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

Theorem 1  For the utility function given by (3.2), the optimal proportionally fair channel access probability for node $i$ with respect to the receiver $c$ is given by

$$p_i^c = \frac{1}{1 + |I_i^c|}. \quad (3.3)$$

Proof: We omit the receiver index $c$ for conciseness. According to (3.2), the objective function $U(p) = \sum_{i \in \mathcal{N}} \log(p_i) + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{I}_i} \log(1 - p_j)$. Since $\log(p_i)$ and $\log(1 - p_i)$ are both strictly concave functions of $p_i$, $U(p)$ is strictly concave of $p$ and therefore has a unique global maximum. By taking the derivative of $U(p)$ with respect to $p_i$, we have

$$\frac{\partial U(p)}{\partial p_i} = \frac{1}{p_i} - \frac{|I_i|}{1 - p_i},$$

where $|I_i|$ is node $i$’s interference factor, also the number of terms $\log(1 - p_i)$ appearing in $U(p)$. With $\frac{\partial U(p)}{\partial p_i}|_{p^*} = 0$, we have $p_i^* = \frac{1}{1 + |I_i|}$. □

Theorem 1 establishes the optimal channel access probabilities for senders that achieve the maximum overall packet success rate (i.e., the throughput) at the receiver. The values of those optimal probabilities only depend on the interference factors of the senders. The higher the value of IF, the lower the access probability should be, due to a higher chance of packet collisions. In the case that no node is compatible with any other node, i.e., $|I_i| = N - 1$, the optimal access probability $p_i^* = 1/N, \forall i \in \mathcal{N}$, which is the established result for RF-based networks where the propagation delay is negligible. The result in Theorem 1 also implies that DAP-MAC can run in a distributive manner as long as the senders know the values of their IFs. We will show in Section 3.2.3 that in DAP-MAC, the values of IFs are broadcast by the receiver upon any change of its compatible graph, with very small communication cost.

3.2.3 Neighbor Discovery and Compatibility Relation Update

Nodes discover neighbors via periodically broadcasting HELLO messages, which are widely used in ad hoc networks for building local connectivity [41], in addition to received or overheard packets. When a packet is sent, the time stamp is also included in the header. When a node receives or overhears a packet, it can calculate the propagation delay (and also the distance) between itself and the packets sender. Each node maintains an active neighbor table which includes neighbor IDs and their distances to the node. Based on the neighbor table, each node, as a potential receiver, is able to set up the channel access compatible graph of its group, as described in Section 3.2.2.

A node updates its compatible graph whenever there is any change on the topology of the group, for example, a neighbor leaving, a new neighbor joining, or distance changes because of node movements. Upon detecting the compatibility relation change, the node has to notify its neighbors
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

(potential senders). We design a small control packet, UPDATE, for this usage. It contains the ID of the node (receiver) and the interference factor for each neighbor in the receiver’s group. The receiver transmits an UPDATE packet either by piggybacking to the ACK packet when an ACK is scheduled, or when the UPDATE packet is about to expire (the elapsed time since UPDATE is generated reaches an upper limit $T_{update}$).

3.2.4 Data Transmission Strategy

Each node is potentially a sender and a receiver, for different data traffic. Figure 3.3 depicts the transmission timeline in DAP-MAC with two senders (A and B) and one receiver (C).

As a sender, a node keeps a list of optimal channel access probabilities to all its neighbors (receivers). Whenever the node realizes a change on its interference factor to a neighbor via the neighbor’s UPDATE packet, it updates the new optimal probability to the neighbor according to Theorem 1.

When a sender has a DATA packet to send (the head-of-line packet in the data queue), it chooses the action of transmit or wait at the beginning of a slot according to the channel access probability to the packet’s receiver. After sending a DATA packet, the node keeps listening to the channel for an ACK packet from the receiver. A packet collision is detected when the node does not receive an ACK from the receiver within two time slots since it sends the packet, or the overheard ACK does not contain the sender’s ID. Then the sender initiates a retransmission according to the prescribed access probability in the following slots. The number of retransmissions is bounded by a maximum retransmission number ($L_{max}$) to avoid standstill of the data queue due to adverse communication links.

As a receiver, if a node successfully receives DATA packets (may be from multiple senders) in the current slot, it will send an ACK packet in the next slot immediately acknowledging reception of those packets. Other pending DATA packets are transmitted later after the ACK.

The pseudo codes for implementing the transmission strategy of a node (e.g., node $i$) in DAP-MAC are shown in Algorithm 1 where we use $curState$ and $preState$ to indicate the current state and the previous state of a node, respectively. Since a node can be both a sender and a receiver, it has three possible states: IDLE, WAITACK and ACK. A node is initialized with an IDLE state and is allowed to send a DATA packet only when it is in IDLE state. Whenever it sends a DATA packet, it switches to the WAITACK state to wait for an ACK packet arriving in the next slot. If it receives a DATA packet destined for it in either IDLE or WAITACK state, it changes to the ACK state and
Figure 3.3: An example transmission timeline of DAP-MAC, where two senders A and B send packets to receiver C in slot $n - 1$ and the receiver C replies an ACK packet with a piggybacked UPDATE packet in slot $n$.

prepares to send an ACK packet with a piggybacked UPDATE packet (if any) at the beginning of the next slot. After sending the ACK packet, the node will switch back to its previous state (IDLE or WAITACK). It is worth noting that we give higher sending priority to an ACK packet over a new DATA packet, to favor the links associated with the already transmitted but unacknowledged packets.

### 3.2.5 Mobility Support

DAP-MAC is adaptive to network mobility and can provide robust services in dynamic underwater environment. As shown in previous sections, nodes are capable of capturing the changes of distances between themselves and their neighbors, updating the compatibility relations of their groups, and notifying their neighbors of such changes by sending UPDATE packets.

In addition, when nodes move, the sender-receiver distance difference of two compatible senders may reduce, which possibly makes them incompatible after the change. Since the group compatibility relation is detected at one time and the receiver generates an UPDATE packet, it may take at most $T_{update} + T_{prop}$ time for the neighbor nodes to recognize this relation change and send packets with their new prescribed channel access probabilities. During this time, the two senders may still send packets using unnecessarily higher access probabilities, leading to more packet collisions. Therefore, in order to guarantee that senders use the correct channel access probabilities, we add a guard time $\delta$ to the distance difference threshold, $\Delta_{thresh}$, to further improve robustness. Assume
Algorithm 1: Transmission strategy for node $i$ in DAP-MAC

1: \textbf{Initialize:} $\text{curState} \leftarrow \text{IDLE}, \text{preState} \leftarrow \text{IDLE}, L_{\text{retx}} \leftarrow 0$

2: \textbf{while} in a time slot \textbf{do}

3: \hspace{1em} \textbf{if} at the beginning of the slot \textbf{then}

4: \hspace{2em} \textbf{if} $\text{curState} == \text{ACK}$ \textbf{then}

5: \hspace{3em} Send an ACK packet and piggyback an UPDATE packet (if any)

6: \hspace{3em} $\text{curState} \leftarrow \text{preState}, \text{preState} \leftarrow \text{ACK}$

7: \hspace{2em} \textbf{else if} $\text{curState} == \text{IDLE}$ \textbf{then}

8: \hspace{3em} Send the HOL packet (if any) for $j$ with probability $p_j^{i*}$

9: \hspace{3em} \textbf{if} packet is transmitted \textbf{then}

10: \hspace{4em} Set ACK timer to the end of the next slot

11: \hspace{4em} $\text{preState} \leftarrow \text{IDLE}, \text{curState} \leftarrow \text{WAITACK}$

12: \hspace{1em} \textbf{else if} received a packet in the slot \textbf{then}

13: \hspace{2em} Maintain the neighbor table

14: \hspace{2em} \textbf{if} compatibility relation change detected \textbf{then}

15: \hspace{3em} Update the compatible graph of my neighborhood

16: \hspace{3em} \textbf{if} there is an UPDATE packet pending \textbf{then}

17: \hspace{4em} Update the packet with the new compatibility relation

18: \hspace{3em} \textbf{else}

19: \hspace{4em} Generate a new UPDATE packet and set timer ($T_{\text{update}}$)

20: \hspace{2em} \textbf{if} this is a DATA packet for me \textbf{then}

21: \hspace{3em} Record the sender ID and send the packet to upper layer

22: \hspace{3em} Update any pending ACK otherwise generate a new ACK packet

23: \hspace{3em} $\text{preState} \leftarrow \text{curState}, \text{curState} \leftarrow \text{ACK}$

24: \hspace{2em} \textbf{else if} this is an ACK packet for me \textbf{then}

25: \hspace{3em} Succeed. Remove the HOL packet from queue and $L_{\text{retx}} \leftarrow 0$

26: \hspace{3em} \textbf{if} there is a piggybacked UPDATE packet \textbf{then}

27: \hspace{4em} Update the channel access probability according to (3)

28: \hspace{3em} \textbf{if} $\text{curState} == \text{WAITACK}$ \textbf{then}

29: \hspace{4em} $\text{preState} \leftarrow \text{WAITACK}, \text{curState} \leftarrow \text{IDLE}$

30: \hspace{2em} \textbf{else if} this is an UPDATE packet \textbf{then}

31: \hspace{3em} Update the channel access probability according to (3)
32: if timeout for receiving an ACK packet then
33: \( L_{\text{retx}} \leftarrow L_{\text{retx}} + 1 \)
34: if \( L_{\text{retx}} > L_{\max} \) then
35: Failed. Remove the HOL packet from queue and \( L_{\text{retx}} \leftarrow 0 \)
36: if \( \text{curState} == \text{WAITACK} \) then
37: \( \text{preState} \leftarrow \text{WAITACK}, \text{curState} \leftarrow \text{IDLE} \)
38: if timeout for sending an UPDATE packet then
39: Send the UPDATE packet immediately when channel is free.

The average node moving speed is \( \zeta \). The guard time can be set to \( \delta = \frac{2\zeta(T_{\text{update}} + T_{\text{prop}})}{\nu} \), where \( \nu \) is the sound speed in water.

3.3 Theoretical Analysis

In this section, we first study the statistical mean of the compatibility factor, a measure of the potential for concurrent transmission in DAP-MAC. Then we give the optimal network throughput of DAP-MAC to show how the optimal transmission strategy achieves higher throughput with the compatibility relation considered.

3.3.1 Statistics on Compatibility Factor

We consider the scenario that \( N \) senders are independently and identically distributed (i.i.d.) in \((0, d_{\max})\) with respect to a common receiver, where \( d_{\max} \) is the characterized maximum transmission range of the acoustic modem. Let \( f(x) \) be the probability distribution function (PDF) of the sender-receiver distances. We first focus on two senders and calculate the probability that they are compatible. Then we extend our analysis to multiple senders to obtain the mean of compatibility factor.

Let \( X_1 \) and \( X_2 \) be two random variables representing the individual distances between the two senders and their receiver, with \( f_{X_1X_2}(x_1, x_2) \) as their joint PDF. Then the cumulative distribution function (CDF) of their sender-receiver distance difference, i.e., \( Z = |X_1 - X_2| \), can be represented by \( F_Z(z) = P(|X_1 - X_2| \leq z) = \int_0^{d_{\max}} dx_2 \int_{x_2-z}^{x_2+z} f_{X_1X_2}(x_1, x_2)dx_1 \). As two senders are defined to be compatible with each other if and only if their distance difference to the receiver is larger than a distance threshold \( \Delta \), we can obtain the probability that the two senders are...
compatible by \[42\]

\[ P_c = 1 - F_Z(\Delta) = 1 - \int_0^{d_{max}} dX_2 \int_{x_2-\Delta}^{x_2+\Delta} f_{X_1X_2}(x_1, x_2) dX_1. \]  

(3.4)

This two-sender compatibility probability is also the probability that an observed sender has one compatible colleague from the rest of \((N - 1)\) senders. With the i.i.d. assumption of the sender-receiver distance distribution, the event that any sender is compatible with the observed sender is an independent event, which occurs with the probability \(P_c\). According to the definition of binomial distribution \[42\], the probability that \(m\) senders from the \((N - 1)\) senders are compatible with the observed sender follows the binomial distribution, \(B(N - 1, P_c)\). Such \(m\)-sender compatibility probability can be expressed by \[42\]

\[ P^m_c = \binom{N - 1}{m} (P_c)^m (1 - P_c)^{N-1-m}. \]  

(3.5)

Note that the term \(N - 1\) counts the number of senders (potential competitors) excluding the observed sender.

Recall that the compatible set for a sender includes the sender itself, we finally obtain the mean of compatibility factor for a set of \(N\) senders as

\[ E[CF] = 1 + \sum_{m=0}^{N-1} m \cdot P^m_c = 1 + (N - 1)P_c, \]  

(3.6)

where \(P_c\) is given in (3.4). The variance of the compatibility factor can be given by \(\sigma^2[CF] = (N - 1)P_c(1 - P_c)\).

We next take an example to evaluate the mean of \(CF\) numerically. Assume that the sender-receiver distance follows a uniform distribution, i.e., \(f(x) = 1/d_{max}, x \in (0, d_{max})\). According to (3.4), we can calculate \(P_c\) as \(P_c = (d_{max} - \Delta)^2/d_{max}^2\). Substituting \(P_c\) in (3.6), we obtain the mean of compatibility factor as

\[ E[CF] = 1 + \frac{(N - 1)(d_{max} - \Delta)^2}{d_{max}^2}. \]  

(3.7)

We see that the mean of \(CF\) varies with the number of senders \((N)\), the maximum transmission range \((d_{max})\) and the distance threshold \((\Delta)\). Figure 3.4 depicts how the mean of \(CF\) changes with each factor while the other two keep fixed. It can be seen that under some cases a sender can be compatible with many other senders. For example, given a typical set of underwater parameters with \(N = 10, d_{max} = 750\) m and \(\Delta = 160\) m, \(E[CF]\) equals approximately 5.5, indicating that a sender can successfully transmit its packets when transmitting simultaneously with another five
more senders. The higher compatibility factor, the higher probability that DAP-MAC is able to schedule successful concurrent transmissions. Our proposed protocol fully takes advantage of such parallel sending opportunities to efficiently utilize the channel bandwidth, and therefore is capable of increasing the network throughput significantly.

Figure 3.4: $E[F]$ changes with the number of senders ($N$), the maximum transmission range ($d_{max}$) and the distance threshold ($\Delta$), under a typical set of underwater parameters ($N = 10$, $d_{max} = 750$ m and $\Delta = 160$ m).

### 3.3.2 Optimal Network Throughput

In this section, we first analyze the optimal network throughput of DAP-MAC. Then, we compare DAP-MAC to $p$-persistent slotted Aloha (S-ALOHA) \[13\] to show how the optimal channel
access probabilities in DAP-MAC improve the throughput.

We consider a network composed by a couple of senders and a common receiver and assume a saturated condition where senders always have packets to send. With the optimal access probabilities obtained according to Theorem 1 and the definition of the utility in (3.2), the optimal network throughput in packets per second can be represented by

\[ S^* = \frac{1}{T_s} \sum_i u^*_i = \frac{1}{T_s} \sum_i p^*_i \prod_{j \in I_i}(1 - p^*_j), \] (3.8)

which is the sum of each individual throughput achieved by the senders normalized by the slot duration. The total packet success rate (\(\sum_i u^*_i\)), as part of the throughput, is contributed by two components: the success rate of single transmissions (\(P_{\text{single}}\)) and the success rate of concurrent transmissions (\(P_{\text{concu}}\)), which are expressed as

\[ P_{\text{single}} = \sum_i p^*_i \prod_{j \neq i}(1 - p^*_j), \] (3.9)

\[ P_{\text{concu}} = \sum_{|C_i| \neq 1} p^*_i (1 - \prod_{j \in C_i \setminus i}(1 - p^*_j)) \prod_{k \in I_i} (1 - p^*_k). \] (3.10)

The success rate of concurrent transmissions, \(P_{\text{concu}}\), counts all the packet success rates of such senders: 1) the sender has at least one compatible sender, i.e., \(|C_i| \neq 1\), and 2) at least one compatible sender transmits a packet simultaneously with the sender, i.e., \(p^*_i (1 - \prod_{j \in C_i \setminus i}(1 - p^*_j))\). Then the relation between the total packet success rate and its two components can be shown as \(\sum_i u^*_i = P_{\text{single}} + P_{\text{concu}}\). Further, considering that simultaneous transmissions by incompatible senders cause packet collisions, we can also obtain the packet collision rate by

\[ P_{\text{colli}} = \sum_i p^*_i (1 - \prod_{j \in I_i}(1 - p^*_j)), \] (3.11)

which can be further represented by \(P_{\text{colli}} = \sum_i p^*_i - \sum_i u^*_i\).

In order to investigate how the optimal channel access probabilities in DAP-MAC improve the network throughput, we first define two measured ratios: the concurrent ratio (\(r_{\text{concu}}\)) and the collision ratio (\(r_{\text{colli}}\)). The concurrent ratio is defined as the ratio between the success rate of concurrent transmissions (\(P_{\text{concu}}\)) and the total packet success rate (\(\sum_i u^*_i\)). The higher the concurrent ratio, the higher portion of concurrent transmissions in all successful transmissions. The collision ratio is defined as the ratio between the packet collision rate (\(P_{\text{colli}}\)) and \(\sum_i u^*_i\).

We then compare DAP-MAC to \(p\)-persistent slotted Aloha (S-ALOHA) with its optimal access probability setting of \(p = \frac{1}{N}\). As a quick note, considering the interference factor of
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

node \( i \), \(|\mathcal{I}_i|\), is bounded by \( N - 1 \), according to Theorem \[ \text{[1]} \] the optimal access probability for node \( i \) is \( p^*_i \geq \frac{1}{N} \), always higher or equal to the ones predefined in S-ALOHA.

We take the network shown in Figure 3.2a as an example. Figure 3.5 depicts the optimal access probabilities and the distribution of packet success rates achieved by the senders. It can be seen that by considering the group compatibility relation, the optimal probabilities in DAP-MAC are scaled in accordance with individual sender’s concurrent transmission capability, and the throughputs achieved by the senders are redistributed. We also find that the concurrent ratio \( (r_{\text{concu}}) \) of DAP-MAC is 0.3, approximately 1.7 times higher than S-ALOHA (0.18), while \( r_{\text{colli}} \) is about 13% higher than S-ALOHA. The slightly higher collision ratio for DAP-MAC is caused by its more aggressive transmissions. However, compared to S-ALOHA, the total network throughput of DAP-MAC is still significantly improved, about 1.2 times higher than S-ALOHA.

Note that our utility-optimization framework considers proportional fairness. If the objective function is represented by the true throughput without fairness constraint, i.e., \( \hat{U}(p) = \sum_i u_i \), it can be shown that its optimal transmission strategy is only to allow one of the existing compatible pairs to use the channel (e.g., nodes 1, 3 or nodes 1, 4 for the above example) and prohibit others from sending packets. Though the throughput for this case can be very high because of the exclusive channel access by a compatible pair, this is an unfair channel allocation which should be avoided in networking design. The proportional fairness considered in DAP-MAC allocates the channel access according to packet success rates of senders, which balances the throughput and fairness among competing senders.

3.4 Performance Evaluation

3.4.1 Simulation Settings

We evaluate our protocol through extensive simulations on the widely used network simulator OMNeT++-4.3 \[ 43 \] with INET-1.99.4 extension \[ 44 \]. We implement the underwater acoustic communication link in the physical layer. We use an error-free channel model where packets propagate in the channel with a speed of 1500 m/s and all lost packets are due to collisions and exceeding the maximum number of retransmissions (set at 10). We set the data transmission rate at 10 kbps, and the distance threshold \( \Delta_{\text{Thresh}} \) at 160 m. Data packets are generated following a Poisson process with an average rate of \( \lambda \). Unless otherwise mentioned, all data packets are 128 bytes long and the MAC header is 4 bytes. All results are obtained by averaging 30 runs with each
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

run lasting 2000 simulation seconds, and shown with 95% confidence.

We compare DAP-MAC with the following representative MAC protocols: $p$-persistent slotted Aloha (S-ALOHA) [13], slotted FAMA (S-FAMA) [6] and DOTS [7]. We use S-ALOHA as the handshaking-free baseline for comparison to show the advantage of dynamically assigned probabilities with concurrent transmissions considered in DAP-MAC. S-FAMA and DOTS are well-known underwater protocols which both use two-way handshaking. S-FAMA adopts the basic two-way handshaking in a time-slotted manner, while DOTS schedules parallel transmissions opportunistically via overhearing. We compare these two protocols with DAP-MAC to show how the system performance is impacted by handshaking as well as concurrent transmissions.

We evaluate the protocols under three types of networks: (a) A star network: shown in Figure 3.2a. We use it to demonstrate why DAP-MAC is able to improve the system throughput with concurrent transmissions in a probabilistic random access framework. (b) A static ad hoc network: we deploy 25 nodes in a 2000m $\times$ 2000m horizontal square area in a 5 $\times$ 5 grid. We use this network to evaluate the performance of MAC protocols in typical multi-sender multi-receiver UWSNs. (c) A mobile ad hoc network: we randomly deploy 25 nodes in the same square area and allow nodes to move within a maximum range of 100 m. The underwater mobility model we adopt is the kinematic model [45] for water currents composed of tides and eddies. We use this type of network to test the robustness of our protocol to network dynamics.

The following performance metrics are evaluated in the simulations: (a) Packet sending
rate: the number of packets sent per second, which is a measure of channel occupation. (b) Number of retransmissions per successfully delivered packet: a direct measure of collisions which includes all retransmissions. (c) Throughput: the total number of successfully received data packets per second, which reflects how the channel bandwidth is utilized in a given amount of time. (d) Packet end-to-end delay: the average time between a data packet’s release time to the MAC layer and the time when it is successfully delivered at the receiver. It consists of queuing delay, transmission delay, and propagation delay. Propagation delay is defined as the packet travel time in the underwater channel. It can be calculated by the time it takes for the first bit of a data packet to travel from the sender to the receiver. A well-designed MAC protocol should achieve high throughput with controlled packet delay.

3.4.2 Simulation Results

3.4.2.1 The Star Network

We first investigate how concurrent transmissions effectively improve the system throughput. We set $\lambda$ at 1 pkt/s as the saturated condition. Figure 3.6 presents the total number of successfully received packets by the receiver, and also among them the number of packets by single and concurrent transmissions. It is seen that the throughput improvement of DAP-MAC is mainly contributed by concurrent transmissions with the concurrent ratio $1.7$ times higher than S-ALOHA. By taking advantage of nodes’ locations and exploring the multi-sending opportunities, the throughput of DAP-MAC is greatly increased (also $1.2$ times higher). Note that the total number of received packets in experiments is $8\%$ lower than the theoretical value due to the channel usage by the control packets (e.g., ACK).

The performance metrics of the four protocols for the star network are evaluated in Figure 3.7. For all protocols the packet sending rates monotonically increase with $\lambda$ until leveling off from some points. S-FAMA is the worst in sending rates because of its exclusive channel access policy and the long handshaking delay. S-ALOHA is better without handshaking but suffer from limited channel access probability. DOTS shows a higher sending rate than DAP-MAC. However, as Figure 3.7b shows, the high sending rate exceeds what the network can handle which results in lots of packet collisions.

Figure 3.7b gives the number of packet retransmissions for each successfully delivered packet. As the packet generation rate increases, S-FAMA keeps low retransmissions because of more reliable transmissions after handshaking, while DAP-MAC demonstrates slightly higher retransmis-
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

Figure 3.6: Star network: total number of successfully received packets and among them the number of packets by single and concurrent transmissions under the saturated condition ($\lambda = 1$ pkt/s).

Fig. 3.7c and 3.7d present the throughput and packet end-to-end delay results, respectively. By keeping appropriate packet sending rates and relative fewer collisions, DAP-MAC is able to achieve the highest throughput among the four protocols, and the lowest packet delay at low data loads ($< 0.4$ pkts/s). S-FAMA has the lowest throughput due to low packet sending rate, but fewer collisions result in the smallest delay at high data loads. DOTS and S-ALOHA exhibit similar throughput performance and also similar delay (high). At high data loads, S-FAMA shows a bit lower delay than DAP-MAC. This is because when the channel bears high contention, the time spent in long handshaking is more rewarded by the reliable transmissions in terms of reducing the packet queuing delay. In this case, S-FAMA may be more suitable for the heavy traffic condition with delay-sensitive services.

We next evaluate the impact of data packet size to the network throughput of the four protocols. We vary packet size from 64 bytes to 256 bytes and set $\lambda$ to 1 pkt/s as the saturated condition. Note that the slot size in DAP-MAC also changes with packet size as packet size affects the packet transmission time. Figure 3.8 shows the throughput results. It can be seen that as the packet size increases, the throughputs of all protocols decrease due to longer packet transmission time.
Figure 3.7: Star network: packet sending rate, number of retransmissions per successfully delivered packet, throughput and packet end-to-end delay for the four protocols.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

and lower packet sending rate. At smaller packet sizes, for example, 64 and 128 bytes, DAP-MAC demonstrates significantly higher throughput than the other protocols. This is because given a fixed data rate, packets with smaller sizes require shorter packet transmission time, resulting in a higher opportunity of successful concurrent transmissions. DAP-MAC is capable of capturing such opportunity by optimally assigning higher access probabilities to senders who have more compatible nodes, so as to efficiently utilize the channel. In contrast, S-FAMA has the worst throughput due to its exclusive channel access policy, and S-ALOHA and DOTS are better than S-FAMA by scheduling limited concurrent transmissions. At a higher packet size (e.g., 256 bytes), longer packet transmission time reduces the chance of successful concurrent transmissions. In the studied network, no sender has compatible nodes and their optimal access probabilities equal \(1/N\), which is the channel access probability of S-ALOHA. Therefore, the two probabilistic protocols, S-ALOHA and DAP-MAC, achieve the similar throughput in this case.

Figure 3.8: Star network: throughput performance for the four protocols with different data packet sizes (64 bytes, 128 bytes and 256 bytes) under the saturated condition (\(\lambda = 1\) pkt/s), which is used to show the impact of the packet size to the network throughput.

3.4.2.2 The Static Ad Hoc Network

In a multi-sending multi-receiving ad hoc network, a node serves as both a sender and a receiver for different star subnetworks. The studied network, with nodes labeled from 1 to 25 row by
row in a $5 \times 5$ grid, can be viewed as 25 overlapping star subnetworks. Three types of compatible graphs, classified by the positions of receivers (corner, side or inner), characterize the compatibility relations of these subnetworks. Figure 3.9 shows the examples of these graphs, which represent the compatibility relations of corner node 1, side node 6 and inner node 7, and these overlapping star subnetworks consist of 4 networks with graph (a), 12 with graph (b) and 9 with graph (c). The optimal channel access probabilities, the mean of CFs and the packet success rate ($\sum u_i$) are also given for each graph (star network). We see that a receiver’s average CF can be as high as 5 in graph (c), which implies a high opportunity of successful concurrent transmissions that can be scheduled by DAP-MAC.

![Diagram of three types of compatible graphs](image)

(a) Corner.  
(b) Side.  
(c) Inner.

Figure 3.9: Ad hoc network: three types of compatible graphs for the 25 overlapping star subnetworks, classified by the positions of receivers (corner, side or inner).

We present the simulation results on the throughput and packet end-to-end delay of the ad hoc network in Figure 3.10, which shows similar throughput results as in Figure 3.7c. As for the packet delay, DAP-MAC achieves the smallest delay at lower data loads, and has similar performance with the other three at higher loads. S-FAMA is the worst in packet delay due to its long handshaking.

We next estimate the throughput upper-bound of the overlapping star subnetworks in the studied ad hoc network, assuming no inter-network interference among the subnetworks and that every node always has packets to send. According to (3.8), by summing all the normalized packet success rates ($u_i/T_s$) of the 25 subnetworks with each discounted by the different sending rates
of the senders, we obtain the throughput upper-bound for the studied network as approximately 7 pkts/s. It can be seen that the saturated throughput (4 pkts/s) achieved in the simulation, as shown in Figure 3.10a, is 40% lower than this upper bound. The reason for such throughput gap is due to the bandwidth usage by the control packets as well as the interferences caused by transmissions of nodes on the edges of the star subnetworks.

![Throughput and Packet end-to-end delay](image)

Figure 3.10: Ad hoc network: throughput and packet end-to-end delay for the four protocols.

We further examine these interferences by removing some network connections to reduce interference and observing the throughput change. We create a new network, as shown in Figure 3.11, which is composed of four disjoint star subnetworks. The four networks are partitioned from the original network and the new network consists of the groups of one corner receiver 25, two side receivers 10 and 22, and one inner receiver 7. Other nodes are all senders, and packets are destined to their respective receivers with the packet generation rate unchanged. Compared to the original network, in the new network only the links associated with the four receivers are kept, and hence there are no DATA-DATA interferences and fewer DATA-ACK interferences. We compare the throughputs achieved at the four receivers for the two networks in Figure 3.12. It can be seen that approximately 0.3 pkts/s (30%) throughput reduction in the original network is caused by the inter-network interferences, which include both DATA-DATA and DATA-ACK interferences between the four subnetworks and others.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

Figure 3.11: Ad hoc network: four networks are partitioned from the original network and the new network consists of the groups of one corner receiver 25, two side receivers 10 and 22, and one inner receiver 7.

Figure 3.12: Ad hoc network: throughput performance between the new network and the original network, which is used to investigate the interference.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

3.4.2.3 Mobility Test

We next examine the robustness of DAP-MAC under mobile underwater environment, where node mobility may cause link disruptions and result in data transmission failures. In the mobility testing, we set the update sending limit ($T_{update}$) at 10s (approximately 20 times longer than $T_s$) and $\lambda$ at 0.4 pkts/s by default. We adjust the node moving speed to control the link disruption time.

With node mobility, the distance variation between a sender and a receiver may result in a change on the group compatibility relation, and senders may not have the most updated compatibility information which leads to non-optimal channel access probabilities. In order to enhance robustness in the dynamic underwater networks, DAP-MAC adds a guard time interval ($\delta$) in the compatibility threshold ($\Delta_{thresh}$). The setting of the guard time is critical to system performance, as a shorter guard time interval promotes concurrent transmissions but may result in higher chance of packet collisions, while a longer guard time interval decreases the chance of unsuccessful concurrent transmissions but may lead to lower channel utilization. In Figure 3.13 we show the throughput and packet retransmissions with the guard time interval varying from 0 to 30 ms at the node moving speed of 1.7 m/s. It can be seen that a shorter guard time interval achieves a bit higher throughput but causes more packet retransmissions, and vice versa. We can choose a guard time interval of 20 ms to balance the throughput and packet retransmissions.

Figure 3.13: Mobility testing: throughput and number of retransmissions per successfully delivered packets under different guard time settings: 0 ms, 10 ms, 20 ms and 30 ms.

We next observe how the compatibility factors of a receiver’s neighbors change with time.
CHAPTER 3. A PROPAGATION-DELAY-AWARE STOCHASTIC MAC PROTOCOL

to show the opportunity of successful concurrent transmissions. We choose an observation node near the center of the network. We set the node moving speed at 1.7 m/s. As Figure 3.14 depicts, the average CF of the node’s neighbors is above 1.5 for all the time instants and may go as high as 5, which indicates large opportunities for DAP-MAC to schedule successful concurrent transmissions and increase the channel utilization.

Figure 3.14: Mobility testing: variation of compatibility factor of an observation node with node moving speed of 1.7 m/s.

To examine how a node reacts to the compatibility relation changes and how fast the change is broadcast to its neighbors, we measure the average delay of a node sending UPDATE packets since the compatibility relation changes. The results are shown in Figure 3.15a, which also includes the average relation change intervals. It can be seen that with the speed increasing, changes occur more frequently. Under all the speeds, the change can be quickly recognized and broadcast to neighbors within the time that is only a small portion ($< 1/5$) of the real change intervals, which shows DAP-MAC’s fast responsiveness to network dynamics.

The curves in Figure 3.15b show the system throughput achieved by the four protocols varying with the node moving speed. We see that the handshaking-based protocols (S-FAMA and DOTS) are more vulnerable to nodes movement, especially under higher mobility. This is because the two-way handshaking requires two propagation delays and therefore the probability of link disruptions in each round of data transmission is twice that of handshaking-free probability-based protocols. As the mobility becomes higher, link disruptions are more likely to happen. In contrast, the probability-based protocols schedule data transmissions much faster and they are less vulnerable
to dynamics. With the mobility support, DAP-MAC is able to schedule successful concurrent transmissions to achieve higher throughput over the other protocols.

Figure 3.15: Mobility testing: (a) UPDATE sending latency of a node sending UPDATE packets since the compatibility relation changes. (b) Throughput comparison for the four protocols.
Chapter 4

A Traffic-Adaptive Receiver-Synchronized MAC Protocol

4.1 Motivation

Spatial uncertainty, caused by the varying and low sound propagation speed and the sender-receiver distance difference, complicates the MAC protocol design in UWSNs. For example, in RF-based terrestrial networks, with the slot size equal to one packet transmission time ($T_{tx}$), the time synchronization in slotted Aloha eliminates the cross-slot packet receptions, which decreases the vulnerable period for packet collisions from $2T_{tx}$ to $T_{tx}$, and hence doubles the throughput of pure Aloha. In contrast, in UWSNs, due to the possible different propagation times of multiple senders, time synchronization in the sender side no longer eliminates the cross-slot receptions and the throughput of slotted Aloha will be decreased [5].

In this work, we focus on addressing the spatial-temporal uncertainty issue in random-access MAC of long-delay UWSNs. We propose a Traffic-Adaptive Receiver-Synchronized protocol, TARS, a stochastic light-weight channel access protocol that maximizes the overall network throughput. We explicitly handle the spatial uncertainty by adopting the receiver-synchronization approach, which adjusts the packet transmission time (phase) in a slot to align packet receptions for collision reduction. Significantly different from the previous work, we specifically consider the variation of sound propagation speed in water as well as the node mobility in dynamic underwater environment, and present the optimal settings of both the transmission phase and the slot size. Moreover, under such receiver-synchronization scheme, we formulate the network throughput in a queue-aware utility-
optimization framework, which accounts for not only the packet interference but also the data queue status to dynamically determine the best transmission strategy, i.e., the optimal sending probability profile. Such optimal transmission strategy is traffic adaptive and can be achieved in a distributive manner, which is very suitable for mobile and traffic-varying UWSNs.

4.2 The TARS Protocol Design

4.2.1 The Protocol Overview

We represent the network using an undirected graph $G = (N, E)$, where $N$ denotes the set of network nodes and $E$ is the set of all one-hop pairs (i.e., the edges). For a node $i \in N$, we denote $K_i$ as $i$'s neighbor set which includes all node $i$'s one-hop neighbors, and $|K_i|$ as the number of node $i$'s neighbors. Every node maintains a node sending probability, $P_i \in [0, 1], i \in N$, which is the probability that a node chooses to send a packet in a slot. The pending packets with different intended receivers are buffered in separate data queues, with each queue given a link sending probability, $p_{ij} \in [0, 1], j \in K_i$. The link sending probability is the probability that a node sends a packet to a particular receiver. It can be seen that $P_i = \sum_{j \in K_i} p_{ij}$. We define $\alpha_{ij} \in [0, 1]$ as the conditional link sending probability, satisfying the constraint $\sum_{j \in K_i} \alpha_{ij} = 1$. Then, we can represent $p_{ij}$ by $p_{ij} = \alpha_{ij} P_i$. We do not consider any capture, i.e., any overlapping packet receptions are considered as collisions and all the involved packets are discarded.

Our proposed protocol, TARS, uses the receiver-synchronized transmission approach to handle the spatial uncertainty with channel dynamics considered. A sender transmits a packet only at the transmission phase in a slot, which is sender-receiver distance dependent, to align packet receptions in single slots to reduce packet collisions. Any pending packet is sent with its optimal link sending probability, which is dynamically obtained in a utility-optimization framework subject to the proportional fairness constraint $[35, 38]$. The utility-optimization framework considers not only the packet interference but also the data queue status (i.e., the $Q$-values, which will be defined later) in maximizing the overall network throughput across all network links. Thus, the optimal sending probabilities are traffic adaptive, which can be achieved distributively with a small amount of queue status exchange. In addition, TARS is also integrated with a mobility support mechanism across the whole design to handle network dynamics. Therefore, TARS is suitable for distributive and mobile UWSNs.
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

4.2.2 Receiver-Synchronized Transmission Scheme

The receiver-synchronized transmission scheme changes the synchronization from the sender to the receiver. Nodes adjust their transmission phases to the slotting pattern such that their packets can be received within single slots. Ideally, by taking this approach, we can still have a modified slotted Aloha with doubled throughput in UWSNs. However, underwater channel dynamics, such as the sound speed variation [40], may violate the in-slot reception setting. TARS adopts such receiver-synchronized transmission scheme and considers the underwater channel dynamics to make the scheme suitable for UWSNs.

Figure 4.1 depicts an example of the receiver-synchronized scheme in TARS, where nodes A and B send packets to node C and node A has longer propagation delay than node B. It is shown that with the appropriate settings of the transmission phases with respect to the slotting pattern, i.e., \( \delta_{i,C}, i \in \{A, B\} \), all packets can be received within single slots and the vulnerable period for packet collisions reduces from twice the packet transmission time to the much smaller slot duration. We define the back-tracked slot, \( t_{ij}(r) \), as the sending slot of a packet sent from node \( i \) and received by \( j \) in slot \( r \). Figure 4.1 shows that with such transmission scheme, collisions only occur when packets are sent in their back-tracked slots with the same receiving slot, e.g., slot \( (r - 3) \) by node A and slot \( (r - 2) \) by node B. We see that by aligning packet receptions within single slots, the spatial uncertainty caused by the sender-receiver distance difference is eliminated.

![Figure 4.1](image)

Figure 4.1: An example on receiver synchronization, where nodes A and B send packets to node C and node A has longer propagation delay than node B.

We next describe how to set the transmission phase as well as the slot size, considering...
the sound speed variation in water. We will consider the node mobility in Section 4.2.6. Figure 4.2 depicts a pair of nodes with a fixed distance of $d_{ij}$, where a data packet is sent by the sender $i$ in the back-tracked slot $t_{ij}(r)$ and reaches the receiver $j$ in slot $r$. The transmission phase ($\delta_{ij}$), the changing propagation delay ($T_{\text{prop,min}}, T_{\text{prop,max}}$), the packet transmission time ($T_{\text{tx}}$) and the slot size ($T_{\text{slot}}$) are all indicated in the figure. Consider the sound speed underwater varies within the range of $(v(1-\beta), v(1+\beta))$, where $\beta$ is the maximum variation percentage, the propagation delay between a pair of nodes may change between $T_{\text{prop,min}} = \frac{d_{ij}}{v(1+\beta)}$ and $T_{\text{prop,max}} = \frac{d_{ij}}{v(1-\beta)}$.

![Figure 4.2: The setting of slot size and transmission phase.](image)

In order to make any packet received within a slot, we should firstly guarantee that the first bit of a received packet arrives after the beginning of a slot. Therefore, we set the transmission phase according to the lower bound ($T_{\text{prop,min}}$) of the propagation delay, which is given by

$$\delta_{ij} = T_{\text{slot}} - \text{mod} \left( \frac{d_{ij}}{v(1+\beta)}, T_{\text{slot}} \right),$$

where $\text{mod}(x, y)$ is the operation of $x$ modulo $y$. In addition, we also want to ensure that the last bit of the packet is received before the end of the same slot. Considering that the latest possible arriving time for the last bit since the packet is transmitted is ($T_{\text{prop,max}} + T_{\text{tx}}$) and the earliest possible ending time of the receiving slot $r$ is ($T_{\text{prop,min}} + T_{\text{slot}}$), we have the following inequality:

$$T_{\text{prop,min}} + T_{\text{slot}} \geq T_{\text{prop,max}} + T_{\text{tx}}.$$  (4.2)

By substituting the expressions of $T_{\text{prop,min}}$ and $T_{\text{prop,max}}$ in (4.2), we have $T_{\text{slot}} \geq d_{ij} \left( \frac{1}{v(1-\beta)} - \frac{1}{v(1+\beta)} \right) + T_{\text{tx}}$. Considering the maximum transmission range is $d_{\text{max}}$, we obtain the slot size which
guarantees the in-slot packet reception by

\[ T_{\text{slot}} = d_{\text{max}} \left( \frac{1}{v(1 - \beta)} - \frac{1}{v(1 + \beta)} \right) + T_{\text{tx}}. \] (4.3)

We see from (4.3) that the additional time beyond the packet transmission time \( T_{\text{tx}} \) in slot size serves as the guard band, which is designed to protect the in-slot packet reception setting under the sound speed variation. With typical UWSN parameters \( d_{\text{max}} = 750 \) m, \( v = 1500 \) m/s, \( \beta = 0.08 \) (see [4] and [40]), the fixed guard time is roughly 0.08 s. Note that a packet size of 256 bytes and a data rate of 10 kbps give a packet transmission time of 0.2048 s, which is about 2.5 times longer than the guard time. With longer packet and smaller data rate, the ratio of the guard time to packet transmission time becomes negligible.

Neighbor discovery in TARS is made through regular broadcast hello messages as well as the received and overheard packets. A node calculates the distances to its neighbors by measuring the propagation delays from the sending and receiving time stamps of the packets. The transmission phases to its neighbors are then determined and maintained in a local transmission phase table.

4.2.3 The Utility-Optimization Framework

In TARS, a node who has pending packets is allowed to send a packet in any slot with its pre-calculated transmission phase to the packet’s receiver. The decisions of whether to send a packet in a slot as well as which packet to send are made at the beginning of each slot, according to the optimal node and link sending probabilities, which are determined in a utility-optimization framework targeting at maximizing the overall network throughput. We first formally define the queue status indicator \( Q \)-value. Then we show how to obtain the utility-optimal transmission strategy.

TARS is a queue-aware protocol. It uses the empty queue probability to represent the individual queue status. We define the \( Q \)-value, \( Q_{ij} \in [0, 1] \), as the probability that node \( i \)'s data queue for node \( j \) is empty. Intuitively, the higher the \( Q \)-value, the fewer queuing packets for the associated receiver. Recall that \( \alpha_{ij} \) is the conditional link sending probability from node \( i \) to node \( j \). We have the following definitions: 1) if \( Q_{ij} = 1 \), then \( \alpha_{ij} = 0 \); and 2) if \( \sum_{j \in K_i} (1 - Q_{ij}) = 0 \), then \( P_i = 0 \). The second definition specifies the node sending probability to zero when all the data queues are empty. Note that the \( Q \)-value may vary with both the packet generation rate and the node sending probability vector \( P = \{ P_i \}, i \in N \). However, we found that \( Q \)-value is less dependent on the sending probabilities for the extremely low and high packet generation rates. To keep our protocol computationally feasible and tractable, we only consider the dependency of \( Q \)-value on the
packet generation rate. In this section, we assume that \( Q \) -values are known to the nodes. We will discuss how to obtain the \( Q \) -values in Section 4.2.4.

We first define the packet success rate of a link between sender \( i \) and receiver \( j \), \( u_{ij}(r) \), as the probability that a packet sent by node \( i \) is successfully received at node \( j \) in slot \( r \), which can be expressed as

\[
u_{ij}(r) = (1 - Q_{ij}) \alpha_{ij} P_{i}^{t_{ij}(r)} P_{0,j}^{r} \prod_{k \in K_{j} \setminus i} P_{0,k}^{t_{kj}(r)}.
\] (4.4)

The first term \((1 - Q_{ij}) \alpha_{ij} P_{i}^{t_{ij}(r)}\) is the probability that node \( i \) sends a pending packet to node \( j \) in the back-tracked slot \( t_{ij}(r) \) with \( P_{i}^{t_{ij}(r)} \) as the node sending probability in slot \( t_{ij}(r) \). The second term \( P_{0,j}^{r} \) is the idle probability for node \( j \) in slot \( r \). The third term \( \prod_{k \in K_{j} \setminus i} P_{0,k}^{t_{kj}(r)} \) is the probability that receiver \( j \)'s other neighbors are all not sending in their corresponding back-tracked slots, with \( P_{0,k}^{t_{kj}(r)} \) as the idle probability for neighbor node \( k \). A node keeps idle in a slot because either the node does not have any packet to send in the queues, or the node chooses not to send a packet according to the sending probability. Therefore, the idle probability can be expressed as \( P_{0,k}^{t_{kj}(r)} = 1 - (1 - \prod_{m \in K_{k}} Q_{km}) P_{k}^{t_{kj}(r)} \). It is shown from (4.4) that a packet sent from node \( i \) to node \( j \) is successful in slot \( r \) only when: 1) node \( j \) is not sending any packet in slot \( r \); and 2) all node \( j \)'s neighbors except node \( i \) do not send packets in their back-tracked slots. It should be noted that interference to a packet reception can still occur when one of node \( j \)'s neighbors sends a packet to a node other than node \( j \).

The node sending probability vector \( P \) may change over time due to network dynamics and data traffic variation. We assume that the node sending probabilities keep constant over a number of consecutive time slots (at least for the maximum propagation time period). This assumption is valid under a slowly changing network topology with short-term stable data demands. Then we can remove the slot index in (4.4) and obtain the packet success rate as follows:

\[
u_{ij} = (1 - Q_{ij}) \alpha_{ij} P_{i} P_{0,j} \prod_{k \in K_{j} \setminus i} P_{0,k}.
\] (4.5)

By summing up all the packet success rates of sender-receiver pairs in the network and considering the proportional fairness assignment, we obtain the total network throughput as the utility function by \( U = \sum_{j \in N} \sum_{i \in K_{j}, k_{1} - Q_{ij} \neq 0} \log u_{ij} \). The optimal node sending probability vector \( P^{*} = \{P_{i}^{*}\}, i \in N \), and the optimal conditional link sending probability vector \( \lambda^{*} = \{\alpha_{ij}^{*}\}, i, j \in N \),
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

should satisfy the following multi-variable optimization problem:

\[
\begin{align*}
\text{maximize} & \quad U = \sum_{j \in N} \sum_{\{i \in K_j \& 1-Q_{ij} \neq 0\}} \log u_{ij} \\
\text{subject to} & \quad \sum_{j \in K_i} \alpha_{ij} = 1, \quad i \in N, \\
& \quad 0 \leq \alpha_{ij} \leq 1, \quad i, j \in N, \\
& \quad 0 \leq P_i \leq 1, \quad i \in N.
\end{align*}
\] (4.6)

We observe that after taking the logarithm operation, the vectors \(P\) and \(\Lambda\) in (4.6) can be separated apart. Therefore, we can solve their optimal values sequentially. We first solve the optimal \(P_i\). By expanding the network throughput \(U\), we see that the terms which are related to a node’s sending probability are not associated with the sending probabilities of others. Thus it is expected that the optimal sending probability of node \(i\), \(P_i^*\), is independent of \(P_j^*, j \in N \setminus i\). To facilitate our analysis, we define three parameters \(O(i)\), \(I(i)\) and \(M(i)\) as follows:

Outgoing \(Q\)-sum: \(O(i) = \sum_{j \in K_i} (1 - Q_{ij})\), \hspace{1cm} (4.7)

Incoming \(Q\)-sum: \(I(i) = \sum_{j \in K_i} (1 - Q_{ji})\), \hspace{1cm} (4.8)

Outgoing \(Q\)-multi: \(M(i) = 1 - \prod_{j \in K_i} Q_{ij}\). \hspace{1cm} (4.9)

In the expansion of \(U\), there are the number of \(O(i)\) terms of \(\log P_i\) and the number of \(I(i) + \sum_{j \in K_i} I(j) - O(i)\) terms of \(\log(1 - M(i)P_i)\), which are the only terms that involve \(P_i\). By letting \(\frac{\partial U}{\partial P_i} = 0\), we have \(P_i^* = O(i)/\{M(i)[I(i) + \sum_{j \in K_i} I(j)]\}\), which is a feasible solution only when \(0 \leq P_i \leq 1\), i.e., \(O(i)/\{M(i)[I(i) + \sum_{j \in K_i} I(j)]\} \leq 1\). If \(O(i)/\{M(i)[I(i) + \sum_{j \in K_i} I(j)]\} > 1\), we have \(\frac{\partial U}{\partial P_i} > 0\), which indicates that \(U\) is a monotonically increasing function of \(P_i\) and in this case the optimal \(P_i\) should be equal to 1. To summarize, the optimal node sending probability can be obtained by

\[
P_i^* = \min \left\{ \frac{O(i)}{M(i)[I(i) + \sum_{j \in K_i} I(j)]}, 1 \right\}, \quad i \in N.
\] (4.10)

It is seen from (4.10) that \(P_i^*\) is not only determined by node \(i\)’s own parameters of \(O(i)\), \(I(i)\) and \(M(i)\), but also affected by the incoming \(Q\)-sum \(I(j)\) of neighbors. We take a complete graph network as an example. Assume a saturated scenario, where all nodes have high packet generation rates and their \(Q\)-values of the data queues (i.e., \(Q_{ij}\)) keep close to 0. According to (4.7)-(4.10), the
optimal node sending probability $P_i^*$ becomes $\frac{|K_i|}{|K_i| + \sum_{j \in K_i} |K_j|}$, which is consistent with the result shown in [38]. In addition to the already considered saturated condition, TARS also specifies the optimal sending probability for unsaturated conditions, which could be much higher than the sending probability in the saturated case to achieve higher network throughput (see Section 4.3.2.1).

Next we solve the optimal $\alpha_{ij}$. Since there are two sets of constraints that involve $\alpha_{ij}$, we use the more convenient Lagrangian method to get its optimal value. We introduce a new set of Lagrange multiplier $\gamma_i$, $i \in N$, and modify the utility function in the optimization problem (4.6) as $U' = U + \sum_{i \in N} \gamma_i (1 - \sum_{j \in K_i} \alpha_{ij})$. Note that such operation does not change the optimal values for both $P_i$ and $\alpha_{ij}$. By letting both $\frac{\partial U'}{\partial \alpha_{ij}} = 0$ and $\frac{\partial U'}{\partial \gamma_i} = 0$, we get the optimal conditional link sending probability by

$$\alpha_{ij}^* = 1 - \frac{Q_{ij}}{O(i)}, \quad i, j \in N. \quad (4.11)$$

Then the optimal link sending probability $p_{ij}^*$ can be given by $p_{ij}^* = \alpha_{ij}^* P_i^*$.

The vector $\Lambda^*$ distributes the link sending probabilities among a node’s data queues in a throughput-optimization way. Equation (4.11) indicates that the smaller the $Q$-value for a particular receiver (i.e., higher packet generation rate), the higher probability that a node chooses to send packets intended to this receiver. This optimal setting ensures that a large amount of data for specific receivers, for example, in the situation that an urgent event is detected by a node, can be quickly delivered, which decreases both packet delay and packet dropping rate due to queue overflow.

### 4.2.4 $Q$-value Estimation and Maintenance

We see from (4.10) and (4.11) that in order for a node $i$ to obtain the optimal probability vectors $P$ and $\Lambda$, it needs to know the following information related to $Q$-values:

- **Outgoing $Q$-values:** $Q_{ij}$, the $Q$-values for node $i$’s outgoing data queues;

- **Incoming $Q$-values:** $Q_{ji}$, the $Q$-values for node $i$’s neighbors’ data queues destined to node $i$; and

- **Neighbors’ incoming $Q$-sums:** $I(j)$, $j \in K_i$.

In TARS, each node maintains a $Q$-value table which keeps the outgoing and incoming $Q$-values related to its neighbors as well as the neighbors’ incoming $Q$-sums. We next discuss how a node acquires and maintains these $Q$-values.

The outgoing $Q$-values measure the empty queue probabilities of a node’s data queues to its neighbors, which are obtained through queue status estimation. Figure 4.3 illustrates the
estimation process of node \( i \) for the queue associated with node \( j \). We use a binary empty queue indicator, denoted as \( s_{ij}(t) \), to represent the queue status in slot \( t \), where \( s_{ij}(t) = 0 \) indicates a nonempty queue slot and \( s_{ij}(t) = 1 \) represents an empty queue slot. The time series of \( s_{ij}(t) \) can be viewed as stationary over a number of consecutive time slots under a slowly changing network and short-term stable data traffic. We use a moving window based estimation approach with each window containing \( n_W \) slots, where \( n_W \) can take its empirical value for a balance between accuracy and cost. At the beginning of a slot \( t \), node \( i \) counts the number of empty slots in the past \( n_W \) slots, i.e., \( n^t_{ij} = \sum_{r=t-n_W}^{t-1} s_{ij}(r) \), and calculates its exponentially weighted moving average \( \tilde{n}^t_{ij} \) by

\[
\tilde{n}^t_{ij} = \theta n^t_{ij} + (1 - \theta)\tilde{n}^{t-1}_{ij},
\]

where \( \theta \) is the smoothing factor. We adopt the exponentially weighted moving average approach in TARS for its accepted accuracy and computational feasibility for the simple queue status estimation. Then, the outgoing \( Q \)-value at slot \( t \), \( Q^t_{ij} \), can be obtained by

\[
Q^t_{ij} = \frac{\tilde{n}^t_{ij}}{n_W},
\]

which converges after a small number of time slots and is updated in the corresponding entry of the \( Q \)-value table.

The incoming \( Q \)-values and the incoming \( Q \)-sums are obtained by message exchange. We design a short packet, QINFO, for nodes to periodically broadcast its related \( Q \)-values (e.g., every 10 s). The packet QINFO includes the node ID \( i \), the outgoing \( Q \)-values for its neighbors, and its incoming \( Q \)-sum \( I(i) \). A neighbor who receives a QINFO packet from node \( i \) decodes any related \( Q \)-values, for example, its incoming \( Q \)-value from node \( i \) and the node \( i \)’s incoming \( Q \)-sum, and updates its own \( Q \)-value table.
4.2.5 Acknowledgment Sending Mechanism

TARS uses the group and delayed acknowledgment sending mechanism to reduce the transmission cost and further increase the channel utilization in the resource-constrained UWSNs.

As a receiver, a node who receives a packet does not necessarily send an acknowledgment (ACK) packet right away. Instead, it waits until it receives the maximum number ($N_{ack}$) of unacknowledged packets, or the time elapse since it receives the first unacknowledged packet reaches the maximum allowable waiting time ($T_{ack}$). In either case, the node generates a group ACK packet, which contains the receiver’s ID, the sender ID(s) and the packet serial number(s), to acknowledge all the unacknowledged packets. The ACK packet is scheduled to be sent immediately whenever the node becomes idle.

As a sender, a node removes the packet(s) from the data queue(s) upon receiving an ACK packet acknowledging the receptions of its packet(s). If the sender does not obtain acknowledgment for a packet within a certain period of time ($T_{timeout}$), it initiates a packet retransmission in the following time slots, according to the optimal link sending probability to the packet receiver, as described in Section 4.2.3.

4.2.6 Mobility Support

The protocol TARS is integrated with a mobility support mechanism to handle the dynamic underwater environment. As previously mentioned, nodes are capable of figuring out the distances to their neighbors through neighbor discovery and adjusting the individual transmission phases accordingly. Moreover, nodes can also update their related $Q$-values and the optimal sending probabilities through periodically broadcast QINFO packets.

In addition, in the case of node moving, the distance between a pair of nodes may decrease, which could make a packet arrive at the receiver prior to the start of a slot and violate the in-slot packet reception setting. Therefore, we add an extra guard time to the transmission phase shown in (4.1). Assume that the maximum node moving speed is $\zeta_{max}$. The additional guard in distance is given by $\Delta d_{max} = 2d_{max}\zeta_{max}/v$, where the factor 2 indicates the worst case when the sender-receiver pair moves toward each other which makes the distance even smaller. Then the transmission phase is adjusted as

$$\delta_{i,j} = T_{slot} - \text{mod}\left(\frac{d_{ij} - \Delta d_{max}}{v(1 + \beta)}, T_{slot}\right). \tag{4.14}$$

Note that the slot size in (4.3) is not affected by node mobility, as we already consider the maximum transmission range in the derivation.
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

4.3 Performance Evaluation

4.3.1 Simulation Settings

We extensively evaluate our proposed protocol, TARS, on our developed OMNeT network simulator. In the simulations, unless otherwise mentioned, all data packets are 256-byte long with additional 4 bytes for the MAC header, generated following the Poisson process with a total average rate of $\lambda$. We assume that the packets generated at a node are uniformly destined to its neighbors and are to be transmitted at the rate of 10 kbps. We set the parameters of $N_{ack}$ to 3 and $T_{ack}$ to 5 s. We obtain the results by averaging 20 runs with each lasting 2000 simulation seconds. All results are shown with 95% confidence intervals.

We simulate the following networks in the evaluation: (a) The star network: containing a group of $N$ senders and one sink with the maximum transmission range of 750 m. We use this network to evaluate the key features of TARS. (b) The ad hoc network: composed of 25 nodes deployed as a 5x5 grid in a 2000m x 2000m area, which is used to evaluate the protocol performance in the multi-sender multi-receiver scenario. We also examine the robustness of TARS under dynamic underwater environment using a random topology, where we adopt the kinematic mobility model for water currents composed of tides and eddies [45].

We evaluate the protocol using the following system performance metrics: (a) Packet sending rate: the number of packets sent per second, which reflects the level of channel occupation. (b) Number of retransmissions per successfully delivered packet: a measure of packet collisions. (c) Throughput: the total number of successfully delivered packets per second, representing the channel utilization in a given amount of time. (d) Packet end-to-end delay: the average time between a data packet’s release time to the MAC layer and the time when it is successfully delivered at the receiver, which consists of queuing delay, transmission delay and propagation delay. The first two metrics provide the intermediate results, and we expect a well-designed MAC protocol with high throughput and low packet end-to-end delay.

We compare TARS with three representative underwater MAC protocols: LiSS [18], M-FAMA [8] (conservative mode) and DSH-MAC [9]. LiSS is a stochastic protocol, where the optimal sending probability is only determined by the network topology, i.e., the number of neighbors. We use it to compare with TARS to show how TARS’s traffic-adaptive sending probabilities with the receiver synchronization help to improve the network throughput. M-FAMA and DSH-MAC are both handshaking-based protocols, where M-FAMA seeks for concurrent transmissions using the
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

traditional two-way handshaking and DSH-MAC [9] reduces handshaking by the receiver-initiated approach. We compare TARS with these two protocols to demonstrate the advantages of the stochastic transmission-based protocols in improving the network throughput and packet end-to-end delay.

4.3.2 Simulation Results

4.3.2.1 The Star Network

We first choose $N = 4$ senders and set their distances to the common sink to 500 m, 750 m, 500 m and 750 m, respectively. In the network, each sender only has one neighbor (the sink) within its transmission range. In this section, we assume a homogeneous Poisson packet generation scenario, where each sender has the same rate for packet generations.

We first examine the $Q$-value of a sender’s data queue by varying the packet generation rate as the data load. We set the window size $n_W$ to 50 time slots and the smoothing factor $\theta$ to 0.6. We see from Figure 4.4a that as data load increases, the estimated $Q$-value decreases. The reason is that at lower data loads, a new arrived packet can be quickly transmitted and depleted from the queue, because such packet generation rate is much smaller than the network capacity for packet delivery. Therefore, the data queue keeps empty most of the time. As the data load increases, the contention becomes higher and a new packet has to be enqueued to cope with the potential higher chance of packet collisions, which results in lower probability of an empty queue.

We use Normalized Mean Square Error (NMSE) as the metric to evaluate the estimation accuracy of the $Q$-value. NMSE is defined as $E[(\frac{Q^*_ij - Q^e_{ij}}{Q^*_ij})^2]$, where $Q^*_ij$ is the true $Q$-value and $Q^e_{ij}$ is the average of the estimated $Q$-values over the simulation period. It can be seen from Figure 4.4b that the NMSE keeps very low ($< 0.01$) for the rates lower than 0.9 pkts/s. For the rates higher than 0.9 pkts/s, the error increases a bit due to the limit window size in estimation. However, at such high rates, the true $Q$-value is extremely small (close to 0). According to (4.7)-(4.10), the optimal sending probability becomes a constant for such saturated condition. Therefore, the impact of the estimation error to the optimal sending probability and the network throughput is negligible.

We next evaluate how the receiver-synchronized (RS) transmission scheme in TARS reduces packet collisions. We vary the packet size from 128 bytes to 512 bytes and adjust the slot size accordingly based on (4.3). We compare the RS scheme with the transmitter-synchronized (TS) scheme, where packets are sent at the beginning of a slot, with the same slot size as the RS scheme. Both schemes use the same sets of optimal sending probabilities determined by (4.10)-(4.11). It can be seen from Figure 4.5 that by handling the spatial uncertainty, the RS approach achieves lower
Figure 4.4: $Q$-value estimation. (a) Estimated $Q$-value; (b) Estimation accuracy.
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

packet collisions than TS, because of the decreased vulnerable period. Further, Figure 4.5 also shows that their difference in the number of collisions increases with the increasing packet size. This is because the time added in slot for sound speed variation also serves as the guard time for collision avoidance in the TS scheme. As the packet size increases, the percentage of such time to the packet transmission time becomes smaller, and the collision reduction is more contributed by the alignment of received packets than the guard time.

![Figure 4.5: Collision reduction by receiver-synchronized (RS) scheme, compared to the transmitter-synchronized (TS) scheme.](image)

We then investigate how the optimal traffic-adaptive sending probability in TARS improves the network throughput. In the studied network, according to (4.10) and (4.11), all senders have the same optimal node sending probability for a given packet generation rate and the $\alpha_{ij}$ equals to 1 for all queues. Figure 4.6a shows the optimal node sending probability. It can be seen that as the data load increases, the optimal sending probability decreases and eventually converges to the lower bound of $1/N$, which is the optimal sending probability for $p$-persistent slotted Aloha under the saturated condition [13].

The optimal sending probability setting is the design basis of our traffic-adaptive utility-optimization framework. When packets arrive sparsely (i.e., at lower data loads), a higher sending probability grants the nodes to access the under-utilized channel more aggressively yet with low chance of packet collisions, which effectively increases the network throughput. We define the normalized throughput gain as the throughput gain of TARS normalized by the throughput achieved
using \( p = 1/N \). We found that such throughput gain achieves its maximum at some middle range \((\lambda_l, \lambda_h)\) of data loads, e.g., \( \lambda_l = 0.4 \) pkts/s and \( \lambda_h = 0.8 \) pkts/s. The reason is that at the rates lower than \( \lambda_l \), the throughput gain is constrained by the data demands, while at higher rates the two sending probabilities converge to the same value. We vary the number of senders to investigate such throughput gain. The results are shown in Figure 4.6b with \( \lambda = 0.6 \) pkts/s. It can be seen that the more the senders, the higher throughput gain can be achieved. This is because with large number of senders, the sending probability of \( 1/N \) is rather low compared to the optimal sending probability in TARS, which results in an unnecessarily conservative channel utilization. In contrast, TARS considers the traffic demands and assigns a higher optimal sending probability to efficiently utilize the channel.

![Figure 4.6: Traffic adaptive sending probability. (a) Optimal node sending probability; (b) Normalized throughput gain compared to the case using \( p = 1/N \).](image-url)
Figure 4.7 shows the performance metrics of the four protocols for the 4-sender-1-sink star network. Note that the optimal sending probability obtained in LiSS for this topology is 0.2. As Figure 4.7a shows, the packet sending rates of all protocols monotonically increase with the packet generation rate prior to their saturation at some points. TARS keeps the highest sending rates at high data loads. At middle loads, LiSS achieves the highest sending rates, which, however, as Figure 4.7b indicates, include lots of packet retransmissions. DSH-MAC has the lowest sending rates due to the long handshaking and the exclusive channel access policy. M-FAMA is better than DSH-MAC because of scheduled concurrent transmissions. However, the long handshaking prevents it from scheduling packet transmissions in a timely manner.

(a) Data packet sending rate.

(b) Number of retransmissions.

(c) Throughput.

(d) Packet end-to-end delay.

Figure 4.7: Star network: packet sending rate, number of retransmissions, throughput and packet delay.
Figure 4.7b shows the number of retransmissions per successfully delivered packet. It can be seen that at low data loads, all protocols exhibit similar retransmissions. As the packet generation rate increases, the two handshaking-based protocols maintain the smallest number of retransmissions because of the more reliable transmissions after handshaking. TARS shows fewer collisions than LiSS because of the receiver-synchronized approach and the utility-optimal sending probabilities.

Figure 4.7c and Figure 4.7d give the network throughput and packet end-to-end delay. We see that among the four protocols, TARS achieves the highest throughput and the lowest packet delay at all data loads. LiSS is worse in throughput and delay due to its higher number of collisions and limited packet sending rates. DSH-MAC has the worst throughput (at high loads) and delay because of its lowest sending rates. M-FAMA performs better than DSH-MAC as a result of more aggressive packet sending contributed by concurrent transmissions.

4.3.2.2 The Ad Hoc Network

The 25-node ad hoc network offers a multi-sending multi-receiving scenario, where nodes may serve as senders or receivers for different data flows. Such network can be viewed as composed of 25 overlapping star subnetworks, where packet transmissions from adjacent subnetworks may cause interference that leads to packet collisions. We use this type of networks to evaluate the performance of TARS under the multi-sending multi-receiving scenario under both homogeneous and heterogeneous packet generations, as well as the mobile underwater conditions.

We first study the static ad hoc network. We consider homogeneous data loads where each node has the same average rate of Poisson packet generations. Figure 4.8 presents the simulation results of the throughput and packet end-to-end delay for such network. It can be seen that, as similar to the results shown in Figure 4.7, TARS achieves the highest throughput and lowest packet delay among the four protocols, because of collision reduction by the receiver-synchronized transmission strategy and the network-wide throughput optimization framework, which considers both the data demands and the packet interference among adjacent subnetworks. According to the node position as a sender in the subnetwork, the 25 nodes in the ad hoc network can be classified into 6 classes, as shown in Figure 4.9, where nodes are labeled from Class I to Class VI. Under homogeneous data loads, the nodes in the same class have the same optimal node sending probability.

We then demonstrate how TARS performs adaptively to the variation of data demands. We consider a heterogeneous data load scenario, where the packet generation rate of the Class VI node (the center node in Figure 4.9), is five times higher than the rates of others ($\lambda_0$). Figure 4.10a presents
CHAPTER 4. A TRAFFIC-ADAPTIVE RECEIVER-SYNCHRONIZED MAC PROTOCOL

Figure 4.8: Ad hoc network: throughput and packet delay.

Figure 4.9: Classification of nodes, where nodes are labeled from Class I to Class VI.
the optimal node sending probabilities for the 6 classes of nodes under both the homogeneous (i.e., all nodes are with $\lambda_0$) and heterogeneous data loads (i.e., $5\lambda_0$ for the Class VI node), where $\lambda_0$ is set to 0.05 pkts/s as the unsaturated condition ($\lambda = 1.25$ pkts/s for the homogeneous case). We also show the optimal sending probabilities in LiSS, which are only determined by the network topology and do not change with data demands.

It can be seen from Figure 4.10a that under the homogeneous data loads, TARS keeps higher optimal sending probabilities than LiSS for all classes of nodes and also achieves higher network throughput, as shown in Figure 4.10b. This is because TARS not only considers the network topology but also considers the ongoing data demands between a node and its neighbors so as to make a better decision on the sending probabilities. Under the heterogeneous condition, as the data load of Class VI node increases, TARS increases its sending probability and meanwhile decreases the sending probabilities of others to control packet collisions. In this way, as shown in Figure 4.10b, TARS is able to keep up with the increasing data demands and therefore improve its throughput by approximately 7% compared to the homogeneous condition. In contrast, LiSS fails to capture such sending opportunity and still achieves the relatively low throughput. M-FAMA and DSH-MAC achieve limited throughput gain compared to the homogeneous case due to the large handshaking overhead.

We next examine the robustness of TARS under dynamic underwater environment, where node mobility may cause link disruptions and result in data transmission failures. We randomly

Figure 4.10: Traffic adaptation under heterogeneous data loads. (a) Node sending probability; (b) Throughput comparison for the homogeneous and heterogeneous data load cases.
deploy 25 nodes in a 2000m x 2000m area, and let them move at an average speed of 2 m/s according to the kinematic mobility model. We set the maximum node moving range to 100 m.

As nodes are moving, packets may arrive at the intended receivers at variant times, challenging the in-slot packet reception setting in TARS. We first observe the distribution of the packet receiving time (start and end) in a slot by the intended receivers to show how the in-slot reception setting is still valid. As Figure 4.11 shows, most packets can be received within single slots. More specifically, we define a safe region, which is the region of the packet receiving ending time that guarantees the in-slot reception. Under the studied parameters, the region is roughly \([0.72, 1]\). It can be found that about 99.7% of packets are received within the safe region, indicating that the proposed receiver-synchronized approach works well under node mobility.

![Figure 4.11: Probability distribution of packet receiving start and end time within a slot.](image)

Figure 4.11: Probability distribution of packet receiving start and end time within a slot.

Figure 4.12 shows the performance of the protocols. With the mobility support, TARS is still able to achieve the highest throughput and lowest packet delay at all data loads. We also find that, under node mobility, the handshaking-free protocols (TARS and LiSS) provide more robust performance than the handshaking-based protocols (M-FAMA and DSH-MAC). In addition, as opposed to the results shown in Figure 4.8, M-FAMA exhibits worse throughput than DSH-MAC. The reason for these observations is that, for the handshaking-based protocols, the probability of link disruptions in each round of handshaking and data transmission is higher than the handshaking-free protocols, resulting in more handshaking and data transmission failures. As handshaking becomes longer (as in M-FAMA), link disruptions are more likely to happen. In contrast, the handshaking-free
protocols schedule data transmissions much faster, and thus are less vulnerable to network dynamics.

Figure 4.12: Mobility test: throughput and packet delay.
Chapter 5

A Stochastic MAC Protocol with Randomized Power Control

5.1 Motivation

All the above proposed work assumes that a packet transmission is successful if only one packet is received at each time and thus any overlapped receiving packets are treated as collisions. This is a conservative assumption with the existence of capture effect, i.e., when a packet with the strongest receiving power captures the receiver and is successfully decoded even in the presence of interferences by other overlapping packets [19]. Power capture happens only when there is significant difference in receiving powers of overlapping packets, caused by the difference in their channel propagation losses or the transmission power levels. Some prior work proposes to use randomized power control to improve the performance of slotted Aloha in RF-based wireless networks [19, 27, 28, 29, 30], without considering the effect of randomized power control on sending probabilities in a random access MAC.

In this work, we propose a high-throughput Stochastic MAC protocol with Randomized Power control for UWSNs, SMARP. The transmission power level is uniformly selected among a number of levels to maintain fairness, with each corresponding receiving power level being a multiple of the minimum acoustic intensity required for packet decoding. We adopt the basic queue-aware receiver-synchronized scheme in TARS. We formulate a new stochastic utility-optimization framework, which takes into account the capture success probability, and employ a distributive gradient-based algorithm to iteratively obtain the optimal sending probabilities.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

5.2 The SMARP Protocol Design

5.2.1 The Protocol Overview

We assume that nodes can be both senders or receivers for different types of traffic. As a sender, a node can send a new data packet or retransmit a data packet in each time slot based on its optimal link sending probability. As a receiver, a node is responsible for sending the acknowledging packet (ACK), which can acknowledge the receptions of multiple packets from multiple senders during an allowed period of time.

In SMARP, we use the same queue-aware receiver-synchronized approach in TARS for traffic adaptation and to solve the spatial uncertainty issue. However, SMARP differs significantly from TARS in that we consider power control and utilize it to take advantage of the capture effect to improve the network performance. Further, we formulate a new utility-optimization framework by considering power capture and provide the numerical optimal solutions, i.e., the best transmission strategy that maximizes the network throughput.

5.2.2 Randomized Power Control

Our randomized power control scheme is based on the underwater acoustic path loss model. Underwater acoustic channel is one of the most difficult communication channels in use [3], which suffers from high path loss, severe multipath, long and varying propagation delay, etc. The transmission path loss, $PL(d, f)$ [dB], for a signal with frequency $f$ (in kHz) traveling a distance of $d$ meters in an underwater acoustic channel, is characterized by [46]:

$$PL(d, f) = k \cdot 10 \log(d) + d \cdot a(f) + A_0,$$

(5.1)

where $k$ is the spreading factor with 1.5 as a typical value, $a(f)$ [dB/km] is the media absorption coefficient, with a multi-order empirical formula given by Thorp in [47], and $A_0$ [dB] represents the signal degradation due to other acoustic propagation effects, such as multipath and scattering [46].

Let $TVR$ denote the transmitter sensitivity of the underwater transducer, measured in dB re $\mu$Pa/V at 1m, and $V$ [V] denote the input voltage to the transducer, corresponding to the transmission power (in watts). The transmitted acoustic signal level ($SL_t$) (in dB re $\mu$Pa) can be obtained by $SL_t = TVR + 20 \log(V)$. Considering the lossy channel, we can obtain the received acoustic signal level by $SL_r = SL_t - PL$, and the signal to noise ratio (SNR) at the receiver by $SNR = SL_r - NL$, where $NL$ is the ambient noise level.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

It can be seen that under a given sender-receiver distance, the sender can control its transmission power so that the receiving SNR is higher than some target value ($SNR_0$) for packet decoding with a specified accuracy, assuming no existence of other overlapping packets. We denote $S_0$ (in $\mu$Pa) as such minimum receiving signal level that results in the target SNR. To simplify power selection, we use discrete transmission power levels, in that the receiving signal level $s(x_{ij})$ (in $\mu$Pa) for a packet $x_{ij}$ falls into the receiving signal level set $\Omega_{ij}$, which consists of the multiples of $S_0$, i.e., $S_0, 2S_0, \cdots, J_{ij}S_0$, for link $L_{i\rightarrow j}$ for power capture. We consider the SINR capture model used in [19], where the capture condition for node $i$’s packet $x_{ij}$ is expressed by

$$s(x_{ij}) \geq S_0 + \sum_{r \in \Phi^j_i} s(x_{rj}),$$

(5.2)

where $\Phi^j_i$ is the set of receiver $j$’s neighbors whose packets overlap with node $i$’s packet $x_{ij}$. It can be seen that if the receiving power of $x_{ij}$ is higher than the sum of receiving powers of other overlapping packets plus $S_0$, packet $x_{ij}$ is able to capture the receiver and be successfully decoded.

To maintain fairness among senders, the receiving signal levels in the set $\Omega_{ij}$ are chosen equally likely, i.e., with probability $1/J_{ij}$ for each power level.

We denote $\omega_{\text{max}}$ as the maximum transmission power specified by the UWSN modem. The number of allowable power levels for link $L_{i\rightarrow j}$, $J_{ij}$, is determined by the maximum receiving signal level, $s_{\text{max},ij}$, and the power step $S_0$:

$$J_{ij} = \left\lfloor \frac{s_{\text{max},ij}}{S_0} \right\rfloor,$$

(5.3)

where $s_{\text{max},ij}$ is the packet receiving signal level when it is transmitted with the maximum power $\omega_{\text{max}}$. As an example, with the following typical UWSN parameter settings: $A_0 = 5$ dB, $f = 15$ kHz, $NL = 60$ dB, $SNR_0 = 10$ dB, $\omega_{\text{max}} = 1$ W, and $TVR = 130$ dB as in [43], we show the number of levels under different sender-receiver distances in Figure 5.1. It can be seen that as distance decreases, the number of allowable power levels increases, as a result of more feasible transmission power options. Moreover, within the 100 m to 750 m range (the maximum transmission range), the number of levels can be as large as 10. As we will show in Section 5.2.4, a large number of power levels indicates a larger opportunity for capture success, and thus higher network throughput.

In SMARP, nodes regularly broadcast hello messages for neighbor discovery and distance detection. All hello packets are transmitted with the maximum transmission power. Upon receiving a hello packet, a node calculates the distance to its neighbor by measuring the propagation delay from the packet sending and receiving time stamps. Based on (5.3), a node can obtain all the
allowable power levels to its neighbors. Such number of power levels will also be used in the throughput-optimization framework, which will be described in Section 5.2.3 for optimal sending probabilities.

5.2.3 The Stochastic Transmission Framework

In this section, we formulate our stochastic utility-optimization framework. We denote $u_{ij}$ as the packet success rate for link $L_{i\rightarrow j}$, which is the probability that a packet from sender $i$ is successfully received at receiver $j$. Under the receiver-synchronized approach, a packet can be successfully received, either because no other packets arrive in the same slot or the packet captures the receiver because of its dominating receiving power. The packet success rate can be represented by:

$$ u_{ij} = u^1_{ij} + u^c_{ij}, \quad (5.4) $$

where $u^1_{ij}$ and $u^c_{ij}$ are the single-packet success probability and the capture success probability with multiple overlapping packets, respectively.

A single-packet success for link $L_{i\rightarrow j}$ can be achieved only when the following three conditions are met: 1) node $i$ has an in-queue packet pending for node $j$ and sends it based on the link sending probability $p_{ij}$; and 2) node $j$ is not sending in the packet’s receiving slot; and 3) all node $j$’s other neighbors do not have packets arriving in the same slot. Recall that we denote the node sending probability and the conditional link sending probability as $P_i$ and $\alpha_{ij}$, respectively, the
The single-packet success probability for link $L_{i \rightarrow j}$ can be calculated by

$$u_{ij}^1 = (1 - Q_{ij}) \alpha_{ij} P_i P_{0,j} \prod_{l \in K_j \setminus i} P_{0,l},$$

(5.5)

where $Q_{ij}$ is the empty queue probability, and $P_{0,j}$ is the idle probability for node $j$, due to either no pending packets in node $j$’s queues or its choosing not to send a packet according to its node sending probability. The idle probability can be expressed by $P_{0,j} = 1 - M_j P_j$, where $M_j$ is the probability that node $j$ has at least one in-queue packet to send, expressed by $M_j = 1 - \prod_{m \in K_j} Q_{jm}$. Note that in (5.5) we assume that the node sending probabilities keep constant over a number of consecutive time slots (at least for the maximum propagation time period), which is a valid assumption under a slowly changing network topology with short-term stable data demands.

5.2.3.1 Capture Success Probability

We next derive the capture success probability, $u_{ij}^c$, under the case of multiple overlapping packets. We first obtain the probability that a packet on link $L_{i \rightarrow j}$ successfully captures a receiver $j$ by:

$$P(L_{i \rightarrow j} \text{ capture}) = \sum_{|\Phi^i_j|} P(L_{i \rightarrow j} \text{ capture} | \Phi^i_j) P_{L_{i \rightarrow j}}(\Phi^i_j),$$

(5.6)

where $P(L_{i \rightarrow j} \text{ capture} | \Phi^i_j)$ is the conditional capture probability for link $L_{i \rightarrow j}$ given that packets from senders in the set $\Phi^i_j$ overlap in a slot with node $i$’s packet, which can be calculated using the capture model in (5.2) (we will show this later). The term $P_{L_{i \rightarrow j}}(\Phi^i_j)$ is the probability distribution of the overlapping packets, which can be further expressed by $P_{L_{i \rightarrow j}}(\Phi^i_j) = \prod_{r \in \Phi^i_j} (1 - P_{0,r}) \prod_{l \in K_j \setminus \{i, \Phi^i_j\}} P_{0,l}$. It can be seen that (5.6) considers all the combinations of senders in receiver $j$’s neighborhood who may have packets arriving in the same slot with node $i$’s packet. Given that node $i$ sends a packet to receiver $j$ who is not busy transmitting a packet, we can obtain the capture success probability for link $L_{i \rightarrow j}$ by

$$u_{ij}^c = (1 - Q_{ij}) \alpha_{ij} P_i P_{0,j} P(L_{i \rightarrow j} \text{ capture}).$$

(5.7)

Let $n_j$ denote the number of overlapping packets in a slot at receiver $j$, i.e., $n_j = |\Phi^i_j| + 1$ (including node $i$’s packet), and $J_j$ denote the vector of number of allowable power levels for receiver $j$’s neighbors. We can see from (5.2) that for capture success, $n_j$ can take values between 2 and $\min(\max(J_j), K_j)$, because otherwise the condition in (5.2) never holds and thus no power capture could happen. We can expect that for a broad range of available power levels and for a large
neighborhood, the probability expression for (5.6) can be very complex. We observed that the capture success probability with more than two overlapping packets is much smaller than that with two overlapping packets (e.g., more than 80\% smaller). Therefore, to keep our protocol computationally tractable for UWSNs with controlled complexity, we only consider the capture scenario when two packets overlap in a slot (i.e., \( n_j = 2 \)) in the following analysis.

Consider the two overlapping packets are from nodes \( i \) and \( r \). According to (5.2), the conditional capture probability, \( P(L_{i\rightarrow j} \text{ capture} | r) \), can be expressed by:

\[
P(L_{i\rightarrow j} \text{ capture} | r) = P(s(x_{ij}) \geq (s(x_{rj}) + S_0) | i, r \text{ send})
\]

\[
= P(\tilde{J}_{ij} \geq \tilde{J}_{rj} + 1 | i, r \text{ send}),
\]

(5.8)

where \( s(x_{ij}) = \tilde{J}_{ij}S_0 \), \( s(x_{rj}) = \tilde{J}_{rj}S_0 \), and \( \tilde{J}_{ij}, \tilde{J}_{rj} \) are both positive integers taking values from \([1, J_{ij}]\) and \([1, J_{rj}]\), respectively. It can be seen that there are \( J_{ij}(J_{ij}-1)/2 \) combinations for \((\tilde{J}_{ij}, \tilde{J}_{rj})\) that satisfy \( \tilde{J}_{ij} \geq \tilde{J}_{rj} + 1 \), among a total of \( J_{ij}J_{rj} \) combinations. Considering that all receiving power levels are chosen equally likely, we can obtain the probability in (5.8) by

\[
P(L_{i\rightarrow j} \text{ capture} | r) = \frac{J_{ij}-1}{2J_{rj}}.
\]

(5.9)

By substituting (5.9) in (5.7), we can calculate the capture success probability for link \( L_{i\rightarrow j} \). Combining with (5.4) and (5.5), we obtain the following total packet success rate for link \( L_{i\rightarrow j} \), which considers both single-packet success and capture success probabilities:

\[
u_{ij} = (1-Q_{ij})\alpha_{ij}P_iP_{0,j} \prod_{l\in K_j\setminus\{i,j\}} P_{0,l} \left(1+\frac{J_{ij}-1}{2}\sum_{r\in K_j\setminus\{i,j\}} \frac{1-P_{0,r}}{J_{rj}P_{0,r}}\right).
\]

(5.10)

5.2.3.2 The Multi-Variable Optimization Problem

We are looking for the optimal node sending probability vector \( P^* = \{P_i^*\} \), and the optimal conditional link sending probability vector \( \Lambda^* = \{\alpha_{ij}^*\}, i, j \in N \), which maximize the total packet success rate (i.e., the network throughput), subject to proportional fairness \( \lbrack 38 \rbrack \). As in TARS, the
multi-variable optimization problem can be given as follows:

\[
\text{maximize } \quad U = \sum_{j \in N} \sum_{i \in K, \& \ 1 - Q_{ij} \neq 0} \log u_{ij} \\
\text{subject to } \quad \sum_{j \in K_i} \alpha_{ij} = 1, \quad i \in N, \\
0 \leq \alpha_{ij} \leq 1, \quad i, j \in N, \\
0 \leq P_i \leq 1, \quad i \in N.
\] (5.11)

where the vectors \( P = \{ P_i \} \) and \( \Lambda = \{ \alpha_{ij} \}, i, j \in N \). Compared to (6) in [49], we add one additional term \( u_{ij}^c \), the capture success probability, in (5.11), representing the throughput contribution by power capture. By expanding the cost function \( U \), we find that the vector \( \Lambda^* \) is not affected by the added term \( u_{ij}^c \), and thus takes the same optimal value as in TARS: \( \alpha_{ij}^* = \frac{(1 - Q_{ij})}{O_i} \), where \( O_i = \sum_{j \in K_i} (1 - Q_{ij}) \), i.e., the sending probability distribution among a node’s neighbors is only determined by the status of neighboring data queues.

However, the additional capture success probability, \( u_{ij}^c \), greatly affects the optimal node sending probability \( P_i^* \). Considering that the cost function \( U \) is the sum of \( \log u_{ij} \) for all potential network links, for \( P_i \in (0, 1) \), we can obtain the first-order partial derivative\(^1\) of \( U \) with respect to \( P_i \) as sum of two terms:

\[
\frac{\partial U}{\partial P_i} = f_1(P_i) + f_c(P_i, \bar{P}_i). 
\] (5.12)

Here the first term \( f_1(P_i) \) is related to the single-packet success probability, which can be represented by

\[
f_1(P_i) = \frac{O_i}{P_i} \cdot \frac{M_i(1 + \sum_{j \in K_i} I_j - O_i)}{1 - M_i P_i}, \quad (5.13)
\]

where \( I_i = \sum_{j \in K_i} (1 - Q_{ji}) \) is a parameter related to the \( Q \)-values of node \( i \)'s neighbors’ data queues destined for node \( i \). It can be seen that \( f_1(P_i) \) is only determined by node \( i \)'s sending probability \( P_i \) and the involved \( Q \)-values.

The second term in (5.12), \( f_c(P_i, \bar{P}_i) \), related to the capture success probability, can be expressed by

\[
f_c(P_i, \bar{P}_i) = \frac{M_i}{(1 - M_i P_i)^2} \cdot \sum_{j \in K_i} \Gamma_{ij}, \quad (5.14)
\]

\(^1\)As \( M_i \in [0, 1] \) and \( P_i \in (0, 1) \), \( 1 - M_i P_i \) cannot be zero and thus \( \frac{\partial U}{\partial P_i} \) is valid.
where $\bar{P}_i$ is the vector containing all the sending probabilities of node $i$’s two-hop neighbors (i.e., the neighbor’s neighbor). The term $\Gamma_{ij}$ can be further obtained by

$$
\Gamma_{ij} = \frac{1}{J_{ij}} \sum_{l \in K_j \setminus i} \left( 2 + \frac{J_{ij} - 1}{\sum_{r \in K_j \setminus l} M_r P_r J_{rj}(1 - M_r P_r)} \right).
$$

We see that (5.14) involves not only node $i$’s own sending probability $P_i$, but also the sending probabilities and queue status of node $i$’s two-hop neighbors. Therefore, it is not easy to obtain a closed-form solution for the optimal $P^*_i$ that satisfies $\frac{\partial U}{\partial P_i} |_{P_i = P^*_i} = 0$. In SMARP, we design an iterative gradient-based algorithm for nodes to obtain their optimal sending probabilities distributively, which is very suitable for the mobile and traffic-varying UWSNs. The algorithm is described below.

### 5.2.3.3 A Distributed Gradient-based Algorithm

In SMARP, a node (as a sender) is responsible for calculating its optimal node sending probability according to (5.12). The needed information, such as the $Q$-values, is obtained through message exchange. We design a short packet, INFO, for this usage. An INFO packet of sender $i$ contains the sender ID, the parameter $I_i$ and product of $M_i P_i$, and its neighbor IDs and their $Q$-values $Q_{ij}, j \in K_i$. This packet can be piggybacked to the regularly broadcast hello packets and transmitted using the maximum transmission power. When a node $j$ receives the value of $M_i P_i$ from node $i$, it can calculate and update $\Gamma_{ij}$ for its neighbors based on (5.15). Such information can be sent along with an ACK packet or be sent immediately upon reaching the maximum allowable time since the last update.

A node $i$ conducts an iterative gradient-based algorithm to find its optimal node sending probability, upon sensing any updates on the related $Q$-values or the values of $\Gamma_{ij}$ from its neighbors. The algorithm starts with an initial sending probability $P_i^0$. At each iteration $t$, the gradient $g_i(t)$ that moves towards the optimal value $P^*_i$ can be calculated by

$$
g_i(t) = \frac{\partial U}{\partial P_i} |_{P_i = P^*_i}.
$$

The new sending probability is updated using a gradient-descent method: $P_i^{t+1} = P_i^t + \beta g_i(t)$. The parameter $\beta$ represents the rate at which the algorithm moves towards the direction pointed by the gradient, which can be set to its empirical value to balance the trade-off between iteration time and convergence requirement. To make the sending probability feasible, after each update, $P_i^{t+1}$ is projected onto a convex set $[\delta, 1 - \delta]$ using the Euclidean projection, where $\delta$ is a small value (e.g., $10^{-15}$) which does not affect the optimal values of the probability. The gradient in (5.16) may...
need to be constrained to avoid unwanted $P_i$ oscillations. We use the maximum number of iterations ($t_{\text{max}}$) and the minimum norm ($d_{\text{p,min}}$) of the difference between new and old sending probabilities to determine the termination conditions.

5.2.4 A Case Study

In this section, we give an example to illustrate how SMARP is able to achieve higher network throughput. We consider a star network composed of 4 senders and one central receiver, where the sender-receiver distances are roughly equal. We assume a saturated condition, i.e., all senders always have packets to send. Under such case, all senders have the same number of allowable power levels ($J$) and also the same optimal sending probability to the receiver.

We first study the optimal sending probability in SMARP. For comparison, we also show the sending probability in TARS, where all packets are transmitted with the minimum allowable power and received with the power $S_0$. As shown in Figure 5.2a, both protocols have the same optimal sending probability for $J = 1$ when capture is unavailable. For $J \geq 2$, the optimal sending probability in SMARP is all higher than that in TARS. This is because by considering power capture in SMARP, the chance of packet success is higher. Therefore, senders can access channel more aggressively without hurting the network throughput. We also see from Figure 5.2a that the higher number of power levels, the higher optimal sending probability becomes.

![Graph showing sending probability and total packet success rate](image)

Figure 5.2: The optimal sending probability and the total packet success rate for the studied case.

We then investigate the achieved total packet success rate by SMARP. We compare the protocol with TARS and a variant of SMARP, named SMARP (non opt), which does not use the
optimal sending probability in SMARP but the one in TARS. We use them to show how the proposed randomized power control as well as the optimal sending probability in SMARP improve the throughput. The results are shown in Figure 5.2b. It can be seen that by using the randomized power control scheme, both SMARP and SMARP (non opt) achieve higher throughput than TARS. Further, SMARP is able to achieve approximately 8% higher throughput than SMARP (non opt), because of the optimal sending probability under the capture-considered optimization framework. Moreover, as the number of power levels increases, the throughput gain by SMARP becomes larger. This is because the total packet success rate is composed of two components: single-packet success rate and capture success rate. As \( J \) increases, the capture success rate increases and its contribution to the total packet success rate increases as well.

5.3 Performance Evaluation

5.3.1 Simulation Settings

In this section, we extensively evaluate our proposed protocol, SMARP, on our developed OMNeT network simulator. We implement the acoustic propagation channel and use the same acoustic parameters as in Section 5.2.2. The data packets with 256-byte payload are generated following a Poisson process with a total average rate of \( \lambda \), which are to be transmitted at 10 kbps. The hello packet with any piggybacked INFO packet is broadcast for an average of every 20 s. All results are obtained as the average of 20 runs with each lasting 2000 simulation seconds, and shown with 95% confidence intervals.

We use the following networks to evaluate the protocol: (a) A star network: containing 4 senders and one sink with the maximum transmission range of 750 m, used to evaluate the key features of SMARP. (b) An ad hoc network: composed of 25 nodes deployed as a 5x5 grid in a 2000m x 2000m area. We use it to evaluate the protocol performance in the multi-sender multi-receiver scenario, and also the robustness under dynamic underwater environment. We use two network performance metrics to evaluate the protocol: (a) Throughput: the total number of successfully delivered packets per second, representing the channel utilization in a given amount of time, and (b) Packet end-to-end delay: the average time between a data packet’s release time to the MAC layer and the time when it is successfully delivered at the receiver, which consists of queuing delay, transmission delay and propagation delay. A well-designed MAC protocol is expected to achieve high throughput and low packet end-to-end delay.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

We compare SMARP with the following representative underwater MAC protocols: UPC-MAC [34], LiSS [13], and TARS. UPC-MAC is a handshaking-based protocol which uses power control to achieve concurrent transmission. We compare SMARP with UPC-MAC to show the impact of handshaking on network performance. LiSS and TARS are both stochastic handshaking-free protocols with no power control considered. LiSS obtains the optimal sending probability only based on the network topology, while TARS can adapt to traffic variation and uses receiver synchronization to reduce packet collisions. For comparison, we assume that TARS and LiSS use the minimum allowable transmission power that achieves the same packet decoding accuracy as SMARP. We compare SMARP with them to show how the proposed randomized power control scheme and the optimal sending probability by considering power capture help improve the network performance.

5.3.2 Simulation Results

5.3.2.1 The Star Network

In the 4-sender-1-sink star network, we set the distances between senders 1 to 4 and receiver 0 to be 450 m, 750 m, 450 m, and 750 m, respectively. According to the acoustic parameter setting and Eq. (5.3), the allowable power levels for senders in SMARP are listed as follows: \( J_{10} = J_{30} = 3 \) and \( J_{20} = J_{40} = 1 \), where the transmission power set for nodes 1 and 3 is \{0.4 W, 0.6 W, 1 W\} and the transmission power for nodes 2 and 4 is 1 W.

We first study the optimal sending probability of SMARP. In the gradient algorithm, we set \( \beta = 0.001 \), \( t_{\text{max}} = 500 \), and \( d_{p,\text{min}} = 0.001 \). We observe that the algorithm typically takes less than 50 iterations for probability convergence. We compare the optimal sending probability of three stochastic handshaking-free protocols in Figure 5.3. It can be seen that the sending probability in LiSS keeps constant under all data loads, while the sending probabilities in TARS and SMARP decrease with the increase of data loads and eventually converge to their lower bounds. Such traffic-adaptive sending probability setting effectively increases the channel utilization. At lower data loads when packets arrive sparsely, a higher sending probability grants nodes to access the under-utilized channel more aggressively. As data loads increase, the sending probability decreases to control packet collisions. We also see that the sending probabilities in SMARP are higher than those in TARS, because of higher chance of packet success by power capture. Further, in SMARP, the longer-distance senders have higher sending probability than the shorter-distance senders, as a result of the optimal setting to maintain proportional fairness among senders.

We then investigate the packet success rates of single-packet and capture in SMARP. It can
be seen from Figure 5.4 that as the packet generation rate increases, both the single-packet success rate and the capture success rate monotonically increase prior to the saturation at some points. The ratio of capture success rate to the single-packet success rate increases as well and converges to the maximum ratio of about $1/3$. Such successful captures consist of two components: captures among two overlapping packets and among three overlapping packets. Note that captures among four overlapping packets could not happen for the studied network since the maximum power level is 3. We find that most of the successful captures occur under the scenario of two overlapping packets, which hold more than 80% of the total captures.

We next compare the performance of SMARP with the other three protocols. The results are shown in Figure 5.5. As the intermediate results, Figure 5.5a and Figure 5.5b show the packet sending rate and the number of retransmissions per successfully delivered packet, which reflect the level of channel occupation and packet collisions, respectively. We see that SMARP has the highest sending rate at higher data loads and also the lowest packet collisions among the handshaking-free protocols, because of the higher sending probability and higher packet success by power capture. At middle loads, TARS and LiSS show higher sending rates than SMARP, however, as Figure 5.5b indicates, those packet transmissions include many collisions and retransmissions. UPC-MAC shows the lowest sending rate due to the long handshaking, but the more reliable transmissions after handshaking lead to the fewest retransmissions.

Figure 5.5c and Figure 5.5d give the network throughput and packet end-to-end delay.
It can be seen that SMARP achieves the highest throughput and the lowest packet delay among the four protocols under all data loads, as a result of appropriate sending rate and packet collisions. UPC-MAC is the worst in throughput due to the lowest sending rate, and LiSS is the worst in packet delay caused by the highest packet collisions. TARS is better than those two protocols because of the traffic-adaptive sending probability and collision reduction by receiver synchronization.

As higher transmission power may be used in SMARP for short-distance senders, we next compare the power consumption of the four protocols. We use the average transmission power consumption per successfully delivered packet as the metric, which includes all the power used for control and data packet transmissions. Figure 5.6 shows the power consumption under three types of data loads: light (0.9 pkts/s), medium (1.3 pkts/s), and high (2.9 pkts/s). It can be seen that under light data loads, TARS and SMARP keep the lowest power consumption. UPC-MAC is higher due to the additional power consumption in handshaking. LiSS is the worst because of its higher number of retransmissions per packet. As data loads increase, all protocols show increased power consumptions. SMARP consumes the least power under medium loads, because of higher packet success rate and lower transmission overhead. Under heavy loads, UPC-MAC shows the lowest power consumption as a result of low packet retransmissions, while TARS and SMARP are lower than LiSS and show similar performance in power consumption.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

Figure 5.5: Star network: packet sending rate, number of retransmissions, throughput and packet delay.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

5.3.2.2 The Ad Hoc Network

The studied ad hoc network, where each node can be both a sender and a receiver for different traffic, can be viewed as the composition of 25 star subnetworks, with each node as the subnetwork receiver. Due to the overlapping nature of the networks, a subnetwork may receive interference from adjacent networks, leading to reduced network performance. We will evaluate the performance of SMARP in such a multi-sender multi-receiver network.

We first study the optimal sending probability distribution in SMARP under a static network. We assume a homogeneous data load condition, where each node (as a sender) has the same total packet generation rate for neighbors. Under the network setting, all the senders have two power levels for transmission according to the locations of their neighbors. For a given packet generation rate, the optimal sending probability for a node is only determined by the node location, i.e., the number of its one-hop and two-hop neighbors. According to the sending probability, the nodes can be classified into 6 classes. Their probability density is depicted in Figure 5.7a, where a darker color indicates a higher sending probability. It can be seen that the center node (Class VI) has the lowest sending probability because its neighbors bear the highest traffic in the network. We then compare the optimal sending probabilities of SMARP with TARS in Figure 5.7b, under a saturated condition ($\lambda = 4.6$ pkts/s). We see that the sending probabilities in SMARP are higher than TARS for all classes of nodes, because of improved packet success rate by using power capture.

Unlike the single star network, there exists inter-subnetwork interference in an ad hoc network.
network, which is caused by the packet transmission from a subnetwork sender to its neighbor, who is the receiver of an adjacent subnetwork. The power capture considered in SMARP greatly mitigates the impact of such interference in that an intended packet may be successfully decoded even in the presence of interference from adjacent subnetworks. Figure 5.8 shows the intended packet success rates of single-packet and capture. It can be seen that under inter-subnetwork interference, SMARP is still able to make successful captures of intended packets, which occupy about 20% of the total successfully received packets.

Figure 5.8: *Ad hoc network*: Success rates by intended packets.

Figure 5.9 shows the network throughput and packet end-to-end delay of the four protocols. We see that SMARP achieves the highest throughput and the lowest packet delay under all data loads.
Compared to Figure 5.5 in the ad hoc network, UPC-MAC shows the highest packet delay, due to the longer time it takes for a successful handshaking under the interference from adjacent subnetworks.

We next investigate the robustness of SMARP under dynamic underwater environment. We use the kinematic mobility model for water currents composed of tides and eddies [45]. We set the node moving speed to 2 m/s and the maximum node moving range to 100 m. With nodes moving, the nodal distances may change with time. The protocol has to detect the change and determines the new number of allowable power levels for the optimal sending probability. We choose a pair of nodes around the center of the network who bear the most traffic, and show the dynamics on both the distance and the number of power levels in Figure 5.10. We see that as the distance decreases below a threshold (e.g., 460 m), SMARP is able to quickly detect the change and increase the power levels, to keep up with the increased capture opportunity.

Finally, we show the performance of network throughput and packet delay under mobility in Figure 5.11. The results are similar to those in Figure 5.9 except that UPC-MAC exhibits significantly higher packet end-to-end delay than the other protocols. As UPC-MAC relies on the long-time handshaking, the probability of link disruptions, due to mobility during a round of handshaking and data transmissions, is much higher than the handshaking-free protocols, resulting in increased packet retransmissions and longer packet delay.
CHAPTER 5. A STOCHASTIC MAC PROTOCOL WITH RANDOMIZED POWER CONTROL

Figure 5.10: Mobility testing: dynamics on number of power levels.

Figure 5.11: Mobility testing: network throughput and packet delay.
Chapter 6

Testbed-based MAC Protocol
Performance Evaluation

In the previous chapters, we have evaluated the performance of our proposed MAC protocols (DAP-MAC, TARS, and SMARP) based on the OMNeT++ network simulations. In this chapter, we conduct underwater experiments with acoustic modems and test the performance of the TARS protocol under practical underwater network settings. This effort is made as a part of the NU MONET project [20], toward the development of an underwater communication backbone and wireless networking for enabling marine observatory capabilities at the Northeastern University Marine Science Center.

6.1 The Testbed and Protocol Implementation

Our underwater network performance testbed is depicted in Figure 6.1. It consists of a number of acoustic modems and external computers controlling the modems to execute a protocol stack running on MATLAB. The modems are connected to computers via serial ports and they communicate with each other through an underwater acoustic channel.

We use the Teledyne Benthos SM 975 acoustic smart modems, which can be deployed underwater up to 6700 meters [50]. As Figure 6.2 shows, the modem electronics, batteries and a transducer are all housed in a vacuum sealed glass sphere with a burn wire release mechanism.

We use computers to control and configure modems via serial port connection, based on our developed protocol stack. The protocol stack is composed of three functional layers: the application layer, the MAC layer, and the physical layer (PHY) interface. The application layer is responsible for
Figure 6.1: The performance testbed. The modems are connected to computers via serial ports and they communicate with each other through an underwater acoustic channel.

Figure 6.2: The Teledyne Benthos SM-975/976 smart modem.
packet generations, according to a Poisson distribution with a variable average data generation rate (in packets per second). The MAC layer is the key layer of our testbed. We implement our proposed TARS protocol as well as two other representative handshaking-free protocols: slotted Aloha [13] and LiSS [18], for performance comparison. As time synchronization is needed for all the three protocols, a low-complexity time synchronization method is developed. We designate a node to be the network master, which holds the reference clock for the network and exchanges its timing information at the beginning of an experimental testing. Upon receiving the timing information, any other node adjusts its clock based on the estimated propagation delay to synchronize with the network master. In the MAC layer, we also implement the acknowledgment and retransmission process for all protocols. The PHY interface provides a direct interface to the modem, where the modem is controlled for packet transmissions and receptions and its working status is constantly monitored. The operating frequency of modems is 9-14 kHz, and the modulation is Multiple Frequency Shift Keying (MFSK) with incoherent detection.

In the slotted Aloha implementation, we set the slot size to be sum of the maximum propagation delay and the packet transmission time. Nodes send packets at the beginning of a slot whenever they have packets pending in the queue. Upon collision, a node will persistently retransmit the timed-out data packet with a probability of $1/N$, where $N$ is the number of nodes in the network, until either the packet is received successfully or the maximum allowable retransmissions (set to 3) is reached and the packet is discarded. The TARS implementation differs from that of the slotted Aloha in twofold. First, TARS uses a receiver synchronization approach, where nodes compensate for the propagation delay by sending data packets with transmission phases in a slot to guarantee no cross-slot packet receptions. Second, TARS uses an optimal adaptive transmission strategy (i.e., the optimal sending probability) that is dynamically determined by a throughput-optimization framework and adapts to data loads and network topology so as to increase the channel utilization while controlling packet collisions. The LiSS implementation differs from that of TARS in two ways. First, it uses the traditional transmitter synchronization approach as slotted Aloha, where packets are transmitted at the beginning of a slot. Second, its optimal transmission strategy is only determined by the network topology, without adapting to possible variation of the data traffic.
6.2 Performance Evaluation

6.2.1 Experimental Settings

We conduct experiments in an indoor rectangular water tank (filled with fresh water) with dimensions 240.2cm x 56.5cm x 73.1cm. We submerge three SM 975 acoustic modems to form a single-hop network with a line topology, where the distance between two adjacent modems is about 1m, as depicted in Figure 6.3. In this setup, each node is in the transmission range of every other node and can be both a sender and a receiver. We set the payload of each packet to 41 bytes with a data rate of 800 bps and all modems have the same packet generation rate.

Figure 6.3: Experiment setup. Three SM 975 acoustic modems are submerged to form a single-hop network with a line topology, where the distance between two adjacent modems is about 1m.

We compare the TARS protocol with the slotted Aloha and LiSS. We evaluate the performance of the protocols using the following two metrics: (a) **Packet delivery ratio**: the percentage of the packets successfully delivered to their destination, representing the channel utilization, and (b) **Packet end-to-end delay**: the average time between a data packet’s release time to the MAC layer and the time when it is successfully delivered at the receiver, which consists of queuing delay, transmission delay and propagation delay.
6.2.2 Experimental Results

Figures 6.4 and 6.5 show the packet delivery ratio and the packet end-to-end delay of the three protocols, respectively. We see that at low to medium traffic, TARS and LiSS are capable of delivering almost all of the data packets. TARS achieves the highest packet delivery ratio because of its optimized transmission strategy and collision reduction by interference alignment. LiSS shows lower packet delivery ratio than TARS (especially at medium traffic) due to its inability of adapting to varying traffic. Slotted Aloha exhibits the lowest packet delivery ratio at all traffic rates because of the higher number of packet collisions and retransmissions, which lead to a high number of discarded packets.

For the packet end-to-end delay, TARS shows better delay than LiSS at low to medium traffic, because of its more aggressive transmissions with controlled packet collisions. Slotted Aloha exhibits the smallest delay among the three protocols because we compute the delay only over successfully delivered packets and those few successful packets are observed to be often delivered at the first attempts (i.e., with small delays).

From the experimental results, we see that our proposed protocol TARS achieves the highest packet delivery ratio with controlled low packet end-to-end delay. In this chapter, we only show the final results (packet delivery ratio and delay) of the protocol performance. We will conduct more experiments to investigate some intermediate results, for example, the packet sending rate and packet retransmission rate. However, we can expect that TARS is able to maintain an appropriate packet sending rate (in between those of slotted Aloha and LiSS) and lower packet retransmission rate (and therefore fewer collisions) than slotted Aloha and LiSS.

In summary, TARS is able to achieve high network performance with controlled low complexity in implementation. Therefore, it is very suitable for the actual implementation in a single-hop or multi-hop underwater acoustic sensor network.
CHAPTER 6. TESTBED-BASED MAC PROTOCOL PERFORMANCE EVALUATION

Figure 6.4: Packet delivery ratio.

Figure 6.5: Packet end-to-end delay.
Chapter 7

Conclusion

In this dissertation, we focus on the contention-based MAC protocol design for underwater acoustic sensor networks. We have proposed three stochastic handshaking-free random-access underwater MAC protocols, with different MAC design considerations for high network throughput and low packet delay. We have conducted extensive simulations to evaluate the performance of the proposed protocols. In DAP-MAC, we leverage the long propagation delay by using group compatibility for concurrent transmission, which is shown to improve the network throughput effectively. In TARS, we explicitly resolve the spatial uncertainty issue by using a receiver-synchronized approach to relieve packet collisions. The protocol is designed to be adaptive to the traffic loads and is robust under dynamic underwater conditions. In SMARP, we make use of the opportunity of power capture in UWSNs to further improve network throughput. The protocol achieves better network performance than DAP-MAC and TARS in terms of throughput and packet end-to-end delay, with comparable power consumptions. Further, we conduct in-tank underwater experiments using the SM 975 acoustic modems. We evaluate the performance of the TARS protocol under a three-node single-hop network setting. Experimental results show that TARS achieves the highest packet delivery ratio with controlled low packet end-to-end delay, and is very suitable for the actual implementation in underwater acoustic sensor networks.

The work of DAP-MAC, TARS, and SMARP have been published in the conference proceedings of ICNC’15 [51], MASCOTS’15 [49], and SECON’16 [52] respectively. The work of testbed-based performance evaluations of the TARS protocol is to appear in the conference proceeding of OCEANS’16 [53]. The extended version of DAP-MAC has been published in the journal of Ad Hoc Networks [54]. The extended version of TARS has been submitted and the extended version of SMARP is in preparation.
CHAPTER 7. CONCLUSION

Our future work involves further extensive underwater experiments. First, we plan to do a set of experiments in an outdoor pool at the Northeastern University Marine Science Center (MSC) in Nahant, MA. We will evaluate the performance of the TARS protocol in networks with up to 7 nodes, in terms of packet delivery ratio, packet end-to-end delay, and power consumption. Second, we will deploy the modems in the ocean at the MSC and test the protocol performance in larger networks with more nodes and longer nodal distances.
Bibliography


BIBLIOGRAPHY


