DATA COLLECTION AND MEDIUM ACCESS CONTROL SOLUTIONS FOR UNDERWATER WIRELESS SENSOR NETWORKS

EDUARDO PINTO MENDES CÂMARA JÚNIOR

DATA COLLECTION AND MEDIUM ACCESS CONTROL SOLUTIONS FOR UNDERWATER WIRELESS SENSOR NETWORKS

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais - Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Luiz Filipe Menezes Vieira
Co-Advisor: Marcos Augusto Menezes Vieira

Belo Horizonte
April 2020

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6ª Região nº 1510

Câmara Júnior, Eduardo Pinto Mendes

C172d Data Collection and Medium Access Control Solutions for Underwater Wireless Sensor Networks / Eduardo Pinto Mendes Câmara Júnior. — Belo Horizonte, 2020
xvi, 76 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação
Orientador: Luiz Filipe Menezes Vieira
Coorientador: Marcos Augusto Menezes Vieira

1. Computação — Teses. 2. Redes de sensores aquáticas. 3. Redes de sensores aquáticas óptico-acústicas. 4. Coleta de dados. 5. Controle de acesso ao meio. I Orientador. II. Coorientador. III. Título.

CDU 519.6*22 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Data Collection and Medium Access Control Solutions for Underwater Wireless Sensor Networks

EDUARDO PINTO MENDES CÂMARA JÚNIOR

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Luiz Filipe Menezes Vieira - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Marcos Augusto Menezes Vieira - Coorientador
Departamento de Ciência da Computação - UFMG

Prof. Antonio Alfredo Ferreira Loureiro
Departamento de Ciência da Computação - UFMG

Prof. Ítalo Fernando Scotá Cunha
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 8 de Maio de 2020.

To my family and friends.

Acknowledgments

First of all, I would like to thank my family and friends for assisting me during all my life. Special thanks to my mother, Izaura, my grandparents, Irma, Adelti, Áurea, and Evandro, and my sister Larissa, for all their love and support. You were very important in this achievement, and I would be nothing without you.

I am very thankful to my advisor, Prof. Luiz F. M. Vieira, for introducing and guiding me in my academic life. Thank you for all the opportunities and friendship. I also would like to thank my co-advisor, Prof. Marcos A. M. Vieira, for all the help and opportunities provided.

I also thank my friends and colleagues from LECOM and other laboratories for all their help. Also, thanks for the fun moments and all the valuable discussions about work and life. I especially thank Alexander Sousa and Otávio Souza for having been by my side during times of joy and difficulties.

Lastly, I would like to thank every teacher who contributed to my education at some point in my life. Thank you for sharing your knowledge with me and for making me a better person!

"I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist must also be absolutely like a child. If he sees a thing, he must say that he sees it, whether it was what he thought he was going to see or not. See first, think later, then test. But always see first. Otherwise, you will only see what you were expecting. Most scientists forget that."

(Douglas Adams, The Ultimate Hitchhiker's Guide to the Galaxy)

Resumo

Redes de sensores sem fio aquáticas (RSSFAs) possuem o potencial de propiciar diversas aplicações em áreas científicas, comerciais e militares. No entanto, a tecnologia ainda não se encontra amadurecida o suficiente para tal. Nós das RSSFAs geralmente se comunicam através de transmissões acústicas ou ópticas, visto que comunicações por radiofrequência possuem baixo desempenho em ambientes aquáticos. A utilização dessas tecnologias de comunicação, juntamente com as características dos ambientes aquáticos, introduzem muitos desafios para o desenvolvimento das RSSFAs.

Essa dissertação provê soluções para dois problemas existentes em RSSFAs. Sua primeira parte têm foco no problema de coleta de dados em redes de sensores sem fio aquáticas óptico-acústicas (RSSFA-OAs), que são um tipo de RSSFA onde os nós podem se comunicar utilizando transmissões ópticas e acústicas. Nela, propõe-se CAPTAIN, uma solução que visa explorar o melhor de cada tipo de comunicação para melhorar a coleta de dados em RSSFA-OAs. CAPTAIN é um algoritmo que divide logicamente uma RSSFA-OA em agrupamentos, estabelece uma árvore de roteamento entre os nós e utiliza agregação de dados para entregar os dados coletados pela rede ao nó sorvedouro. Avalia-se o CAPTAIN utilizando simulações, onde ele é comparado ao algoritmo de caminho mínimo.

A segunda parte da dissertação aborda o problema de controle de acesso ao meio (MAC, do inglês *Medium Access Control*) em redes de sensores sem fio aquáticas acústicas (RSSFA-As), que são um tipo de RSSFA onde os nós se comunicam utilizando transmissões acústicas. Aqui, propõe-se um protocolo MAC chamado UW-SEEDEX. Esse protocolo utiliza escalonamentos aleatórios de *slots* de tempo, que são gerados a partir de sementes e alterados de acordo com geradores de números pseudo-aleatórios. Dessa forma, nós podem prever todo o escalonamento de seus vizinhos depois de trocarem suas sementes entre si. Eles então podem utilizar as previsões para planejar melhor suas transmissões e assim evitar colisões. Simulações são utilizadas para avaliar tanto como os valores dos parâmetros do UW-SEEDEX afetam seu desempenho, quanto para comparar o protocolo com outras soluções MAC encontradas na literatura.

Palavras-chave: Redes de Sensores Aquáticas, Redes de Sensores Aquáticas Óptico-Acústicas, Roteamento, Coleta de Dados, Agregação de Dados, Redes Aquáticas Acústicas, Controle de Acesso ao Meio, Escalonamentos Aleatórios.

Abstract

Underwater wireless sensor networks (UWSNs) can enable lots of scientific, commercial, and military applications in underwater environments. However, the technology is not mature yet. Nodes from UWSNs usually communicate through acoustic or optical transmissions due to the poor performance of radiofrequency (RF) communication in these environments. The use of these communication technologies, together with the characteristics of underwater environments, introduces many challenges for UWSNs.

This dissertation provides solutions for two existing problems in UWSNs. Its first part addresses the problem of collecting data from underwater optical-acoustic sensor networks (UOASNs), a subset of UWSNs whose nodes can communicate using both acoustic and optical transmissions. We propose CAPTAIN, a solution that explores the best of each communication technology to improve data collection in UOASNs. CAPTAIN is an algorithm that logically divides a UOASN into clusters, establishes a routing tree, and uses data aggregation to deliver all data collected to the sink node. Through simulations, we evaluate CAPTAIN by comparing it to the shortest path algorithm.

In the second part of this dissertation, we address the problem of medium access control (MAC) in underwater acoustic sensor networks (UWASNs), a subset of UWSNs whose nodes communicate using acoustic technology. We propose a MAC protocol for UWASNs called UW-SEEDEX. The protocol employs random time slot schedules that are driven by pseudorandom number generations and produced from seeds. This method allows nodes to predict each other's entire schedules just by exchanging their seeds, so they can better plan their transmissions to avoid collisions. We use simulations to evaluate how each of UW-SEEDEX's parameter affects its performance and how the protocol performs against MAC solutions from the literature.

Keywords: Underwater Wireless Sensor Networks, Underwater Optical-Acoustic Sensor Netwoks, Routing, Data collection, Data aggregation, Underwater Acoustic Networks, Medium Access Control, Random schedules.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Oceans cover more than 70 percent of the Earth's surface and, together with rivers and lakes, are essential for life on the planet. Millions of animals and plants live in them, and they also influence the world's weather and temperature. They are also vital for many human activities, such as transportation, commerce, and sustenance. Despite so much importance to us, more than 80 percent of the oceans remain unexplored [Corinaldesi, 2015; NOAA., 2018]. All these facts show us the importance of studying and monitoring the conditions of underwater environments.

Collecting data from these environments is an essential task for such activities, and it can be performed with the help of sensor networks. Underwater sensor networks consist of sets of sensor nodes that have sensing, processing, storage, and communication capabilities [Coutinho et al., 2018]. These sensor nodes are spread over a region to acquire data and perform other required tasks.

Underwater wireless sensor networks (UWSNs) have emerged as a technology that can enable lots of applications in underwater environments [Heidemann et al., 2012]. Water quality and pollution monitoring [Shakir et al., 2012], oil exploration [Jiejun Kong et al., 2005; Ribeiro et al., 2015], lake monitoring [Vieira et al., 2018], and early warning of natural disasters [Kumar et al., 2012] are just a few of them. Wireless communications allow the networks to be deployed through many distinct configurations, including nodes anchored to the ocean floor, autonomous underwater vehicles (AUVs), floating devices, or submarines, for example [Gussen et al., 2016].

Although wireless communication provides great flexibility for UWSNs, they also introduce many challenges. These challenges are even different from those commonly faced by terrestrial wireless networks.

First of all, energy is a very scarce resource for UWSNs. Their nodes tend to be battery-powered, and it may be very difficult or financially unfeasible to recharge them [Akyildiz et al., 2005]. Thus, energy consumption is a critical factor for UWSNs, which demands communications to be efficient.

Another challenge comes from the fact that features of underwater environments like salt concentration, pressure, and temperature may directly impact communications [Lanzagorta, 2012]. Also, since radio-frequency (RF) systems have some serious disadvantages in these environments, UWSNs are usually enabled by acoustic or even optical communication [Preisig, 2007; Zeng et al., 2017]. Water highly attenuates RF waves, which restraint them at low frequencies (30-300 Hz), and their propagation range to only a few meters [Pompili and Akyildiz, 2009]. Meanwhile, acoustic systems provide long-range communication, but they achieve low throughput and present high delays. Optical systems, on the other hand, provide high data rates over short range links and often require line-of-sight positioning. So, communication solutions must deal with the particularities of the communication technology employed in an UWSN.

Due to these before-mentioned challenges, many of the communication solutions developed for terrestrial wireless sensor networks are not efficient when applied to UWSNs. Therefore, they cannot be directly utilized in these networks. So, it is necessary to either adapt existing solutions to the context of underwater environments or design new and efficient solutions that consider their constraints.

1.1 Motivation

This dissertation addresses two common network problems in two different types of UWSNs. The first problem is data collection, which can be defined as the process of routing data from sensor nodes to a particular node (or set of nodes), denominated the sink node. In this work, we focus specifically on the problem of collecting data from underwater optical-acoustic sensor networks (UOASNs).

As the name suggests, UOASNs are UWSNs where nodes have hybrid communication systems and thus can transmit data through both acoustic and optical links. Routing solutions for UOASNs can, for example, take advantage of the low energy consumption and high data rate offered by optical transmissions, and the long-range offered by acoustic ones. The main challenge is how to explore the possibilities provided by having these two communication technologies available to develop efficient solutions that can deal with the limitations imposed by the underwater environment.

The second problem addressed here is medium access control (MAC) in underwater acoustic sensor networks (UWASNs). MAC solutions must define sets of rules for network nodes to efficiently access a shared medium, focusing mainly on energy efficiency or quality of service (QoS) assurances [Shah, 2009]. They are important because they affect issues such as node power consumption, latency, throughput, and bandwidth utilization [Demirkol et al., 2006]. Therefore, they are one of the most important parts of sensor networks, whether underwater or terrestrial [Jiang, 2018].

In the context of UWASNs, MAC solutions face challenges that are consequences of the characteristics of the acoustic channel. The channel's low capacity, coupled with its low quality and, in particular, the high propagation delay observed, are some of them. These challenges make the use of protocols designed for terrestrial radio networks not so efficient in UWASNs [Aval et al., 2016]. Constant control packet exchanges and carrier sensing, for example, are two common techniques used in terrestrial networks that are not so efficient in UWASNs due to the long propagation delays they experience [Kredo II et al., 2009]. So, it is necessary to create new MAC protocols or adapt existing ones to deal with the characteristics of underwater acoustic channels [Coutinho et al., 2016].

1.2 Objectives

This dissertation has the main objective of providing new solutions for existing problems in UWSNs, namely, the data collection and MAC problems. The solution for the data collection problem is an algorithm focused on UOASNs that explores the best of optical and acoustic communication and combines data collection and data aggregation with network clustering. By doing so, it can achieve high collection rates with low energy consumption.

The MAC solution takes inspiration from existing work for terrestrial wireless networks to provide a protocol where nodes can avoid collisions without frequent control message transmissions. The proposed protocol employs random time slot schedules that are created from seeds and driven by pseudorandom number generators, which allows nodes to predict each other's schedules just by exchanging their seeds.

1.3 Contributions

The main contributions of this work are:

- CAPTAIN, an algorithm for data collection in UOASNs; its techniques to perform cluster formation, routing establishment, and data collection; and the evaluation of the algorithm through simulations, whose results show the benefits of using CAPTAIN instead of the shortest path algorithm.

- UW-SEEDEX, a MAC protocol for UWASNs that avoids collisions with low overhead by employing random time slot schedules produced from seeds; the evaluation of the protocol parameters via simulations; and extensive simulations show that UW-SEEDEX can deliver more packets than other protocols found in the literature, using, on average, fewer transmissions than them and with low energy consumption.

1.4 Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces the basic concepts and challenges of underwater wireless sensor networks and also reviews the channel models used in this work. In Chapter 3, we discuss previous works related to the solutions proposed here. Chapter 4 proposes a cluster-based data collection algorithm, named CAPTAIN, for UOASNs. In Chapter 5, we propose UW-SEEDEX, a MAC protocol for UWASNs that employes random time slot schedules generated from seeds and driven by pseudorandom number generators. Finally, Chapter 6 summarizes the main contributions of this dissertation and lists possible future works.

1.5 List of Publications

Papers Related to the Dissertation

1. Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2020a). CAPTAIN: A data collection algorithm for underwater optical-acoustic sensor networks. *Computer Networks*, 171:107145. ISSN 1389-1286.
2. **(Under review)** Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2020b). UW-SEEDEX: a pseudorandom-based MAC protocol for underwater acoustic networks. *IEEE Transactions on Mobile Computing.*

Other Papers

1. da Silva Santos, E. R., Câmara Júnior, E. P. M., Vieira, M. A. M., and Vieira, L. F. M. (2019). Aplicações de monitoramento de tráfego utilizando redes programáveis ebpf. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 417--430. SBC. ***In Portuguese***.

2. Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2019b). Topology control in underwater sensor networks using cellular automata. *Journal of Cellular Automata*, 14.

3. Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2019a). 3DVS: Node scheduling in underwater sensor networks using 3D voronoi diagrams. *Computer Networks*, 159:73 – 83. ISSN 1389-1286.

4. Tavares, R. C., Carvalho, M., Câmara Júnior, E. P. M., de Britto e Silva, E., Vieira, M. A. M., Vieira, L. F. M., and Krishnamachari, B. (2019). FWB: Funneling Wider Bandwidth algorithm for high performance data collection in Wireless Sensor Networks. *Computer Communications*, 148:136 – 151. ISSN 0140-3664.

5. Vieira, M. A. M., Castanho, M. S., Pacífico, R. D. G., Santos, E. R. S., Júnior, E. P. M. C., and Vieira, L. F. M. (2020). Fast Packet Processing with EBPF and XDP: Concepts, Code, Challenges, and Applications. *ACM Comput. Surv.*, 53(1). ISSN 0360-0300.

Chapter 2

Preliminary Concepts

In this chapter, we present the basic concepts used during the development of this work. Section 2.1 presents the characteristics of underwater wireless sensor networks (UWSNs) and discusses the main challenges encountered when developing solutions for them. In Section 2.2, we review the models for both acoustic and optical underwater channels that have been widely considered in the literature and were used in this work.

2.1 Underwater Wireless Sensor Networks

Underwater wireless sensor networks are groups of sensor nodes dispersed in underwater environments whose objective is to gather information about them. These networks have great potential for benefitting applications in several areas, such as scientific, commercial, and military [Vieira et al., 2010].

The sensor nodes that compose UWSNs have storage, sensing, and processing capabilities to collect many distinct types of environmental conditions, such as water temperature, pH, and dissolved oxygen. To power all their activities, nodes usually have limited power sources such as batteries. Nodes also have communication capabilities to allow data exchanges to transport the data collected by them to one or multiple central points in the networks (sink nodes).

Unlike many terrestrial sensor networks, UWSNs usually do not employ radiofrequency (RF) communication. RF electromagnetic waves are highly attenuated in the water due to the conductivity of the medium [Lloret et al., 2012]. Besides, the attenuation is higher for higher RF wave frequency. As a consequence, RF waves only propagate over long distances at low frequencies (30 - 300 Hz), which requires high transmission power and large antennas [Pompili and Akyildiz, 2009; Melodia et al., 2013]. Therefore, RF waves are suitable only for very short-range communications (up to a few meters) in underwater environments.

As alternatives to RF systems, underwater nodes generally utilize acoustic or optical technology to communicate with others. Among these two technologies, acoustic is the most widely used in UWSNs [Zeng et al., 2017]. Networks whose nodes employ this kind of communication are called underwater acoustic sensor networks (UWASNs).

Compared to their RF counterpart, acoustic communications can reach much longer ranges. For example, EvoLogics S2CR commercial modems have up to 10 km operating range [EvoLogics, 2020]. The use of acoustic systems, however, also introduces many challenges for communication between the nodes of UWASNs. Acoustic modems have high power consumption, which negatively impacts the lifetime of nodes. This consumption is also uneven between transmissions and receptions, as nodes can spend up to 100 times more energy to transmit a packet than to receive one [Partan et al., 2007]. Another challenge is the low data rate achieved with this technology (in the order of kbps), mainly due to underwater channel characteristics. Some of them are:

- High propagation delays, which are five orders of magnitude greater than that observed in terrestrial electromagnetic channels (the speed of sound in the water is about 1500 m/s, while electromagnetic waves travel close to the speed of light in the air);
- High transmission error rates and temporary connectivity losses due to shadow zones (areas with poor propagation signal energy) [Manjula and Manvi, 2011];
- High path loss, noise, multipath propagation, and the Doppler effect that vary over time [Stojanovic and Preisig, 2009].

The other communication alternative for UWSNs uses optical signals to transmit data [Han et al., 2014]. Although optical signals face many difficulties in underwater environments, such as water absorption, scattering due to suspended particles, and disturbance produced by the Sun, experimental studies show their potential for some scenarios [Kaushal and Kaddoum, 2016]. Underwater wireless optical communication systems have some particularities that make them different from their acoustic counterpart.

Table 2.1[1] show a summary of the differences between the two communication systems. The first difference is how their signals propagate. While acoustic waves

[1] Adapted from [Farr et al., 2010].

are omnidirectional and propagate at speeds close to 1500 m/s, optical signals are directional and travel at about 2.55×10^8 m/s in the water. Another difference is that optical modems can send larger amounts of data per second (up to a few Gbps) than acoustic ones, but only for a few hundred meters [Lanzagorta, 2012]. Energy efficiency is yet another difference point. Optical modems are significantly more energy-efficient than the acoustic ones, as they can transmit about a thousand times more bits per joule [Farr et al., 2006].

Characteristics	**Acoustic**	**Optical**
Propagation	Omnidirectional	Directional
Range	Several km	Few hundred m
Data rate	Few kbps	Up to Gbps
Energy efficiency	Order of 10^2 bits/J	Order of 10^5 bits/J

Table 2.1: Trade-offs between acoustic and optical communications.

As we can see, acoustic and optical communications have somewhat complementary properties. An idea that comes from this observation is that both technologies can be used together in hybrid communication systems to get the best of each. UWSNs whose nodes use such a hybrid system are here called underwater optical-acoustic sensor networks (UOASNs). UOASNs can, for example, take advantage of the low energy consumption and high data rate offered by optical transmissions, and the long-range offered by acoustic transmissions. Vasilescu et al. [2005] notes that UOASNs can enable many applications, as they support low-speed broadcasts and high-speed directional data transfer.

Regardless of the type of communication used, efficient energy management is an important factor when developing solutions for UWSNs. Energy is a scarce resource in both underwater and terrestrial sensor networks. However, it is even more relevant for UWSNs, since their nodes have batteries, and it might be very difficult or financially unfeasible to recharge them [Akyildiz et al., 2005]. This fact means that the initial charge of the batteries restricts the lifetime of underwater nodes. Therefore, the development of energy-efficient solutions for these networks is crucial.

Solutions for UWSNs must also consider the three-dimensional (3D) nature of these networks [Vieira et al., 2011]. Most terrestrial sensor networks have their topologies represented using only two dimensions, i.e., in planes [Xiao, 2010]. The same is not true for UWSNs, as underwater nodes might be deployed along different depths.

Challenges may also appear depending on the architecture of UWSNs. Wireless communication opens up many possibilities for creating networks with different types of nodes and topologies. We can, for example, classify UWSNs according to the mobility

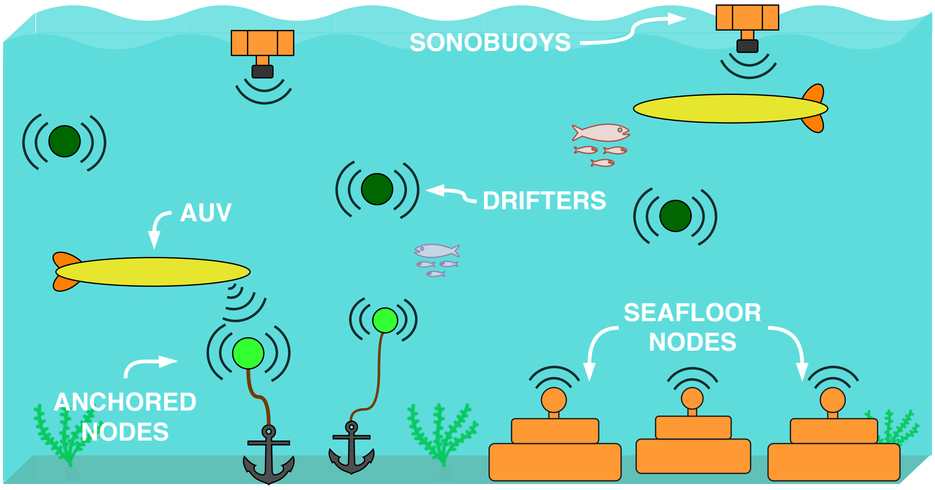


Figure 2.1: Example of UWSN architecture.

of their nodes. Static networks contain, for example, nodes attached to seafloor or docks, where their positions will not change. Alternatively, semi-static UWSNs have nodes whose positions do not vary much over time, such as anchored nodes. Mobile networks, on the other hand, posses nodes that can move around spaces, like nodes attached to drifters, sonobuoys, or autonomous underwater vehicles (AUVs). Figure 2.1 illustrates an UWSN where nodes have different mobility levels.

Heidemann et al. [2012] notes that static and semi-static UWSNs have the benefit of promoting connectivity through topology engineering. Meanwhile, mobile UWSNs propitiate the coverage of large areas with limited hardware and 4D (space and time) monitoring [Lee et al., 2010]. However, the use of mobile nodes also introduces challenges to maintain network connectivity and locate nodes. We note that locating nodes in underwater environments is not an easy task since the Global Positioning System (GPS) can not be used (due to its use of high-frequency RF waves). There are lots of localization alternatives for UWSNs in the literature [Erol et al., 2007, 2008; Luo et al., 2010], but they can be very costly for some applications [Caruso et al., 2008].

Additionally, node density is an impacting factor for UWSNs communication solutions. Even today, UWSNs are still restricted to experimentation given the high costs for their deployment, operation, and maintenance [Coutinho, 2017]. Partan et al. [2007] points out that, because of these points, these networks are expected to be sparse, and therefore contain mobile nodes to improve space coverage. However, Heidemann et al. [2006] believes that UWSNs should include some redundancy to prevent individual node losses from having significant impacts on network performance.

2.2 Underwater Channel Models

In this section, we describe channel models for both acoustic and optical communications in underwater environments. We employed the described models in the simulations we used to evaluate the solutions we propose here. Both models were introduced by previous works in the literature, and we choose them because many other works also consider them when simulating UWSNs.

2.2.1 Underwater Acoustic Channel Model

Here we consider the acoustic channel model used by many works in the literature [Coutinho et al., 2014; Câmara Júnior et al., 2019a; Lima et al., 2019]. Described by Stojanovic [2007], this model is mainly characterized by signal attenuation and noises in the environment. It depends on the distance d traveled by the signal and its frequency f.

The signal attenuation is given by Equation 2.1, where A_0 is a unit-normalizing constant, k is the spreading factor and $\alpha(f)$ is the absorption coefficient, in dB/km for f in kHz.

$$10 \log A(d, f)/A_0 = 10k \log d + 10d \log \alpha(f) \tag{2.1}$$

The first term of Equation 2.1 represents the spreading loss, while the second represents the absorption loss. The spreading factor k describes the geometry of propagation and its common values are 1 (cylindrical spreading), 1.5 (practical spreading), and 2 (spherical spreading). The absorption coefficient $\alpha(f)$ can be expressed using Thorp's formula [Brekhovskikh and Lysanov, 1982]:

$$10 \log \alpha(f) = 0.11 \frac{f^2}{1 + f^2} + 44 \frac{f^2}{4100 + f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003. \tag{2.2}$$

Ambient noise in oceans can be modeled as the sum of turbulence (N_t), shipping (N_s), waves (N_w) and thermal noise (N_{th}). Each one of these components can be expressed, in dB re μPa (which is the amplitude of the sound wave's loudness with a pressure of 1 micropascal) per Hz, as a function of frequency, in kHz, by the following empirical formulas [Coates, 1989]:

$$10 \log N_t(f) = 17 - 30 \log f, \tag{2.3}$$

$$10 \log N_s(f) = 40 + 20(s - 0.5) + 26 \log f - 60 \log(f + 0.03), \tag{2.4}$$

$$10 \log N_w(f) = 50 + 7.5 w^{\frac{1}{2}} + 20 \log f - 40 \log(f + 0.4), \tag{2.5}$$

$$10 \log N_{th}(f) = -15 + 20 \log f. \tag{2.6}$$

While N_t and N_{th} only depend on the signal frequency, N_s and N_w also depend on local environmental conditions. N_s, for instance, depends on distant shipping, which is modeled using the shipping activity factor s. The value of s ranges between 0 (low activity) and 1 (high activity). The water surface motion produces the N_w, and it is affected by the wind (w is the wind speed in m/s).

The SNR observed in an acoustic receiver can be evaluated using the signal attenuation $A(d, f)$ and the total ambient noise $N(f) = N_t\ (f) + N_s\ (f) + N_w\ (f) + N_{th}\ (f)$. When considering only the path loss gains and losses, the narrow-band SNR_A is given by

$$\mathrm{SNR_A} = \frac{P/A(d, f)}{N(f)\Delta f} \tag{2.7}$$

where P is the signal transmission power and Δf is the receiver noise bandwidth.

2.2.2 Underwater Optical Channel Model

We consider in this work the optical channel model described by Anguita et al. [2011]. This model, which was also employed by many previous works [Han et al., 2014; Campagnaro et al., 2015; Qiu et al., 2015], mainly considers water turbidity and the parameters of the transmitters. It can be divided between (i) the light propagation and distribution model and (ii) the communication channel characteristics.

Equation 2.8 shows the light propagation and distribution model in the water. P_r is the received light power (dBm), P_0 is the transmitter emitting power (dBm), A_r is the receiver area (m^2), and A_t is the transmitter area (m^2). β is the inclination angle (rad), θ is the transmitter light beam diverge angle, d is the distance between the receiver and the transmitter, and L is the perpendicular distance between the transmitter and receiver plane, as can be seen in Figure 2.2. c is the light attenuation coefficient, and Table 2.2 shows its common values for a wavelength $\lambda = 514$ nm (the one used in the simulations).

$$P_r = \frac{2P_0 A_r \cos\beta}{\pi L^2(1 - \cos\theta) + 2A_t} \mathrm{e}^{\dot{-}cd} \tag{2.8}$$

As noted by Anquita et al., Equation 2.8 models a single light source and it is the combination of the geometric light dispersion of the transmitter, the alignment difference between the transmitter and the receiver, and the light attenuation in the water.

Using P_r, it is possible to estimate the SNR in the optical communication channel. Equation 2.9 shows the SNR formula for optical transmissions, where q is the electric charge (1.6×10^{-19} C), K is the Boltzmann constant (1.38×10^{-23} J/K), T is the

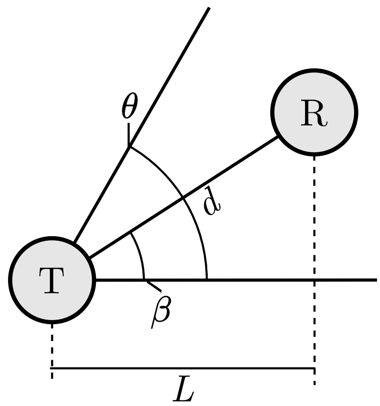


Figure 2.2: Parameters in the light distribution model.

Water Type	**Attenuation Coefficient ($\mathbf{m^{-1}}$)**
Pure sea water	0.043
Clean ocean	0.151
Coastal ocean	0.298
Turbid harbor	2.190

Table 2.2: Attenuation coefficient values of different types of water.

temperature in K, $\mathrm{B_w}$ is the system bandwidth (Hz), S is the receiver sensitivity (A/W), I_d is the dark photodiode current, I_l is the photocurrent generated by the incident light (A), and R is the photodiode shunt resistance (Ω).

$$\mathrm{SNR_O} = \frac{(SP_r)^2}{2q(I_d + I_l)\mathrm{B_w} + \frac{\mathrm{4KTB_w}}{\mathrm{R}}} \tag{2.9}$$

Chapter 3

Bibliographic Review

In this chapter, we present a review of previous works that relate to our proposed solutions. Section 3.1 discusses previous works related to CAPTAIN, covering studies in underwater optical-acoustic sensor networks (UOASNs) and data collection solutions. Section 3.2 reviews previously proposed solutions to medium access control (MAC) in underwater acoustic sensor networks (UWASNs) and solutions that use pseudorandom schedules.

3.1 Underwater Optical-Acoustic Sensor Networks and Data Collection

Many recent works have proposed the use of UOASNs for performing different tasks. Some of them are:

- Vasilescu et al. [2005] report the prototype of a UOASN with a mix of static and mobile nodes, with the former being used to collect data and the later as data mules. Nodes in this network use optical communication for fast and power-efficient data transfers, and acoustic communication for event signaling and localization.

- Vasilescu et al. [2007] designed and implemented a underwater wireless sensor network (UWSN) where nodes have both optical and acoustic communication to provide automated data collection for marine biology applications.

- Farr et al. [2010] developed an optical-acoustic communication system that offers high data rates and low latency within the range of the optical modem, and robust long-range transmissions out of it. The authors cite wireless remote operation of ROVs (Remotely Operated Vehicle) as a possible application for their system.

- Han et al. [2014] present a hybrid solution that uses acoustic communication to align optical transmitters and receivers to allow an increase in the amount of data transmitted.
- Johnson et al. [2014] propose using hybrid optical/acoustic links to enable duplex communication for applications with asymmetrical bandwidth needs, such as the communication between buoys or ships with Autonomous Underwater Vehicles (AUVs) and divers.
- Tennenbaum et al. [2014] evaluate the addition of low cost, short-range optical transmitter-receiver pairs to acoustic communication systems to improve the performance of applications such as time synchronization and TCP communications in UWSNs.
- Moriconi et al. [2015] study the utilization and the implementation of a hybrid communication system for dense underwater swarms of AUV.
- Wang et al. [2017] proposed an UOASN for real-time wireless transmissions of images and videos of marine exploration, where nodes use optical modems for fast data transmission, and acoustic modems to transmit control commands and node location. Han et al. [2019] propose a similar optical/acoustic hybrid solution to achieve real-time video streaming in UWSNs with mobile nodes.

Currently, there are many routing protocols that can be used to collect data from underwater acoustic networks [Pompili and Akyildiz, 2009; Lu et al., 2017; Khan et al., 2018]. Some examples are VBF [Xie et al., 2006], DBR [Yan et al., 2008], Pressure Routing [Lee et al., 2010], DCR [Coutinho et al., 2013], GEDAR [Coutinho et al., 2014], Hydrocast [Noh et al., 2016] and QERP [Faheem et al., 2018]. Although these protocols can be used in UOASNs, they were not designed to take advantage of the particularities of the two types of communication provided by these networks. Thus, they may suffer from inefficiency.

To the best of our knowledge, only a few routing algorithms were proposed for UOASNs. MURAO [Hu and Fei, 2012] is one of them. It is a cluster-based routing algorithm that performs data collection in UOASNs. Unlike CAPTAIN, our proposed solution, it considers that only cluster heads are equipped with both types of modems, while cluster members have only acoustic receivers and optical transceivers. MURAO also requires nodes to be spread so that the existence of gateway nodes (nodes in the intersection of two clusters) is guaranteed. This requirement does not exist for CAPTAIN since each node can identify whether it should be a cluster member, and transmit data using optical communication, or be a cluster head, and use acoustic communication to forward packets.

Wang et al. [2017] present another routing algorithm for UOASNs. It was proposed in the same paper where they propose a system for real-time transmission of images and videos. Different from CAPTAIN, this algorithm does not use data aggregation since it is focused on the delivery of images and videos.

Unlike the other routing algorithms for data collection cited here, CAPTAIN is the only one that combines data aggregation and data collection for UOASNs.

3.2 Underwater Medium Access Control

Jiang [2018] presents a broad study on MAC protocols for underwater acoustic networks. He claims that MAC solutions developed for RF-based wireless networks are not suitable for underwater networks due to the peculiarities of the acoustic channels in the underwater environment. The high propagation delay found on these channels negatively influences the efficient use of the medium, access fairness, and quality of service (QoS) assurance. Also, node power constraints demand the number of transmissions and receptions to be minimized, which affects protocols that depend on constant message exchanges. In the case of node mobility, it is still necessary to consider the impact of the Doppler effect on communications and in services such as synchronization and localization. The author also reviews many MAC protocols found in the literature, highlighting common structures among them.

Code Division Multiple Access (CDMA)-based MAC protocols allow collision-free simultaneous transmissions to occur, but need to address issues such as spreading of code assignments and power control of transmissions. PLAN [Tan and Seah, 2007] and DPC MAC [Wei et al., 2008] are two CDMA-based MAC protocols proposed for multi-hop underwater sensor networks. PLAN uses a node assignment system to prevent one node from using the same code as another within its two-hop neighborhood. Such a system uses an RTS/CTS scheme for code assignment, and this introduces a delay that can be very long for networks with large numbers of nodes. DPC MAC assigns a common code for control frame transmission and a unique code for data transmission. It also adopts a power control scheme that adjusts the power of each transmission to overcome the near-far problem (a condition where the receiver receives a strong signal that makes it unable to detect other weak signals). Such a scheme requires sending a control frame to each transmission, which is potentially detrimental to node power consumption and data uploads. Unlike these solutions, the one we propose here will avoid the high overhead of control messages by using pseudorandom schedules that can be predicted using only the seed of each node and the current time.

Carrier Sense Multiple Access (CSMA)-based MAC solutions can avoid collisions by sensing the channel, but they can cause the hidden and the exposed terminal problems. Besides, they tend to be unfair, as they favor nodes near the transmitting source by giving them early access opportunities Liao and Huang [2012]. ALOHA-CS [Guerra et al., 2009] is a CSMA protocol that transmits data only when the channel is identified as free and uses a backoff window between two and five times the maximum propagation delay when a transmission attempt fails. CSMA-ALOHA [Azad et al., 2011] is a protocol that senses the channel for a shorter time than the signal propagation time, aiming to improve the access opportunities to the medium at the cost of higher chances of collisions. The protocol proposed here avoid collisions using a Time Division Multiple Access (TDMA) scheme and should not suffer from the hidden and exposed terminal problems as it allows nodes to know the slot scheduling of their neighbors up to two hops apart.

Reserve-based protocols are effective at eliminating collisions at receivers, but they are affected by network topology as negotiations between the nodes are required for reservation [Jiang, 2018]. Tone Lohi (T-Lohi) [Syed et al., 2008] is a class of MAC protocols that use signals (tones) to try to reserve the channel and then sense it to verify the result of the attempt. SF-MAC [Liao and Huang, 2012] uses RTS/CTS message pairs to reserve the channel and try to ensure fair access to the medium. The MAC solution presented in this paper avoids the overhead required by reserving channels by letting nodes know when their neighbors are listening and when they can transmit, thus reducing the chance of collisions.

MAC solutions based on TDMA can minimize collisions in UWASNs, but they require the use of time synchronization (SYN) services. These services face challenges due to the high and varied propagation delays found in underwater environments [Pompili and Akyildiz, 2009], and many solutions already exist in the literature, such as TSHL [Syed and Heidemann, 2006] and Mobi-Sync [Liu et al., 2013].

LT-MAC Mao et al. [2015] and C-MAC Ma et al. [2009] are two TDMA-based MAC protocols that use different techniques to do time slot assignment. In LT-MAC, the slot assignment depends on the relative position between transmitters and receivers, and the duration of each slot is dynamically determined by the nodes based on the load on them. Using such dynamic tuning causes the protocol overhead to be high, which can be harmful to network performance. C-MAC divides the network into several cells and assigns a time slot to each cell to be shared among the nodes that belong to them. Jiang Jiang [2018] notes that, while C-MAC provides large coverage, the protocol is inefficient when the traffic distribution is unbalanced. UW-SEEDEX uses a TDMA scheme to avoid collisions, and, unlike both solutions, it employs a random slot assignment based on a pseudorandom number generator to avoid the need for constant control message exchanges.

During the literature review, we could not find any MAC protocol for UWSNs that uses random slot schedules. However, there are other protocols, such as ARNS [Kosowsky et al., 1988] and SEEDEX [Rozovsky and Kumar, 2001], that target other types of networks. ARNS is a MAC protocol designed for satellite networks, that uses pseudorandom sequences of Transmit, Receive, and Receive acquisition slots that are based on node IDs and time. This way, a node can know when to transmit to a neighbor by calculating its sequence, without requiring message exchanges to express the desire for transmissions. SEEDEX, which served as inspiration for our proposed solution, is a MAC protocol for terrestrial ad hoc networks that uses random schedules for the same purpose as ARNS. The schedules are sequences of time slots that define two states: one where nodes can transmit and another where they must only listen. The schedules are created from seeds, which nodes then disseminate to their neighbors so that they become aware of their schedules and thus avoid collisions. As both ARNS and SEEDEX are focused on networks that have different characteristics from the UWASNs, they do not directly deal with factors such as the high propagation delays faced by these networks. UW-SEEDEX, on the other hand, considers the delay propagation and, different from SEEDEX, adds acknowledgments to transmissions in time slots and employes an improved information dissemination scheme to deal with the characteristics of underwater acoustic channels.

Chapter 4

CAPTAIN: A Data Collection Algorithm for Underwater Optical-Acoustic Sensor Networks

In this chapter we propose CAPTAIN, a data **C**ollection **A**lgorithm for underwater o**PT**ical-**A**coust**I**c sensor **N**etworks. CAPTAIN was designed to be energy efficient since this is critical for UWSNs [Coutinho et al., 2016]. It first divides the network into clusters and then builds a routing tree to collect data. The data collected by members of clusters are aggregated by their respective cluster heads before been forwarded so that the overall message traffic on the network can be reduced. While intra-cluster communication uses optical communication, cluster heads use acoustic communication for inter-cluster communication.

The remainder of this chapter is organized as follows. In Section 4.1 we present the network model that we considered when designing CAPTAIN. The proposed algorithm is presented and illustrated in Section 4.2. Section 4.3 presents the evaluation methodology, describing the simulator developed for the tests, the settings of the simulations, and the test scenarios. Lastly, Section 4.4 discusses the simulation results.

4.1 Network Model

In this work, we consider an UOASN architecture composed of homogeneous sensor nodes, i.e., nodes that have the same capabilities in terms of sensing, communication, and energy. These sensor nodes are considered to be deployed in an underwater environment together with one or more sink nodes spread on the water surface, as shown in Figure 4.1. As nodes might be deployed along different depths, we consider them to create three-dimensional (3D) networks [Vieira et al., 2011]. Node positions are considered to change a little or nothing over time so that they only need to be estimated once.

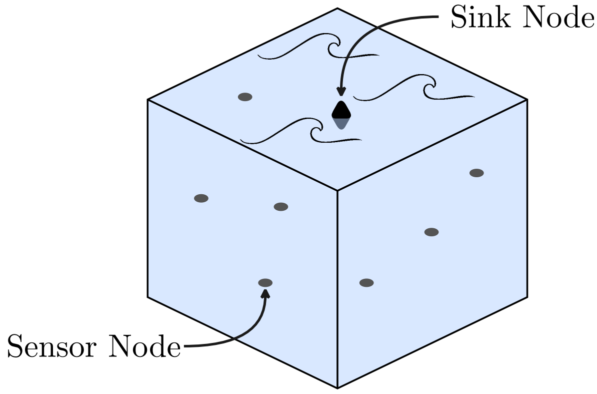


Figure 4.1: Example of UOASN following the considered model.

We assume that each node in the network has a unique identification (ID) and that every node is equipped with both optical and acoustic modems. Both modems are also assumed to be omnidirectional, being able to send and receive data from any direction. Although optical modems are usually directional, Farr et al. [2005] have shown that omnidirectional optical modems can be obtained using multiple LEDs.

For each type of modem, we define a neighborhood. The optical (acoustic) neighborhood of a node is made up of all nodes within the range of its optical (acoustic) modem. As the range of acoustic modems is usually greater than that of the optical modems, the acoustic neighborhood will contain the optical one. For example, in Figure 4.2, every node is an acoustic neighbor of node 4, and only node 2 is also an optical neighbor.

4.2 CAPTAIN Algorithm

CAPTAIN is an algorithm to perform data collection in underwater optical-acoustic sensor networks. It is designed for networks with multiple dense groups of nodes (clusters), where optical links can be used for data exchange within groups and acoustic links possibly connect various groups. The algorithm aims to explore the long range of acoustic transmissions and the high bandwidth of optical communication.

When designing the proposed algorithm, we considered a failure model where links could occasionally fail to deliver messages. Thus, nodes may need to retransmit messages, up to a limited number of times, when their destinations fail to receive them.

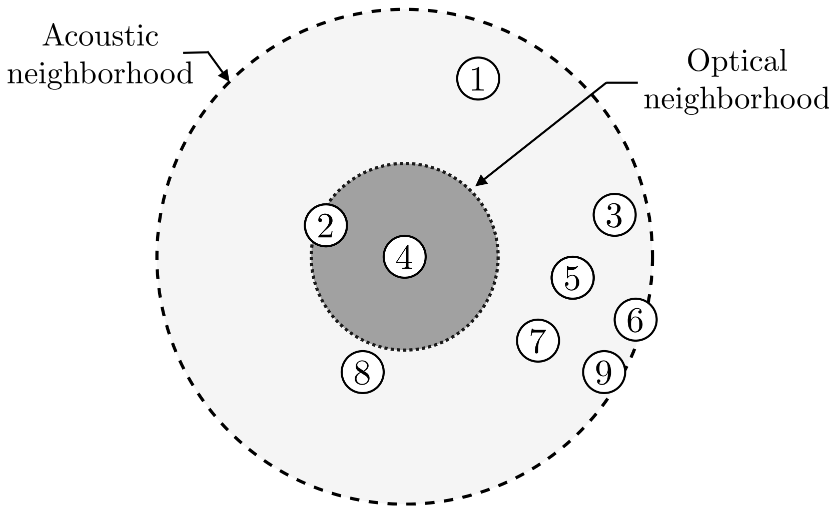


Figure 4.2: Example of optical and acoustic neighborhoods. Nodes within the inner circle (the gray area) are optical neighbors of node 4, while the nodes within the outer one are its acoustic neighbors.

We also assumed that nodes could permanently fail, and so become unavailable to the rest of the network.

CAPTAIN is based on clustering and uses a data aggregation scheme to reduce the overall message traffic and save energy. The algorithm is composed of a configuration period and an operation period. Nodes organize themselves to create routes in the former so they can collect data in the later.

We can consider that CAPTAIN has three phases: 1) cluster formation, 2) route establishment, and 3) data exchanging and route maintenance. While the first two phases represent the configuration period, the last represents the operation period. Next, we describe each one of these phases and also use an example to guide the description.

4.2.1 Phase 1: Cluster Formation

CAPTAIN is based on clustering since this technique allows some advantages for sensor networks such as MAC and routing scalability [Mhatre and Rosenberg, 2004]. In its first phase, it starts the configuration period by dividing the network into clusters, classifying nodes as cluster heads or members. Throughout this phase, as well as the subsequent one, CAPTAIN considers that the network will not change, i.e., no node will leave or enter.

The first phase should begin some time after the nodes have been deployed, and all of them are operational. Nodes must use acoustic transmissions throughout this phase so their messages can reach as many other nodes as possible. Algorithm 1 describes, in a simplified way, the process of cluster formation that must be performed by all nodes of the network.

Algorithm 1 Cluster Formation

broadcast a discovery message
receive the discovery messages from its neighbors
create lists of optical and acoustic neighbors
calculate its score
broadcast its score to its neighbors
receive scores from neighbors
if own score is the highest among its optical neighbors
    become a cluster head
else
    become a cluster member
    define the neighbor with the highest score as its next hop
broadcast whether it is a cluster head or not

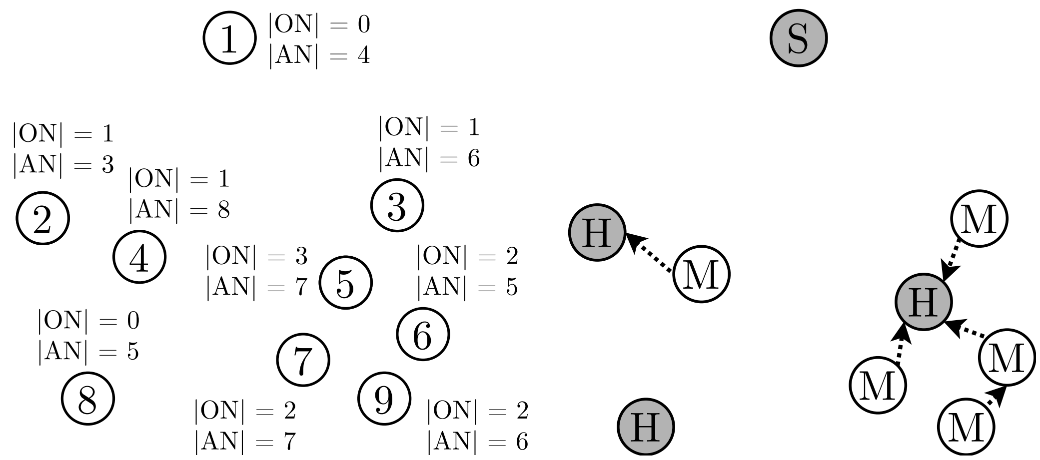
First, some node must begin the neighborhood discovery process by broadcasting a discovery message to the nodes around it. This node can be the sink node, for example. A discovery message should mainly contain the source node's ID and its location. Obtaining the position of an underwater node may be more difficult than getting that of a terrestrial one since Global Positioning Systems (GPS) does not work properly in underwater environments [Akyildiz et al., 2005]. However, localization services such as DNR [Erol et al., 2007], LPS [Vieira et al., 2009], and others [Othman et al., 2006; Zhou et al., 2007; Erol et al., 2008] can be used to estimate their positions. As we assume that nodes are almost static, these positions need only be estimated once.

When a node A receives a discovery message sent by a node B, it registers B as an acoustic neighbor and uses information about the location of B to check if it is also an optical neighbor. It does so by calculating the distance between them and checking if this distance is less than the range of its optical modem. We note that as the range of optical modems depends on conditions such as water turbidity, nodes may need to estimate the range of their optical systems in advance to properly determine their optical neighbors.

After receiving the discovery messages from its neighbors, a node can use the information gathered, together with that on how much energy it has, to calculate its score (see Section 4.2.1.1 for details about score calculation). This score is first used to determine which nodes are becoming cluster heads and which ones are becoming cluster members. A node will be a cluster head only if none of its optical neighbors has a score higher than its own. Otherwise, it will be a cluster member. In the case of a score tie, the node ID is used to break the tie.

After calculating its score, a node broadcasts it to its neighbors and waits to receive the scores of its optical neighbors. The node decides whether it becomes a cluster head or not after receiving the scores of its neighbors and then it announces its state to the other nodes. Nodes that receive announcement messages from cluster heads must save the information to be used later in the route establishment phase.

Figure 4.3 illustrates two moments of the cluster formation process. In the first one (Figure 4.3a), the nodes have already sent and received the discovery messages and, therefore, they now know how much optical and acoustic neighbors they have. They then calculate their scores and share them among themselves to build the clusters. Figure 4.3b shows which nodes have become cluster heads and which have become cluster members.



(a) Nodes gathered information about their neighbors.

(b) Cluster heads (gray nodes with H or S) and members (white nodes with M).

Figure 4.3: Exemplified cluster formation process.

4.2.1.1 Score calculation

CAPTAIN defines the score calculation as follows. The score S of a node is a real number in the interval [0,2], and it is given by

$$S = \begin{cases} 2, & \text{if the node is the sink} \\ 0, & \text{if ON} = \text{AN} \\ \frac{|\text{ON}|}{|\text{AN}|} + \text{P\_ENERGY}, & \text{otherwise} \end{cases}$$

where ON and AN are, respectively, the set of optical and acoustic neighbors, and P_ENERGY is the percentage of remaining energy of the node (value between 0 and 1). A sink node receives a score equals to 2 to assure that it will be a cluster head and that it will directly receive the data from the nodes next to it. A node that only knows its optical neighbors (|ON| = |AN|) gets the lowest score possible (S = 0) because it would not be able to communicate with any other cluster if it becomes a cluster head. The scores of the other nodes are calculated based on the rate of neighbors that are within the range of their optical modems (|ON|/|AN|) and on the percentage of their remaining energy.

To exemplify the score calculation, lets we consider the nodes at the moment of Figure 4.3a and suppose that all of them has 95% of their initial energy. Table 4.1 shows the scores that the nodes would have in this situation. We can see that nodes 2 and 5 have higher scores than its optical neighbors and so they would become cluster heads.

Node	1	2	3	4	5	6	7	8	9
Score	2.000	1.283	1.117	1.075	1.379	1.350	1.236	0.950	1.283

Table 4.1: Possible scores of the nodes of Figure 4.3a.

We designed this way of calculating the score to favor the nodes with the largest numbers of optical neighbors at the beginning of the network operation to benefit the later use of the data aggregation scheme. We note here that we assume that all nodes start with almost the same level of energy available so that the first term is more relevant to the distinction of the scores when the network starts operating. Later in the network execution, it may be necessary to change cluster heads, and when this happens, the node scores will be used again. This time, we expect the energy factor to be more relevant, and so the nodes with the highest amount of energy will tend to become the new cluster heads.

4.2.1.2 Cost for building clusters

Here we analyze the cost of the first phase of CAPTAIN by examining the number of (acoustic) transmissions required for building clusters in a network with n nodes. This analysis does not take into account the cost of using a localization service since many can be used. Also, we suppose that the same message is not transmitted more than d times.

Nodes start the phase by broadcasting discovery messages. If the transmission channels were error- and collision-free, then each node would need to transmit its discovery message only once. As this is not usually the case, some nodes may need to retransmit their messages a couple of times. So, the maximum number of discovery message transmissions is dn.

The next two moments where nodes transmit messages are when they share their score and then their state. Both sharing moments require, again, the broadcast of at least one message per node. Thus, the total cost of the first phase is at least $3n$ and at most $3dn$, plus the cost for the localization service. If d is a constant value, then we can consider that the first phase of CAPTAIN is $O(n)$. Otherwise, if d is not constant and, for example, it depends on n (more nodes in a network may lead to more interference, which consequently requires more retransmissions), than the cost is $O(dn)$.

4.2.2 Phase 2: Route Establishment

After forming the clusters, nodes go to the next phase, where they establish routes to deliver data to the sink node(s). They first define the routes within the clusters and then create routes connecting the cluster heads.

To create the routes within the clusters, their members use only the information already available to them, thus not requiring a new message exchange. Each member defines its neighbor with the highest score as being the next hop for its messages. This definition creates routes to take data from cluster members to cluster heads. Figure 4.3b shows the routes created in the example using arrows to indicate them.

To connect the cluster heads, CAPTAIN defines that the nodes must build a routing tree with the sink node as the root. To do this, they shall use acoustic communication to broadcast tree construction messages informing their neighbors about their distance, in hops, to the sink node. The building starts with the sink node broadcasting a tree construction message informing its neighbors that it is zero hops away from the root. Then, cluster heads that receive this message join the routing tree and acquire the knowledge that they are only one hop away from the sink node. These cluster heads should then send new tree construction messages to others so they may join the routing tree as well. Thus, any cluster head that receives a tree construction message and is not already in the tree should join it and send a new tree construction message.

When a node that is already in the routing tree receives a tree construction message, it must check if the path to the sink node through the message source node has a smaller number of hops (is shorter) than its current one. If not, then nothing should be done. Otherwise, the node must change its next hop and update its neighbors about the change by sending a new tree construction message to them. If both paths have the same number of hops, then the node probabilistically decides whether to change its next hop or not. The closer the message source node is to it, the more likely it is to change its next hop. This probabilistic decision aims to prevent only one node from being the next hop of many others when multiple options, with the same distance to the sink node, are available but are not used because of the arrival time of messages or errors in transmissions/receptions. We note that in this last case, the node does not need to send a new message if it changes its next hop since the distance to the sink node will continue the same.

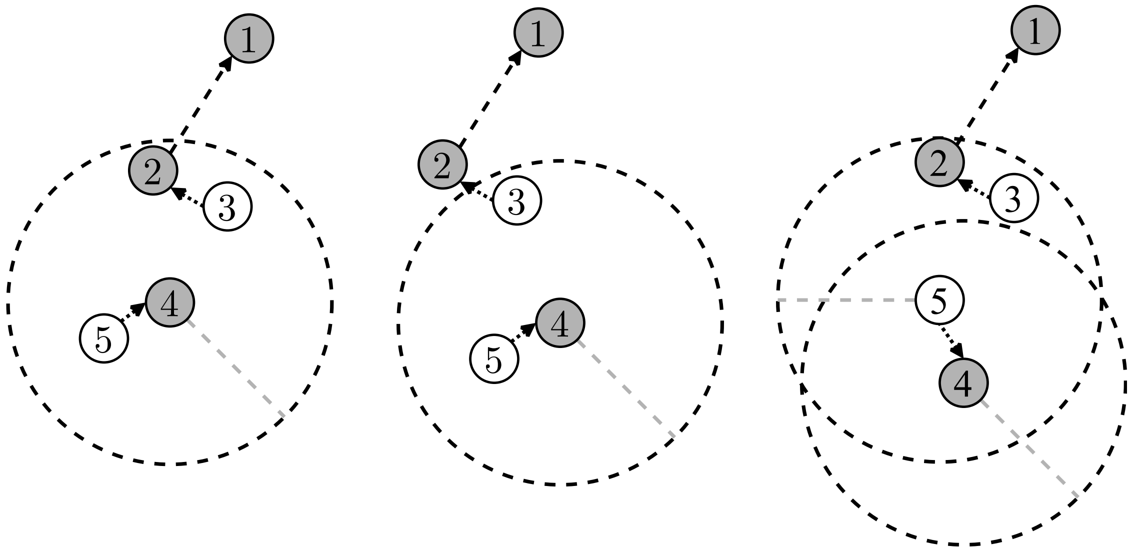
It is worth noting that transmission/reception errors could also cause some nodes to stay out of the routing tree. Therefore a cluster head needs to know if all its neighbors that are also cluster heads have received its tree construction message. If any of them has missed it, then the node must retransmit it.

It may also happen that some cluster head does not have contact with any other head, and then it would need another way to join the routing tree. As a cluster head usually know some node from another cluster (nodes that only know their optical neighbors are penalized with null score), a solution to this problem would be to use this node as its next hop even if it is a member. For this to be possible, cluster members must also transmit tree construction messages and nodes must keep a record of alternative routes using a cluster member as next hop. Thus, when this cluster head receives a message informing of a possible route using a cluster member, it sets that node as its next hop and continues the tree construction message propagation.

Lastly, another situation is one where a cluster head can not contact any other node on the routing tree. This circumstance can potentially create disconnected networks, as some clusters would not be able to send their data to the sink node. One way out of this situation exists if another node in the cluster can contact some other that is in the routing tree. This node, which would be a member, could become the head of its cluster and enter the routing tree, while the old head could become a member. So, this solution would change the head of a cluster.

In CAPTAIN, cluster members are responsible for detecting this situation by also listening for tree construction messages. After receiving tree construction messages from other clusters but none from their own, members can assume that their head could not receive any of these messages. Then, those that know a way for entering the routing tree may request their head for an exchange of roles. After receiving exchanging requests from nodes of its cluster, the head that can not enter the routing tree by its own decides which member should be the new head and then informs its decision to the others. When the chosen member listens to the head decision, it changes its state to cluster head and announces it to the remaining of the cluster. Also, the new head uses the information from the previously received tree construction messages to define its next hop and to continue the propagation of messages.

Figure 4.4 shows examples to illustrate the three situations we just described. In the first situation (Figure 4.4a), node 4 must enter the routing tree by defining node 2 (another cluster head) as its next hop. In Figure 4.4b, node 4 cannot communicate with any other cluster head, so it must define node 3 (a member that is already in the routing tree) as its next hop. Figure 4.4c shows the situation where node 4 cannot communicate with other nodes in the routing tree, but node 5, a member of its cluster, can. So, nodes 4 and 5 should exchange their roles, and then the later must define node 2 as its next hop. A fourth situation exists, where a cluster head is isolated from the network, but we do not consider this one since there would be nothing that could be done.



(a) Cluster head can communicate with another head.

(b) Cluster head can only communicate with a member in the routing tree.

(c) Cluster head cannot communicate with another node in the routing tree but a member in its cluster can.

Figure 4.4: Partial views of networks to illustrate the three situations in which a cluster head may be in the second phase. Again, cluster heads are represented as gray nodes.

Figure 4.5 shows a flow diagram for cluster heads to enter the routing tree. Following the example shown in Figure 4.3, Figure 4.6 shows the routing tree that would be constructed by the nodes after they finish the second phase.

4.2.2.1 Cost for building routes

Here we do a simple analysis of the number of acoustic transmissions required for nodes to build a routing tree in the second phase of CAPTAIN. Again, we suppose that the

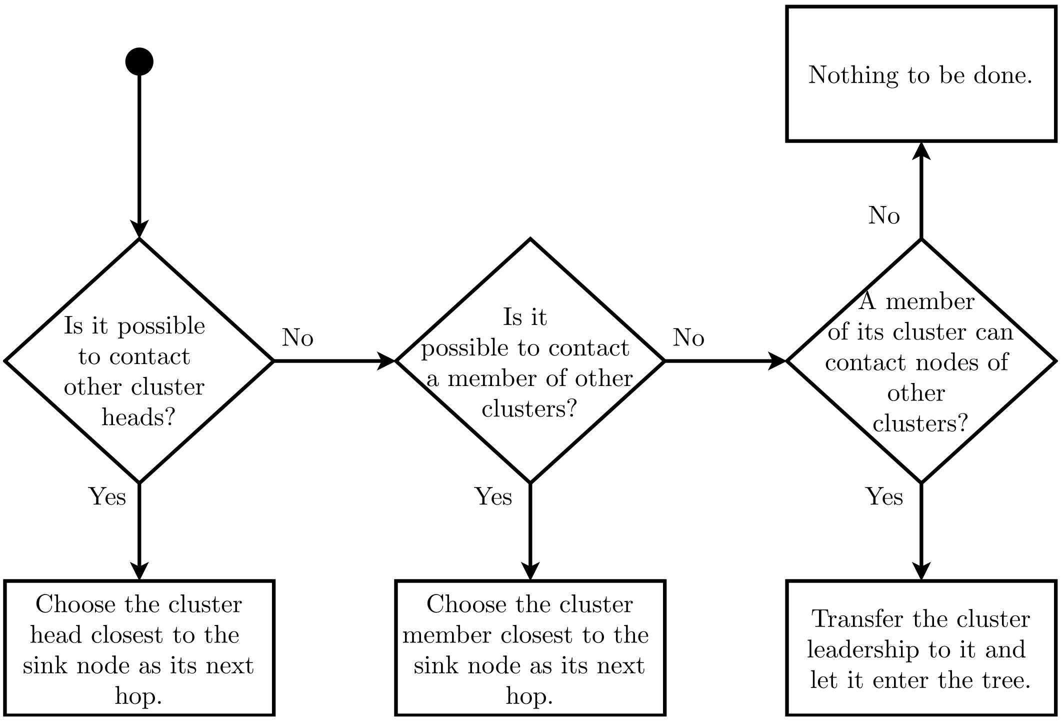


Figure 4.5: Flow diagram for a cluster head to enter the routing tree.

same message is not transmitted more than d times.

In the best case, nodes need to broadcast only one tree construction message to their neighbors, and no role exchange is necessary. Therefore, the second phase has a cost of at least n transmissions.

To analyze the worst-case scenario, we have to consider that nodes only transmit new tree construction messages when they find shorter routes to the sink node. This fact implies that each message comes with a reduction in the path length. As a node can discover at most $n-1$ routes that are shorter than its current one (although the total number of paths is exponential), it will also transmit at most $n-1$ messages. Thus, the second phase of CAPTAIN is $O(n^2)$.

4.2.3 Phase 3: Data Exchanging and Route Maintenance

After joining the routing tree, a node can move to the third phase. This phase marks the beginning of the operation period, where nodes will start sending data to the sink node. In this phase, new nodes will also be able to enter the network.

When a new node enters the network, it will broadcast a discovery message just like the other nodes did in the first phase. This time, however, the other nodes will reply to the message, providing information about their state (cluster member or head) and their place in the routing tree. This way, the new node gets to know its neighbors and is also able to decide its state and join the routing tree based on the responses.

Moving to the data collection activity, we can note that, besides sending the own collected data, some nodes must also forward data collected by other nodes. In the tree routing, each node must forward all the data from its descendants (the nodes belonging to the subtree where it is the root). This requirement creates a potential problem because it can cause the nodes of the highest levels of the tree to have a large volume of data to send and, consequently, much higher energy consumption than the others. To avoid a large number of messages traveling on the network, CAPTAIN uses data aggregation. Data aggregation methods are used along with other USN protocols to achieve better performance results [Goyal et al., 2017]. So, cluster heads must aggregate the data collected by their clusters before forwarding them to their next hops. Assuming that all nodes perform periodic data collections, the data aggregation helps to reduce the number of transmissions that a cluster head needs to do every round from n_c to m, where n_c is the size of its cluster and $m < n_c$.

Following the previous example, Figure 4.7 shows that node 5 is responsible for receiving the data collected by the nodes 3, 6, 7 and 9, aggregate it and then send it to the sink node (node 1). Node 2 should receive, aggregate and forward data from nodes 4 and 8.

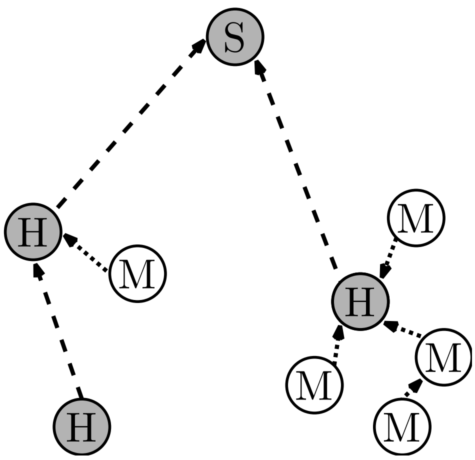


Figure 4.6: Example of a routing tree created in phase 2. Dotted and dashed lines represent optical and acoustic links, respectively.

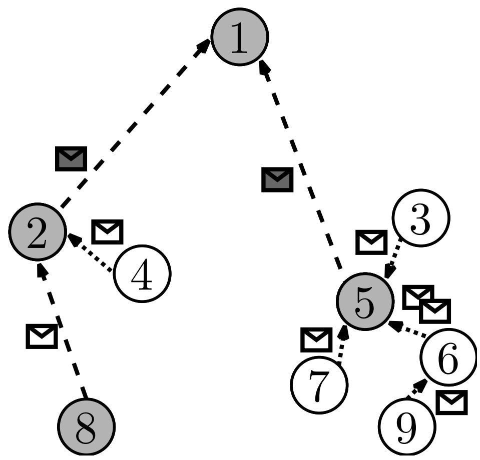


Figure 4.7: Messages transmitted during a data collection cycle. Dark messages represent aggregated messages.

The type of communication employed in data transmissions will depend on the node that holds the data. If the node is a cluster head, then it should use its acoustic modem to transmit the messages to the next hop. Otherwise, if the node is a cluster member, then it should use its optical modem.

As cluster heads communicate using acoustic transmissions, their energy consumption tends to be a lot higher than that of cluster members. Thus it is necessary to use a head rotation scheme to prolong the network lifetime. The scheme used by CAPTAIN works as follows: when the energy of a cluster head is below some threshold value, it performs a local search with its neighbors about their scores and their alternative routes. After receiving their answers, the head selects those who have higher scores than its current one. Then, it looks among the selected ones for those who know alternative paths to the sink node that are no longer than its. If it finds nodes that match the two criteria, then it transfers the cluster leadership to the one with the highest score. Otherwise, it remains the head of the cluster. When the leadership transfer occurs, the old cluster head informs the other nodes about the change and then become a cluster member.

Multiple threshold values can be defined to create a better relay between the cluster nodes. However, excessive amounts of thresholds can harm network performance since each rotation attempt requires nodes to exchange messages. It is worth noting that cluster heads without neighbors do not execute this head rotation scheme for obvious reasons.

Nodes may also run out of energy and die while the network is still operating. The death of one node causes the nodes on its subtree to lose contact with the sink node, so it is necessary to replace the dead node using alternative routes or creating new ones. In CAPTAIN, the nodes that lost their next-hop do this task. A node identifies the death of its next-hop when the later fails to acknowledge n_l messages in a row.

After identifying the death situation, the "orphan" node enters a recovering state, where it will look for ways to rejoin the routing tree. It does so by asking its neighbors information about routes in the routing tree that do not use the dead node. If the "orphan" node is a head, then it will broadcast a message to its neighbors and use the best route replied by them to rejoin the routing tree (if some exist, of course). Otherwise, if the node is a member, it will broadcast the request to its optical neighbors and, after receiving them, it will check for the best alternative path and use it to rejoin the routing tree. If no messages are received, then the node changes its state to head and continue to search for alternative paths using its acoustic neighbors.

We note that the way to identify node deaths may sometimes fail, as nodes can fail to transmit/receive more than n_l messages in a row without necessarily meaning that they died. However, picking appropriate values for n_l should diminish the odds of failures. Also, "orphan" nodes may receive messages from the presumably dead nodes when recovering and so abort the process when this happens.

4.3 Evaluation Methodology

We evaluated CAPTAIN using simulations and compared its performance to that of the shortest path algorithm, since it is a mature algorithm for sensor networks. The implemented shortest path algorithm (from now on called SPA) considers a graph where vertices represent nodes, and edges indicate the possibility of communication, either acoustic or optical, between pairs of nodes. Besides that, edge weights are given by the distance between the nodes. Following these considerations, SPA defines that each node must choose that neighbor that belongs to the shortest path to the sink node as the next hop of its messages. Also, nodes should communicate optically if the next hops of their messages are within the range of their optical modem and acoustic communication otherwise.

Next, we give a brief description of the simulator used in the tests and the channel models used by it. Then, we describe the simulation settings and test scenarios.

4.3.1 Simulator

We implemented both CAPTAIN and SPA using an underwater optical-acoustic network simulator, written in Python, developed for the tests[1]. Nodes are simulated as if they were running an application stack with four layers. The highest layer of the stack contains an application that periodically generates new data messages. The layer just below is the routing one, where CAPTAIN or SPA are used to deliver the data messages to the sink node. In the next layer, a TDMA system where only one node transmits at each time slot is used to prevent message collisions from happening. We note that the use of this system is abstracted so that no message exchanges are simulated for the synchronization service or for distributing time slots between nodes. In the lowest layer, the nodes access the acoustic and optical channels to transmit their data.

Nodes start simulations with a predefined amount of energy and die when it reaches zero. The energy consumption of the nodes is simulated considering only packet transmissions and receptions. Thus, a node consumes a part of its energy whenever it transmits a packet, just as the receiving node also spends energy to receive it.

The successes of transmissions are determined depending on the packet error rate (PER) of the channel used by them. When simulating a transmission, the simulator

[1] Available at `https://github.com/epmcj/captain-sim`.

first estimates the signal-to-noise ratio (SNR) based on the channel model of the communication type used by the node. The channel models used to simulate acoustic and optical transmissions are those described in Sections 2.2.1 and 2.2.2, respectively. To employ this optical model, we consider nodes omnidirectional modems to be composed of multiple light sources so that transmissions could be considered to happen between two single light sources.

Then, the simulator uses the SNR value to calculate the PER for the transmission and decide whether it was successful or not. A packet is successfully received only if all of its bits were correctly received. In other words, a packet error happens if any of its bits were flipped. So, to estimate the PER expected in transmissions between pairs of nodes, it is first necessary to estimate the bit error rate (BER). Considering the use of the BSPK modulation, where each symbol carries a bit, in an AWGN (Additive White Gaussian Noise) channel, the simulator calculates the BER as [Rappaport et al., 1996]:

$$\mathrm{BER} = \frac{1}{2}\left(1 - \sqrt{\frac{\mathrm{SNR}}{1+\mathrm{SNR}}}\right). \tag{4.1}$$

Using the calculated BER, it estimates the PER of an m-bits long packet using

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^m. \tag{4.2}$$

We note that, as there are no concurrent transmissions, the simulator does not address the possibility of interference between optical and acoustic transmissions.

4.3.2 Simulation Settings and Scenarios

Acoustic transmissions are simulated using the following conditions. We considered a spherical spreading of the signal ($k = 2$), no shipping activities ($s = 0$), and that the wind speed was $w = 0$ m/s. The simulations use the Evologics S2CR 18/34 acoustic modem [EvoLogics, 2014] as a base for acoustic communications. The modem frequency was 26 kHz, and we consider a transmission rate of 10 kbps (the modem's maximum data rate is 13.9 kbps). Considering a transmission range of 1000 m, it consumes 2.8 W in transmission mode and 1.3 W in receive mode.

For the optical transmissions, we used the following parameter values. We considered the utilization of an optical wave with wavelength $\lambda = 514$ nm in an ocean with pure water (beam light attenuation coefficient $c = 0.043\ \mathrm{m}^{-1}$). We assume the use of Si PIN photodiode Hamamatsu S5971 high-speed photodiode as optical receptors, as in [Campagnaro et al., 2015]. Its parameters values are transmission area $A_r = 1.1\ \mathrm{mm}^2$, sensitivity $S = 0.26$ A/W, maximum dark current $I_d = 1$ nA and *shunt* resistance $R = 1.43 \times 10^9 \Omega$, and the bandwidth is set to $B_w = 100$ kHz. Optical transmitters have beam diverge angle equals to 0.5 rad, transmitter size $A_t = 10$ mm^2 and a transmission rate of 1 Mbps. Based on the BlueComm 200 optical modem [Sonardyne, 2016] (an omnidirectional optical modem), we consider the transmission range equal to 50 m (BlueComm 200 maximum transmission range is 150 m) and the consumption in receive and transmission mode equal to 10 and 15 W, respectively. We also consider that nodes can receive data transmitted from any direction and send their messages with perfect alignment between the transmitter and the receiver.

The aggregation scheme used is the one proposed by Manjula and Manvi [2012]. In this scheme, one message is generated using the average values of all received data. So a cluster head can reduce the number of transmissions per round from n_c (the size of its cluster) to only one. As the nodes of the same cluster are expected to be close to each other (optical communication has a short-range), the data collected about the environment (e.g., water temperature, electric conductivity, or pH) by them should be very similar so that their average might be a good representation of them.

The simulations did not count the energy cost of the location service since both CAPTAIN and SPA requires it. All nodes but the sink started the simulations with an initial energy of 1000 J. We considered the sink nodes to have unlimited power supplies since, by staying on the surface, they could, for example, use solar energy to recharge their battery whenever necessary.

The number of nodes in the simulated networks varied from 50, 100, 150 or 200 plus one sink node. Nodes were distributed in 3D regions of size 2000 m $\times$ 2000 m $\times$ 2000 m, and we used two different strategies to distribute them. The first is a simple random deployment, where the positions of the nodes were chosen randomly within the 3D region. The second strategy is also random, but it forces the existence of clusters. This strategy first randomly defines central points for each cluster and then distributes equal amounts of nodes within spaces delimited by spheres of radius equal to twice the range of the nodes optical transmitters and centered in these points. We used this second strategy to generate networks with clusters of five or ten nodes. Both random deployments were used to avoid restricting the simulation results to predefined network topologies.

For each test case, we simulated 24 hours of network operation. The first hour of the simulations was used for the nodes to set the network, establishing the routes required to deliver data to the sink node. After that, the nodes started collecting data from the environment. New data collections were made every hour and generated 200 bytes long messages for each node. Transmissions of a data messages required confirmation, which was done by the receiver replying with a 10 bytes long acknowledgment (ACK).

The simulator distributed the time slots of the TDMA system according to node ID: node 1 got the first slot, node 2 got the second slot, and so on. The size of each time slot was defined as 1.6 seconds to ensure the transmissions of at least one data message and its ACK in each time slot. As acoustic transmissions are slower than the optical ones, the maximum amount of time required to send a data message is reached when the receiver is 1000 m (the maximum range of the acoustic transmission) away from the sender. The simulated acoustic modem takes 0.160 s (200 bytes $\times$ 8 bits/byte / 10 kbps) to transmit a data message and 0.008 s to transmit an ACK. In this worst case situation, both messages take around 0.667 s (1000 m / 1500 m/s) to propagate from the sender to the receiver and so the transmissions take $0.160 + 0.008 + 2 \times 0.667 = 1.502$ s. So a time slot of 1.6 s is enough for at least one acoustic data transmission to occur (if no error happens). We note that, since optical transmissions are faster than the acoustic ones, many of them can occur in only one slot.

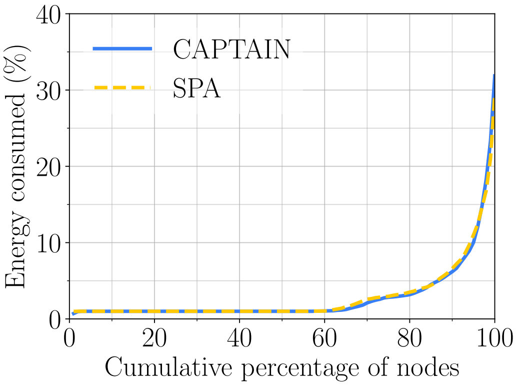
4.4 Results

In this section, we report the results obtained in our tests and analyze them. All the results presented in the figures are the average values of 100 executions, and Figures 4.9 to 4.11 present a 95% confidence interval.

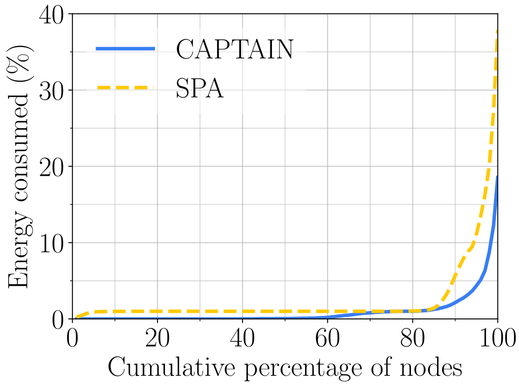
We first checked the energy consumption of the nodes when using CAPTAIN or SPA. Graphs of Figure 4.8 show how much energy was consumed by the nodes in networks with 100 nodes and different cluster sizes. While Figure 4.8a shows similar energy consumption for both algorithms, Figures 4.8b and 4.8c indicate that CAPTAIN spent less energy from some nodes than SPA in networks with clusters. This way of deployment favors the existence of cluster members in the network, which allows greater use of optical communication and a reduction in the number of messages sent by heads due to the data aggregation scheme.

Table 4.2 shows the average network energy consumption on the same three scenarios. The values are consistent with the curves of Figure 4.8, showing a similar energy consumption for CAPTAIN and SPA in networks without clusters, and much lower consumption for CAPTAIN in the others. The best case of network energy consumption was recorded when using CAPTAIN in networks with clusters of 10 nodes. In this scenario, it consumed, on average, up to about 73% less energy than SPA.

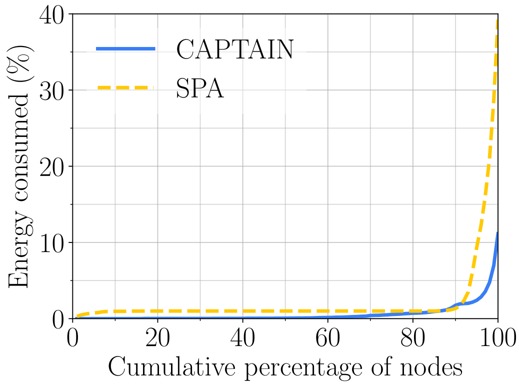
Figure 4.9 compares the number of acoustic and optical transmissions performed



(a) No clusters



(b) Clusters of 5 nodes



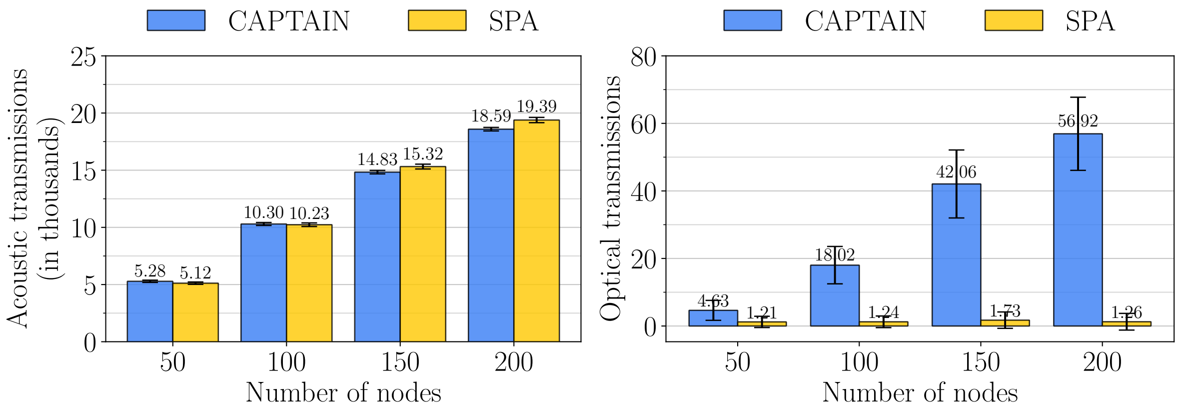
(c) Clusters of 10 nodes

Figure 4.8: Cumulative percentage of nodes that spent a portion of their energy during the period of network activity.

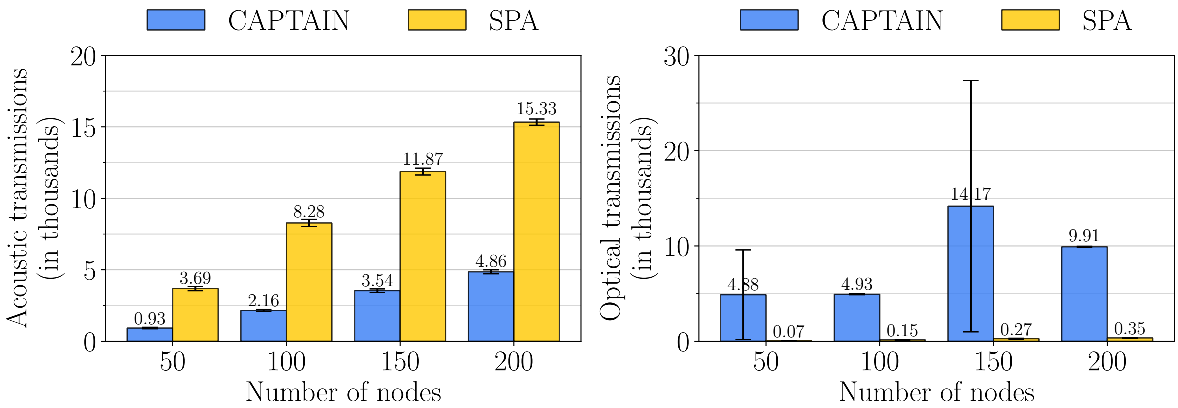
	Number of nodes in clusters		
	0	5	10
CAPTAIN	2.902 ± 0.046	0.978 ± 0.034	0.605 ± 0.028
SPA	2.952 ± 0.055	2.604 ± 0.072	2.280 ± 0.092

Table 4.2: Average network energy consumption (in percentage).

by the network nodes. In the scenarios with no clusters (Figure 4.9a) the numbers of both types of transmissions were very similar for both CAPTAIN and SPA. That explains the very similar curves for energy consumption in Figure 4.8a. The reason for such similarity is that the simple random node deployment does not favor the existence of clusters in the network and this forces almost every node to use only acoustic transmissions It also does not favor the use of data aggregation in CAPTAIN, since almost every node will be a cluster head.



(a) No clusters

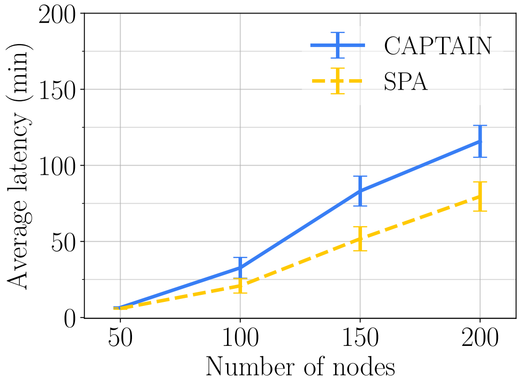


(b) Clusters of 10 nodes

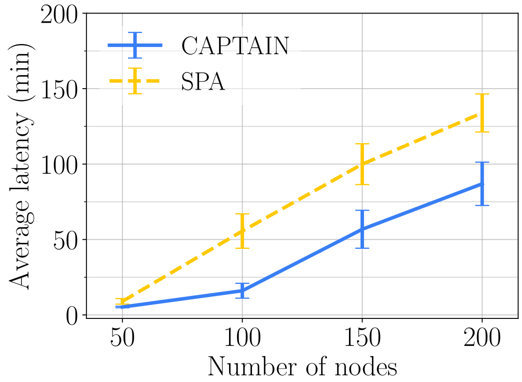
Figure 4.9: Comparison of the number of acoustic and optical transmissions performed by the network nodes.

When used in networks with clusters of 10 nodes (Figure 4.9b), CAPTAIN leads to fewer acoustic transmissions and more optical transmissions than SPA. This situation was good for the network energy, as can be seen in Figure 4.8c and Table 4.2, since acoustic transmissions are much more power hungry than the optical ones.

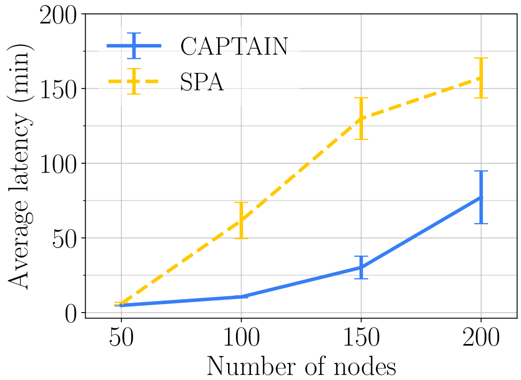
Next, we verified the network latency in all scenarios. Figure 4.10 shows, on average, how much time was required for the collected data to arrive at the sink node with CAPTAIN and SPA. CAPTAIN produced lower latency values than SPA when there were clusters in the network (see Figures 4.10b and 4.10c). In the worst case (200 nodes with no clusters), the use of CAPTAIN led to an increase of about 45% in the average latency, while in the best case (100 nodes with clusters of 10 nodes), the average latency was almost 83% lower. The existence of clusters favors both the use of optical communication, that is faster than the acoustic, and the data aggregation scheme, which reduces the number of messages in the network. When there were no guaranteed clusters in the network (Figure 4.10a), CAPTAIN could not be very effective, and its use has resulted in latency values at least as high as the ones from SPA.



(a) No clusters



(b) Clusters of 5 nodes

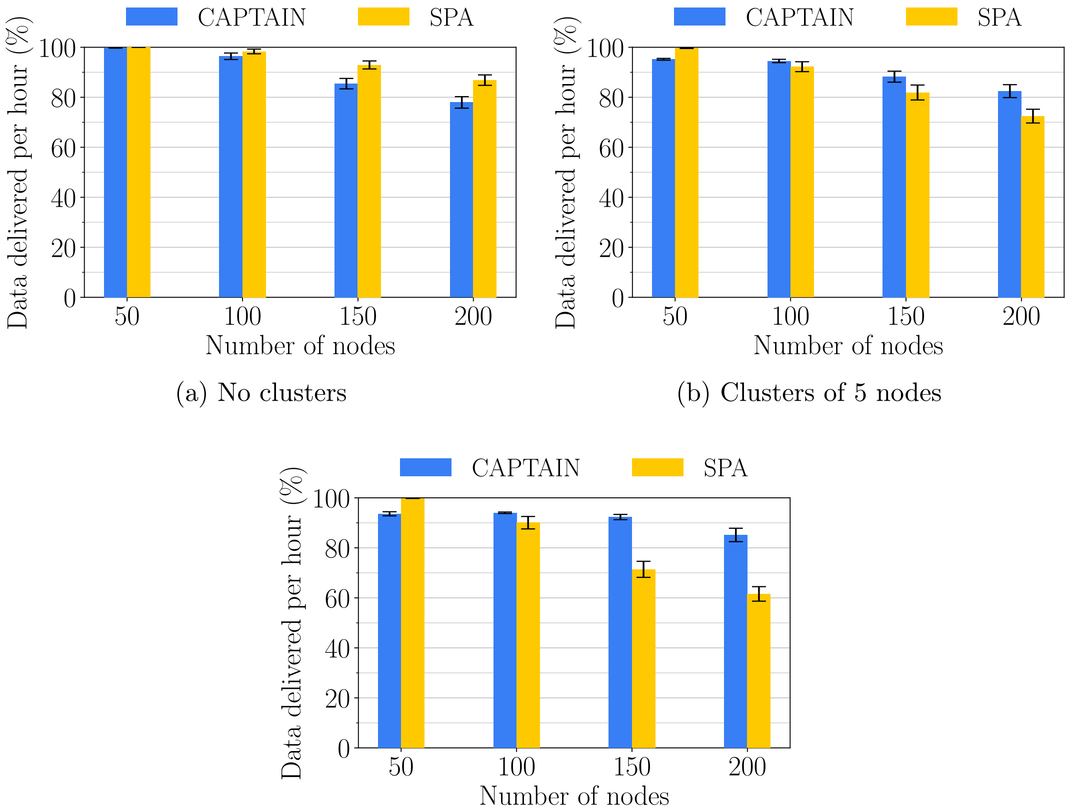


(c) Clusters of 10 nodes

Figure 4.10: Comparison of average latency observed.

The observed latency values were very high in all the simulated scenarios. We believe that the use of the TDMA system together with the simple way of distributing the time slots was the main reason for the high values since it does not allow a node to transmit outside its time slot, even though no one else wanted to. As some nodes will have the secondary task of forwarding data messages from others, they might receive more messages than they can forward in their time slots within the time period between two successive data collections. This problem results in messages overloads in some nodes and may increase the network latency.

We also verified how much of the collected data was delivered to the sink node. As CAPTAIN uses data aggregation, we considered the delivery of a message containing the aggregated data from n different sources as the delivery of n data messages. Figure 4.11 shows, on average, the percentage of collected data delivered to the sink node per hour. We can note that the amount of data delivered decreased with the increase in the network size for both algorithms. This result might also be a consequence of message overload in some nodes caused by the use of the TDMA system.



(a) No clusters

(b) Clusters of 5 nodes

(c) Clusters of 10 nodes

Figure 4.11: Comparison of the percentage of data collected that was delivered to the sink node per hour, on average, using both algorithms.

We can also note that CAPTAIN did a better use of the network clusters than the SPA. Figures 4.11b and 4.11c show that CAPTAIN delivered higher percentages of data per hour than SPA in the simulated networks, with the exception of those with 50 nodes. This advantage was possible due to the use of the data aggregation scheme, which allowed lower message traffic in the network. The data aggregation scheme also decreased the percentage of data delivered when the cluster size was increased, mainly in the scenarios with 50 nodes, where it fell from 99.8% (average value with no clusters) to 93.62% (average value with clusters of 10 nodes). The reason for such decreases is that the scheme demands the cluster heads to receive all data from their cluster to arrive before sending the aggregated data message to the sink node. Therefore, increases in the number of nodes per cluster may result in longer waiting times, increasing the time to deliver the messages.

Chapter 5

UW-SEEDEX: A Pseudorandom-Based MAC Protocol for Underwater Acoustic Sensor Networks

In this chapter, we propose a new MAC protocol for underwater acoustic sensor networks (UWASNs). The proposed solution, named UW-SEEDEX, is aimed at ad hoc UWASNs, i.e., networks where no central point exists for data collection, and sensor nodes organize themselves to be able to exchange data with any other. Also, UW-SEEDEX focuses on both static and semi-static networks, as its mechanisms are better used when the network topology does not change so often.

Inspired on the existing MAC protocol for terrestrial networks named SEEDEX [Rozovsky and Kumar, 2001], UW-SEEDEX avoids collisions with low overhead by employing random time slot schedules produced from seeds. Also, to cope with the characteristics of the underwater acoustic channel, UW-SEEDEX adopts a slightly different update scheme than SEEDEX and considers the propagation delay and acknowledgments when determining the time slot length. Through extensive simulations, we show that the protocol can deliver more packets than protocols such as Slotted FAMA [Molins and Stojanovic, 2006] and UW-Aloha [Peng et al., 2009] within the same time window, using, on average, fewer transmissions than both of them and with low energy consumption. We also extensively evaluated the protocol parameters via simulations. UW-SEEDEX presented reception rates close to 100%.

While protocols based on channel access disciplines such as Multiple Access Collision Avoidance (MACA) and Floor Acquisition Multiple Access (FAMA) require nodes to explicitly reserve the channel before every transmission, UW-SEEDEX avoids collision by using few periodic control transmissions. The proposed MAC protocol employes random time slot schedules that are composed of states where nodes can transmit data or must only listen for transmissions. The schedules are produced using seeds and pseudorandom number generators so that nodes can predict each other's schedules after exchanging only their seeds. Then, nodes can check their neighbors' schedules and find the best opportunities to transmit packets to them. Thus, improving network efficiency.

This chapter is organized as follows. In Section 5.1, we describe the main ideas of the SEEDEX protocol. Section 5.2 presents UW-SEEDEX in detail. In Section 5.3, we explain how we tested the performance of the protocol and show the simulation setup used. Section 5.4 shows the simulation results when testing UW-SEEDEX in different scenarios and checking its performance against two other well-known MAC protocols for UWSNs.

5.1 SEEDEX

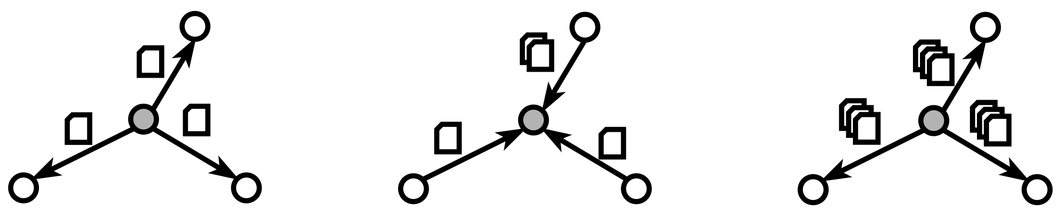
As briefly discussed in Section 3.2, SEEDEX is a MAC protocol for terrestrial ad hoc networks that served as an inspiration for our proposed solution. Its main goal is to avoid collisions without explicit reserving the channel for every packet as some protocols such as MACA [Karn, 1990] and FAMA [Fullmer and Garcia-Luna-Aceves, 1995] do. SEEDEX employes random schedules driven by pseudorandom number generators so that nodes can easily publish their schedule just by sharing their seeds. Therefore, nodes can opportunistically decide when to transmit.

The schedules are sequences of slots that define two possible states for nodes: "Listening" (L) or "Possibly Transmitting" (PT). Nodes must remain silent (do not transmit) when in L states, only listening for new packets. In PT states, they may send packets to others.

The key idea of SEEDEX is to use random time slot schedules that are created based on pseudorandom number generators. Each node initially chooses a seed and then uses it together with some method to generate its slot schedule. One way to produce the schedules is to use a Finite State Machine with states labeled as L or PT, and a pseudorandom number generator to drive the transitions. Another is to employ a Bernoulli process, where each slot state is selected based on a probability parameter p of it to be, for example, a PT one.

By using SEEDEX's key idea, nodes can determine the entire schedules of others just by knowing their seeds and how they generate them. If all nodes use the same generation method, then they only need to exchange their seeds.

After exchanging seeds with its neighbors, a node can predict their schedules and then find suitable slots to transmit packets to them. So, SEEDEX demands nodes to broadcast their seeds (and maybe states) and all the seeds (and also states) they know from their neighbors to all their neighbors. To do so, they use what the authors called a fan-in and fan-out procedure: nodes first broadcast all the seeds they know (fan-out) and listen to the broadcasts of others (fan-in), and then they broadcast their known seeds again. This procedure, illustrated in Figure 5.1, may be periodic to cope with node mobility and keep the information up to date. It is important to observe that nodes must always broadcast the current state of their random number generators, and that of their neighbors, instead of their initial state, which may have occurred at any time in the past.



(a) Step 1: Node broadcasts the information it knows to its neighbors.

(b) Step 2: Node listens to other broadcasts.

(c) Step 3: Node broadcasts, again, the information it knows to its neighbors.

Figure 5.1: Steps of the fan-in and fan-out procedure used for seed dissemination. Ideally, nodes know the seeds of their one-hop neighbors after the second step and the seeds of their two-hop neighbors after the third.

After knowing the seeds of all the nodes in its two-hop neighborhood, a node is able to decide the good moments to transmit packets. Consider the network portion illustrated in Figure 5.2 and suppose that node S wants to send a packet to node D. The transmission can only happen in slots in which nodes S and D are, respectively, in states PT and L. S can use its seed and D's seed to predict both slot schedules and thus easily find such slots.

Once S finds a suitable slot, it can also check the schedules of D's neighbors to determine how many of them will also be in the PT state and so detect the possibility of transmission collisions. If at least one more neighbor of D is going to be in PT state, then there is a chance of a collision to happen, and S should consider sending the packet in another slot. However, as nodes in the PT state do not necessarily transmit, SEEDEX defines that S should send the packet with probability inversely proportional

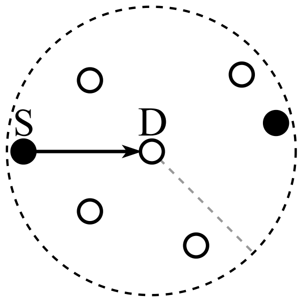


Figure 5.2: Example of forecasting the states of the nodes neighboring a node D in a network. Black nodes are in PT state, while the others are in L state.

to the number of D's neighbors in the PT state. Therefore, after finding a suitable slot, a node should transmit the packet with probability equal to

$$p_t = min\left(\frac{\alpha}{n+1}, 1\right),$$

where n is the number of other neighbors of the destination also in the PT state, α is a parameter used to control how aggressive a node can be while trying to transmit, and the min function restricts the probability values to valid ones.

Returning to the example in Figure 5.2 and checking the predicted state for the nodes, it is possible to see that only one node besides S will be in the PT state (the other black node). So node S will transmit in the slot with $p_t = \frac{\alpha}{2}$ (or 1 if $\alpha \geq 2$). If S decides not to send the packet to D in this slot, then it must repeat the process of finding another suitable slot and trying to transmit until it can be successful.

5.2 UW-SEEDEX

SEEDEX main idea of employing random schedules created from seeds allows nodes to plan their transmissions to avoid collisions, with low overhead. These benefits are very attractive for UWASNs, where communication usually is energetically expensive and has a long propagation delay. The solution proposed here, named UW-SEEDEX, uses the same idea as SEEDEX to provide a MAC protocol for UWASNs that has low overhead and aims to avoid transmission collisions and retransmissions, and therefore save energy. The MAC protocol differs from SEEDEX as it considers the particularities of the underwater channel, such as the long propagation delay, in the process of information dissemination and in the slot schedule design (as will be shown later).

The development of UW-SEEDEX considered a failure model where links could occasionally fail to deliver messages. The model also takes into consideration that nodes can fail temporarily or permanently, becoming unavailable to the rest of the network for some time or forever.

UW-SEEDEX uses slot schedules for packet transmissions. By doing so, nodes can avoid having to explicitly reserve the channel before transmitting as, for example, Slotted FAMA [Molins and Stojanovic, 2006] does by using RTS-CTS-DATA-ACK sequences. This is advantageous as frequent message exchanges can be harmful to network performance due to the small acoustic channel capacity [Jiang, 2018; Chen et al., 2014].

UW-SEEDEX is a TDMA-based protocol, and so it requires SYN. As stated earlier, precise SYN is challenging in UWASNs, but many solutions already exist in the literature [Syed and Heidemann, 2006; Liu et al., 2013].

To generate the slots schedules and to disseminate information, nodes need to keep a small database with the information they know about their neighbors. This database should mainly contain the data required to generate the slot schedules, such as the seeds. This data must be kept up to date to ensure nodes can correct forecast their neighbor's slot schedules. Otherwise, it can negatively affect the transmission planning of the nodes.

The UW-SEEDEX operation cycle, shown in Figure 5.3, is composed of two components: an update and a communication interval. Update intervals are periods where nodes exchange information about their seeds and states so they can better plan their future transmissions. Communication intervals are periods formed by pieces of time slot schedules, where nodes can transmit and receive data packets.

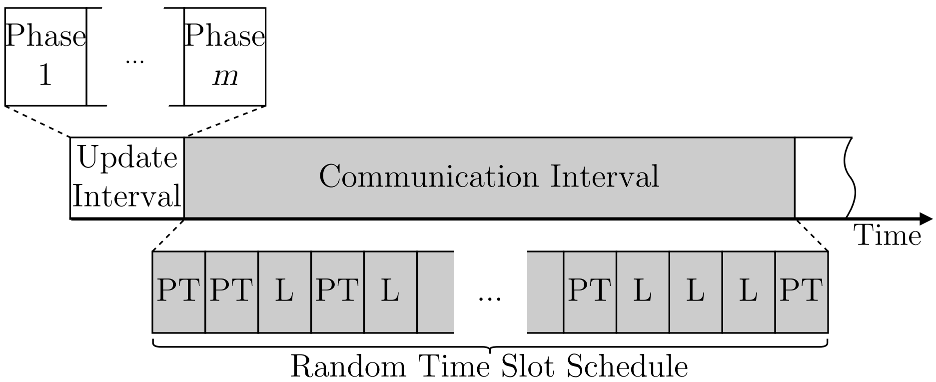


Figure 5.3: UW-SEEDEX operation cycle.

The two intervals are interspersed over time, with an update interval always preceding a communication one. Next, we give more details about these two intervals.

5.2.1 Update Intervals

As previously mentioned, the update intervals are periods where nodes will exchange information about their seeds and states to keep their databases up to date. They do this by broadcasting information packets to their neighbors while listening to broadcasts from others.

The information packets should contain the data about the sending nodes and all their one-hop neighbors they know. They can be created from the node database by, for example, organizing the stored information as a sequence of tuples containing nodes identifier (such as their MAC addresses) and their (current) seeds. Whenever a node receives an information packet from one of its neighbors, it must use it to update its database by adding new information to it and updating old ones.

Nodes can also use the sending of information packets to detect the failure/absence of their neighbors. If a node does not receive any information update from one of its neighbors for some time, then it may assume that it is not part of the network anymore. The time for this detection can be just some update phases or even some update intervals. Using only a few update phases may lead to faster detection of unavailable nodes, but can also produce many misdetections and degrade performance. On the other hand, if nodes wait for some update intervals, then they may avoid misdetections at the cost of possibly slowing correct detections (mainly if communication intervals are too long). While the former approach should favor more mobile networks (mainly due to their frequent topology changes), the latter can be better for static networks.

Instead of using the fan-in and fan-out procedure to disseminate information like SEEDEX, UW-SEEDEX employes a similar scheme, but with multiple phases of transmitting and listening. The use of multiple phases aims to ensure the adequate dissemination of information since, as discussed in Section 2.1, the underwater acoustic channel has characteristics that impose difficulties on the correct reception of data.

Each update phase is a period where nodes broadcast their known information (seeds and states) once and listen to transmissions from others. As nodes need to access the medium to share their information, they must follow some scheme to do so. Each node could, for example, randomly choose a time within the update phase to transmit its message. Another option would be to divide the update phase into slots and then distribute them among the nodes. In this work, we consider the use of the first option as it is simple and does not require any previous message exchange, even though transmission collisions are likely to occur.

Ideally, only two update phases would be required for nodes to know the information of their two-hop neighbors. However, as acoustic communication faces high bit error rates in underwater environments [Akyildiz et al., 2004] and transmission collisions may also occur, nodes may need more than two phases to get information about all their neighbors. Therefore, each update interval consists of m phases.

If the network topology does not change very often, then the update intervals are especially relevant at the beginning of network operation. Later updates should be most useful for correcting possible errors that nodes can have in the perceptions of the pseudo-random generator states of others. So, instead of keeping m constant, the number of update phases could be reduced over time until reaching a minimum value. This reduction could benefit the network, reducing MAC overhead as fewer control transmissions would be made.

5.2.2 Communication Intervals

Communication intervals are sequences of time slots that nodes can use to transmit or receive packets. These slots define PT or L states, just as in SEEDEX. Nodes in the PT state may transmit data, while nodes in the L state should only listen for packets.

Nodes create their slot sequences based on their seed. Like in SEEDEX, they use their seeds to initialize random number generators, which are then used with some method to create time slot schedules. Then, they produce their communication intervals by dividing their slots schedules in sequences of the same pre-defined size. Another way to visualize this generation process is to consider that nodes have different seeds, and maybe different start states, for each of their communication intervals. Thus, after finishing an update interval, a node uses its current seed to produce the slot sequence for the following communication interval, and then it updates the seed for the next one.

After generating a communication interval, nodes are almost ready to plan transmissions to occur in it. Before doing this, nodes must also predict the communication intervals of all their neighbors up to two hops apart, using their most up-to-date information from the last update interval. Then, they can look for packets ready to be sent and try to find suitable slots to transmit them. If no suitable slot is found for some packet in the current communication interval, then its transmission may be delayed for the next one. The same procedure should also be applied for new packets entering the sending queue during the communication interval.

The length of the slot sequences should depend on the network conditions. Long sequences, for example, may produce more transmission opportunities per communication interval and less frequent information updates, which can directly impact factors such as network delay. On the other hand, sequences with few slots do the opposite of that, and may also result in energy waste if topology changes are not too frequent in the network.

In addition to defining the length of communication intervals, it is also necessary to establish the length of the time slots. They must be at least as long as the time required for sending one data packet and, if confirmation is required, also receiving an acknowledge (ACK) for it. Therefore, it must take into account:

- the maximum time to transmit a packet: can be estimated using a predefined maximum packet size and the transmission rate;
- the maximum propagation delay: can be estimated using the maximum transmission range and the speed of sound in water (about 1500 m/s); and
- the maximum time to transmit an ACK: can be estimated using the maximum ACK size and the transmission rate.

A guard time must also be added to the slot length to account for possible variations in the propagation delay and any clock drift. Figure 5.4 illustrates the composition of a time slot where one packet and one ACK can be transmitted.

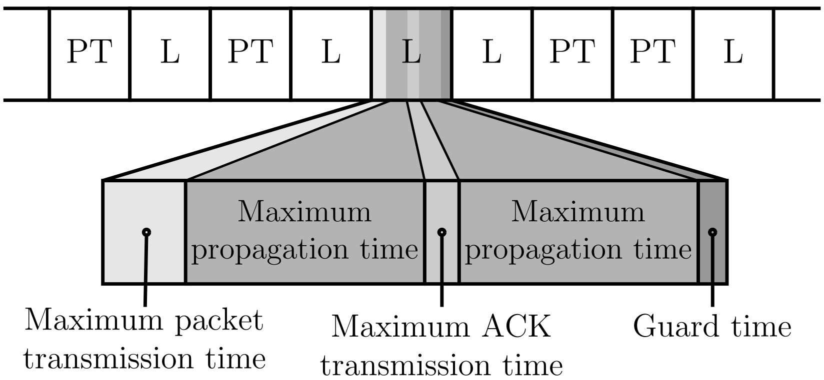


Figure 5.4: Time slot length required to send 1 packet and receive 1 ACK.

Unlike SEEDEX, UW-SEEDEX considers the time for transmitting ACKs inside the same time slots of data packet transmissions to avoid possible very long waits for confirmations. If ACKs were to be transmitted alone in other slots, the low capacity of underwater acoustic channels, coupled with the random nature of the protocol, could make waiting for confirmation take too long and thus impair applications that need them.

5.3 Simulations Setup

We used simulations to evaluate the performance of UW-SEEDEX in multiple test scenarios. We implemented UW-SEEDEX[1], as described in Section 5.2, using the ns-3.30 simulator [Riley and Henderson, 2010]. In this implementation, the number of update phases decreases by one at each new update interval. It starts with m phases and decreases until it reaches only two. The update interval length remains constant during the network operation, which means that as the number of phases decreases, they last longer.

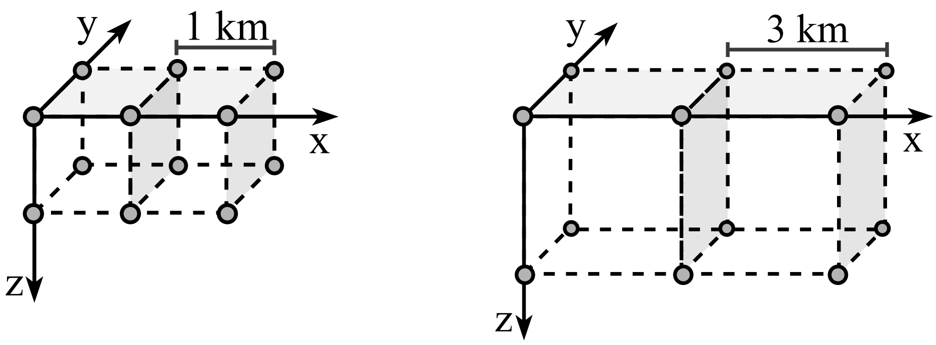
We also implemented two other well-known underwater MAC protocols for comparisons: Slotted FAMA and UW-Aloha. The Slotted FAMA implementation uses RTS-CTS-DATA-ACK sequences for data transferring, trains of packets to send multiple DATA packets with a unique handshake (but with acknowledgment of one at a time), random backoff scheme when packets are lost, and the slot length is defined as the sum of the maximum propagation delay with the maximum DATA packet transmission time and the guard time. The UW-Aloha implementation tries to send new packets as soon as they are available, uses ACKs to detect collisions, and uses a Binary Exponential Backoff to retransmit lost packets (limited up to a certain maximum number of retransmissions). This Aloha implementation does not use time slots as studies show that Slotted Aloha does not have better performance than the pure Aloha in underwater environments [Vieira et al., 2006; De et al., 2011].

All the tests simulated 24 hours of network operation. We considered nodes to generate packets to other randomly chosen nodes following Poisson distributions, whose default average was one packet per 10 minutes. The data packets had a 100 bytes long payload, as in [Coutinho et al., 2017], and we set the maximum packet size, required to estimate the maximum transmission time to 110 bytes. The bit rate was set to 5 kbps based on the WHOI micro-modem [Freitag et al., 2005], which is already implemented in the ns-3. Since all three protocols require an estimate of the maximum propagation delay (whether to calculate the slot length or to set a timeout for ACK reception), we used a transmission range of 3 km (achieved with a 148 dB transmission power) and considered the speed of sound in water as 1500 m/s.

We used two different methods to deploy nodes when simulating the networks. One method was a random deployment of nodes in a $10 \times 10 \times 10$ km area, with the restriction of producing only connected networks. The other method consisted of deploying nodes following a grid structure with the same spacing in all dimensions.

[1]All protocols implementations used here are available at `https://gitlab.com/epmcj/ns-3-dev/-/tree/new-uan-mac-protocols`.

Through changes in the grid spacing, this method allows better control of some network conditions, such as the connectivity, than the random one. For example, both networks in Figure 5.5 follow grid structures and have the same number of nodes, but while the network in Figure 5.5a is fully connected, nodes in the network in Figure 5.5b have degrees of connectivity equal either to 2 or 3. We used $4 \times 3 \times 3$ grids (36 nodes) with a default spacing of 2 km in the simulations.



(a) Grid spacing equal to 1 km

(b) Grid spacing equal to 3 km

Figure 5.5: Networks deployed in grids with different node densities.

The underwater acoustic channel used in the simulations was one of those already available in ns-3, which follows the Thorp's propagation and noise models described in Section 2.2.1. It determines the success of transmissions by estimating the packet error rate (PER) based on the signal-to-interference-plus-noise ratio (SINR), as it considers possible interference from other transmissions. Simulations considered a practical spreading ($k = 1.5$), no shipping activity ($s = 0$), and that the wind speed was 0 m/s ($w = 0$).

As no routing algorithm has been taken into account, the simulations considered that each node was able to choose the paths with the shortest distances to the others. Simulations have also not take into account any time synchronization service, so we assume that nodes are synchronized and, if they desynchronize, they can use a synchronization protocol. Also, both UW-SEEDEX and Slotted FAMA use guard times to account for possible clock drifts.

The energy consumption of the nodes was accounted for based only on that of their simulated acoustic modems. The simulated modem energy model is based on the WHOI micro-modem [Freitag et al., 2005], which was already implemented in the ns-3. Its power consumption values are 50 W when transmitting, 158 mW when receiving or in idle state, and 5.8 mW when in the sleep state. Nodes started the simulations with an initial energy of 1 MJ.

In addition to energy consumption, we also evaluated the following metrics:

- End-to-end delay: the average time elapsed between the generation of a data packet and its reception by the destination;
- Reception rate: the percentage of data packets generated that were successfully received by the destinations during the simulation time;
- Transmissions per data reception: the average number of transmissions per each data packet received.

All the results correspond to average values of 50 executions, and they present a 95% confidence interval.

5.4 Simulation Results

In this Section, we first analyze how UW-SEEDEX's parameters affect its performance. Then, we evaluate how the protocol performs against UW-Aloha and Slotted FAMA.

5.4.1 Evaluating UW-SEEDEX parameters

To evaluate how each UW-SEEDEX parameter affected its performance, we conducted simulations using four different scenarios. All the scenarios presented nodes distributed in grids and, while two of them had different traffic conditions, the other two presented different network densities. Table 5.1 shows the default values of the UW-SEEDEX parameters used in the simulations.

We changed the network traffic by modifying the average value of the Poisson distribution used to generate new packets. Based on preliminary tests with the three protocol implementations, we defined two different traffic conditions: a high traffic scenario, where the average was one packet per five minutes, and a low traffic scenario, with an average of one packet every half hour. We note that, as will be seen below, what we define here as high traffic is not necessarily high for all evaluation scenarios.

For the other two scenarios, we changed the network density by using different grid spacing values. We used grid spacing values of 1 and 2.5 km to produce, respectively, dense and sparse networks. The 1 km spacing value produced networks with fully connected topologies, and the 2.5 km created networks with grid topologies.

MAC Parameter	Value
p	0.5
α	8
Update interval length	30 s
Number of update phases	4
Communication interval size	4096 slots
Guard time	0.2 s
Time slot size	≈ 4.386 s

Table 5.1: Default values used for evaluating UW-SEEDEX parameters.

5.4.1.1 Probability p

We first investigated how the value of p, the probability for PT states, affects the UW-SEEDEX performance. Figure 5.6 shows the performance results that we observed when we varied the value of p between 0.1 and 0.9 in the two different traffic condition scenarios[2]. The first point to notice is that there seems to be an optimal value for p and that it depends on the traffic conditions. The observed optimal value of p was close to 0.4 in the high traffic scenarios, and around 0.5 when the network traffic was low. We can also notice that changes in p affected the evaluated metrics more when the volume of messages in the network was high, as can be seen in the poor performances when $p \geq 0.6$.

Figure 5.7 also indicates the existence of optimal values of p that depend on the scenario conditions, in this case, on node density. It additionally shows that p has more influence over the protocol performance in sparse networks than in dense networks. One factor that may explain this result is the length of the paths taken by the packets. We could verify that the average path length was around 1.13 in the dense networks, and close to 3.11 in the sparse ones. Long paths require packets to travel through more hops than short paths, which can end up creating more traffic and thus degrade network performance.

5.4.1.2 Parameter α

We next verified the performance of UW-SEEDEX for different values of α, the parameter, the parameter that controls how aggressive nodes can be while trying to transmit. Figures 5.8 and 5.9 show how changing the value of α, between 1 and 15, affected the performance metrics. In low traffic conditions or dense networks, increasing α hardly

[2]Note that no transmissions will occur if $p = 0$ or $p = 1$ as these values produce only slots in one state: L or PT, respectively.

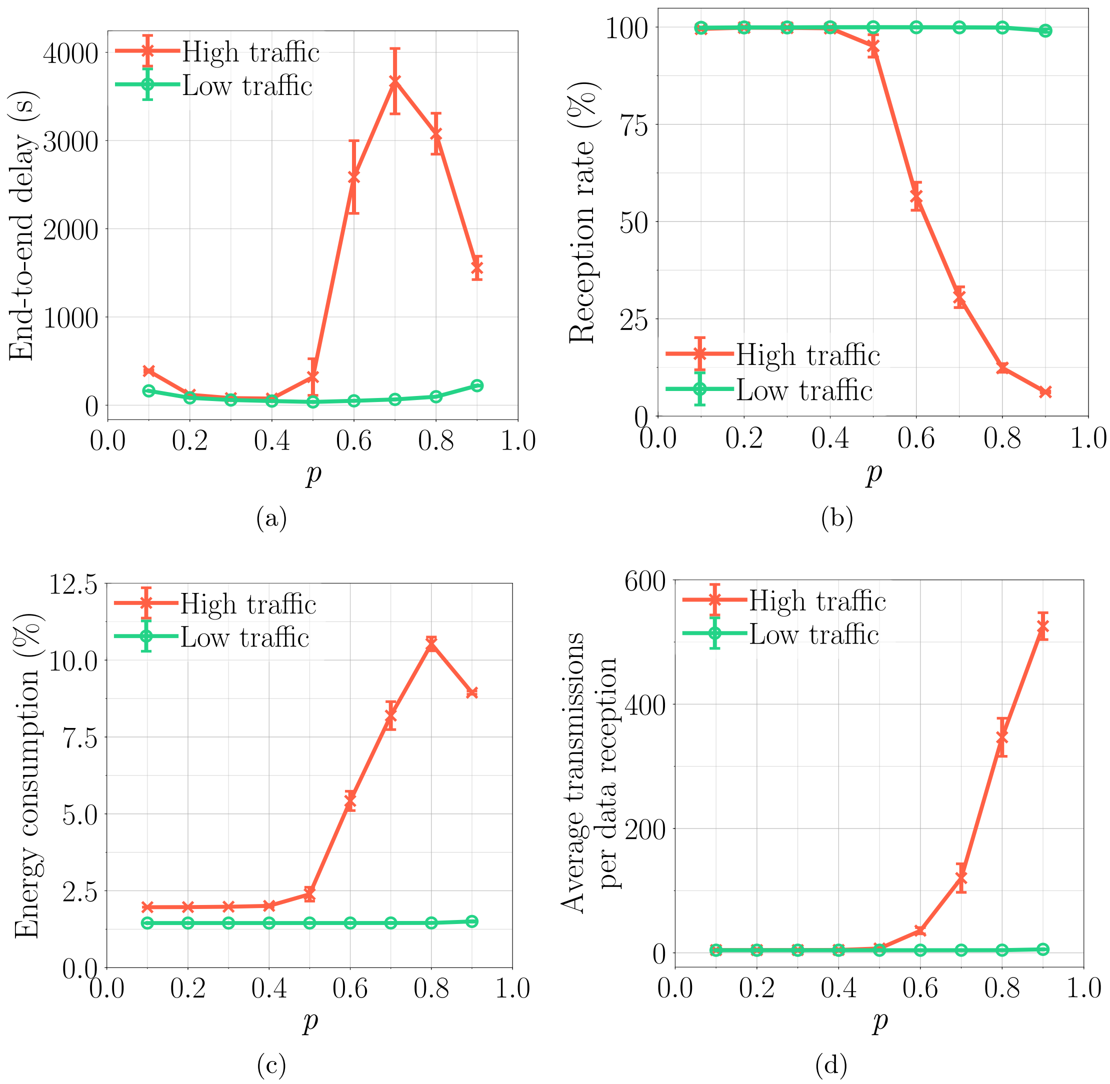


Figure 5.6: UW-SEEDEX performance as a function of p in both high and low traffic conditions.

affected both the network energy consumption and the average number of transmissions per data reception, and it slightly increased the reception rate and significantly decreased the end-to-end delay. These results indicate that, under these conditions, increasing α does not considerably increase the number of transmissions. Instead, it just made them happen earlier as it is directly related to transmission probability p_t.

Increasing α in sparse networks only improved the performance up to some point, from which the metrics values remained almost the same. Meanwhile, α seems to have optimal values by observing the results under high traffic conditions. When $\alpha = 5$, we registered the lowest average end-to-end delay, a high reception rate, and energy consumption and average transmissions per data reception values that were close to

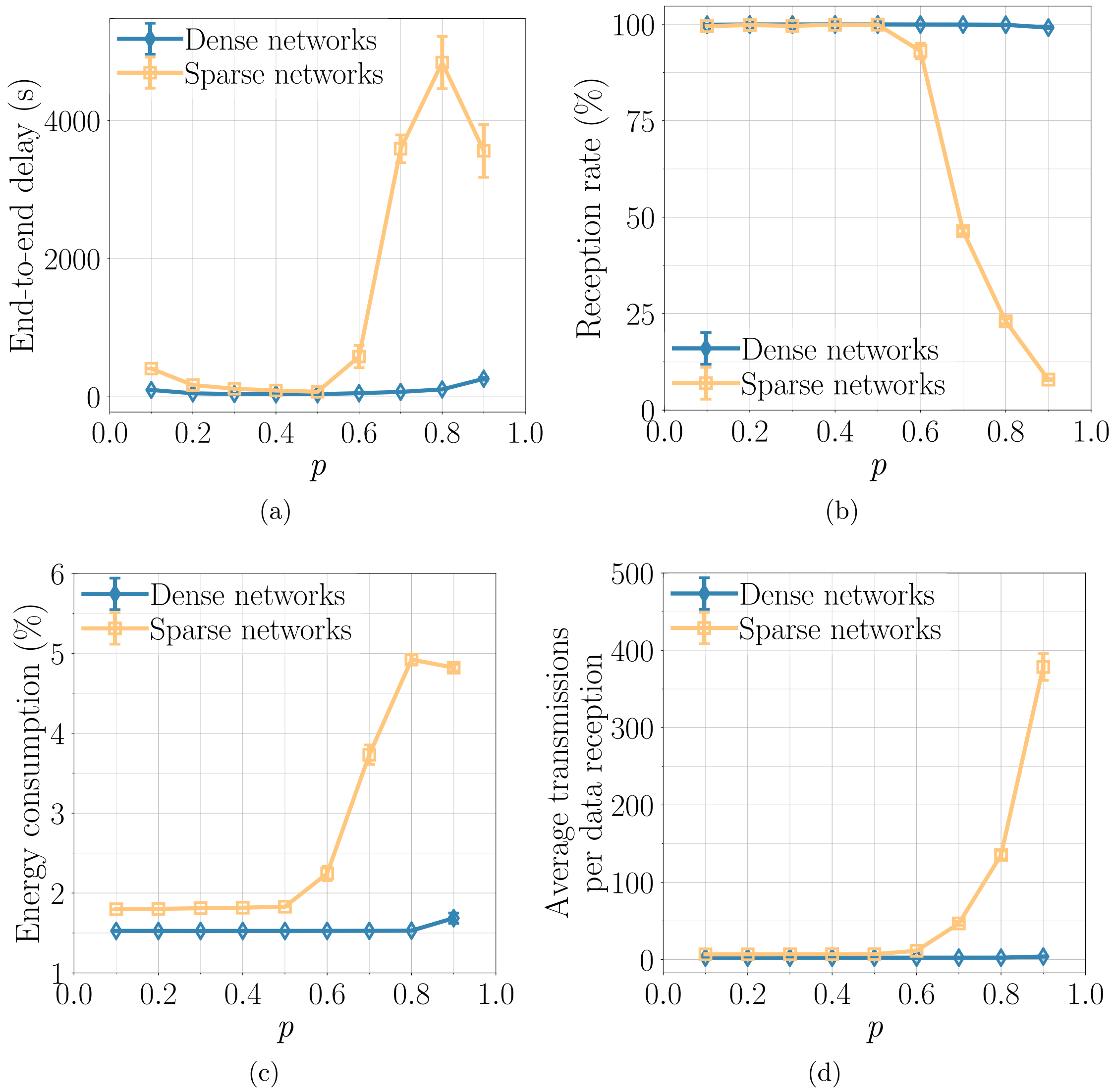


Figure 5.7: UW-SEEDEX performance as a function of p in dense and sparse networks.

the lowest observed (when $\alpha = 1$). Also, all the metrics registered great variance and relatively inferior performance for $\alpha > 7$, which indicates that the protocol should not be too aggressive under high traffic.

5.4.1.3 Update interval length

We next examined how the update interval length affects UW-SEEDEX performance by investigating intervals between 10 and 60 seconds. Results shown in Figures 5.10 and 5.11 indicate that a 15 seconds long update interval was enough in almost all scenarios, except in dense networks. In complete networks, the use of 20 seconds long intervals led to the best results, including the lowest average end-to-end delay. In the

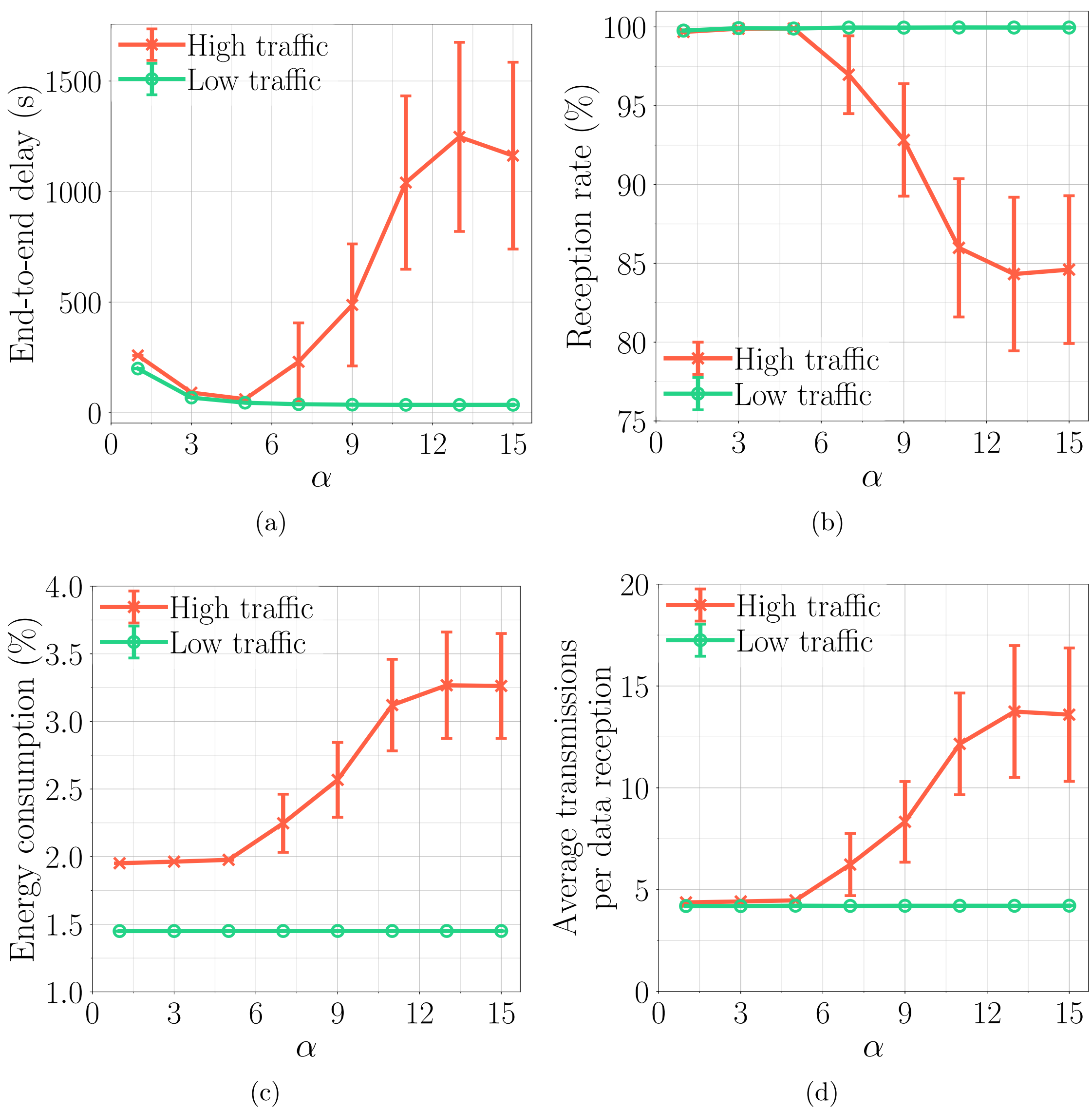


Figure 5.8: UW-SEEDEX performance as a function of α in both high and low traffic conditions.

other scenarios, intervals longer than 15 seconds did not improve the average protocol performance, and so they may waste time.

We can also see in both figures that 10 s update intervals produced the worst results in all scenarios. The main reason for such poor performances is that the update interval was too short. In our UW-SEEDEX implementation, where nodes randomly broadcast their information packets in each update phase, very small intervals may harm network performance as they may cause many collisions in the broadcasts. In this test case, the first update interval produced 4 phases with 2.5 s each, which led the nodes to choose a time to start their broadcasts only in the first 0.3 s of the phases

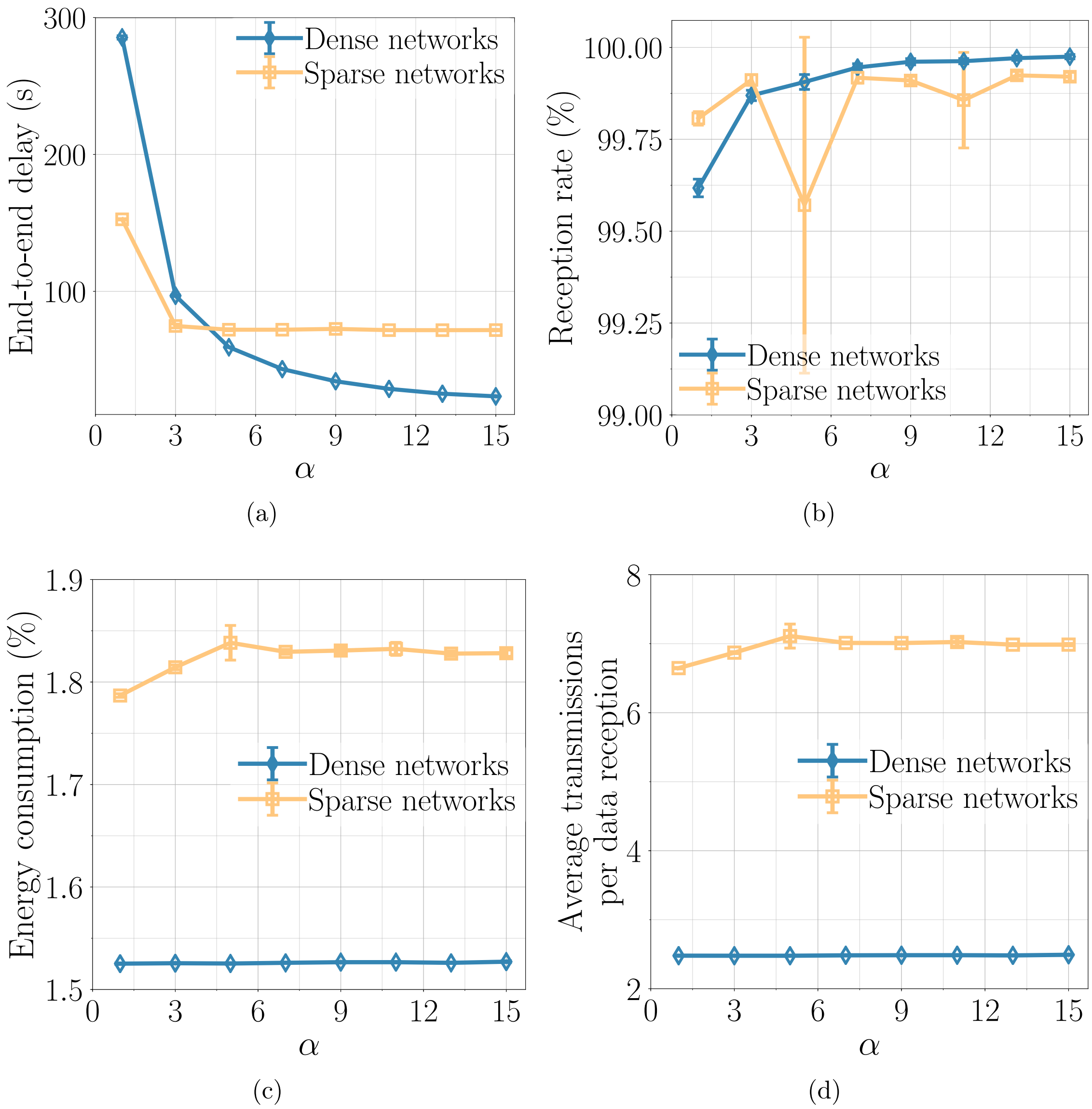


Figure 5.9: UW-SEEDEX performance as a function of α in dense and sparse networks.

to prevent packets from being received outside them.

5.4.1.4 Number of update phases

The other update interval parameter evaluated was the starting number of phases. Figures 5.12 and 5.13 show the results observed for different numbers of phases from 2 to 8. The first point to be noted is that the performance metrics presented high variations in high traffic scenarios, while the same did not happen in the other scenarios.

We can see in the other scenarios that the more update phases, the higher the network overhead, with more transmissions occurring per data reception. These addi-

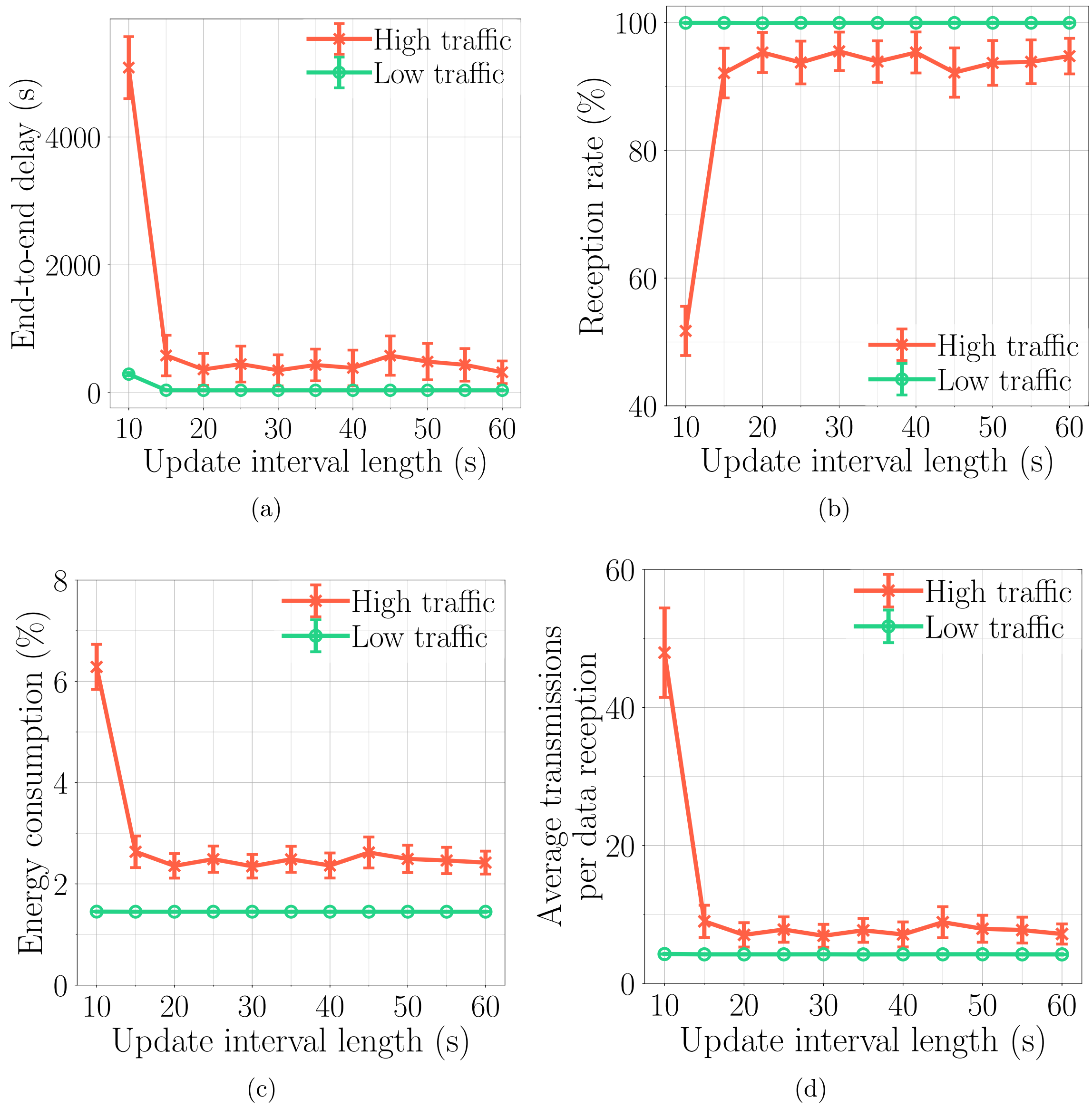


Figure 5.10: UW-SEEDEX performance as a function of the update interval length in both high and low traffic conditions.

tional transmissions slightly increased the energy consumption of the networks, while their reception rate was almost the same as that when the nodes started using only 2 phases.

Lastly, we could observe that the average end-to-end delay only decreased with the increase in the number of phases in dense networks. This fact indicates that having many update phases is better for networks where nodes have several neighbors. Nodes with a high degree of connectivity should listen to a lot of information packets at each update phase, which can make them undergo more reception errors than others due to

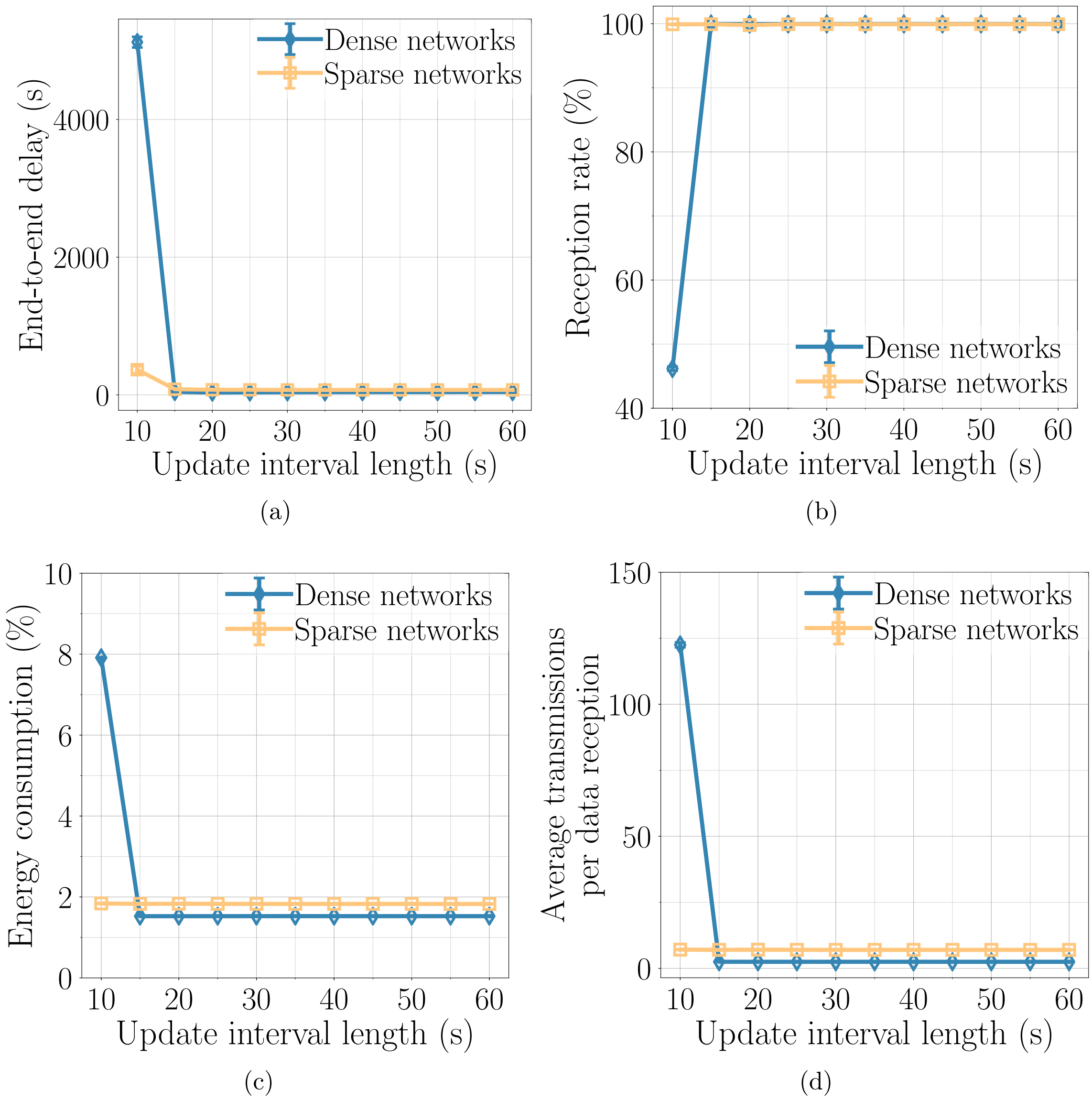


Figure 5.11: UW-SEEDEX performance as a function of the update interval length in dense and sparse networks.

transmission collisions. Therefore, having more update phases can help these nodes to discover their neighbors sooner, which can reduce the end-to-end delay at the cost of increasing energy consumption.

5.4.1.5 Communication interval size

Finally, we evaluated how the number of slots in communication intervals affected UW-SEEDEX performance. Figures 5.14 and 5.15 present the performance metrics for various communication intervals in the different simulation scenarios. It is possible to

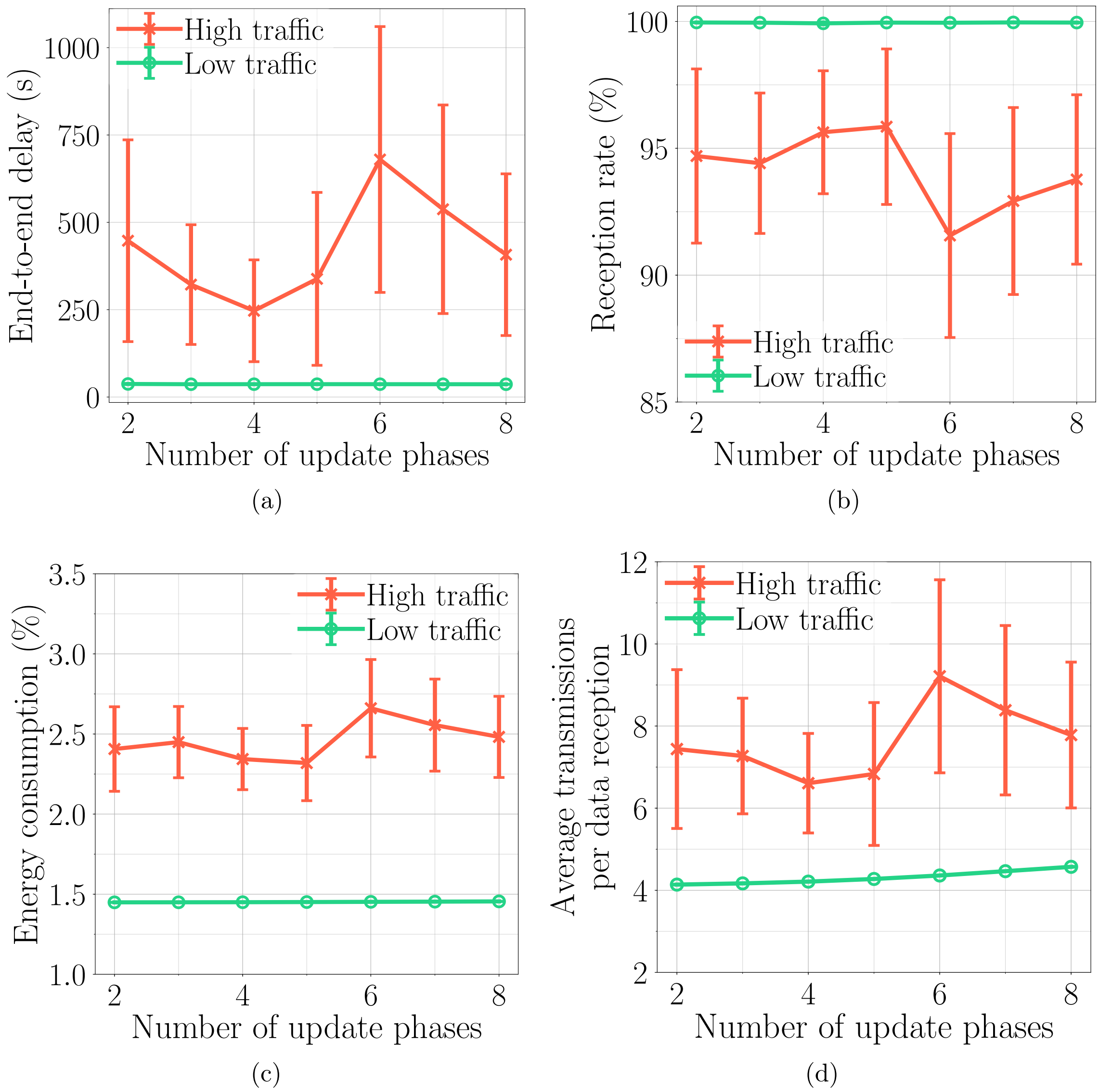
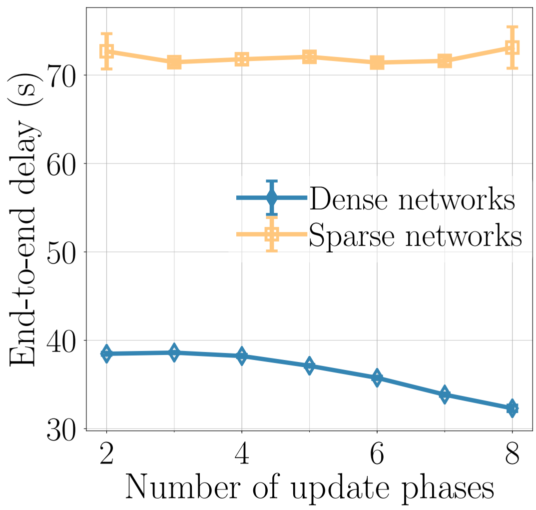


Figure 5.12: UW-SEEDEX performance as a function of the number of update phases in both high and low traffic conditions.

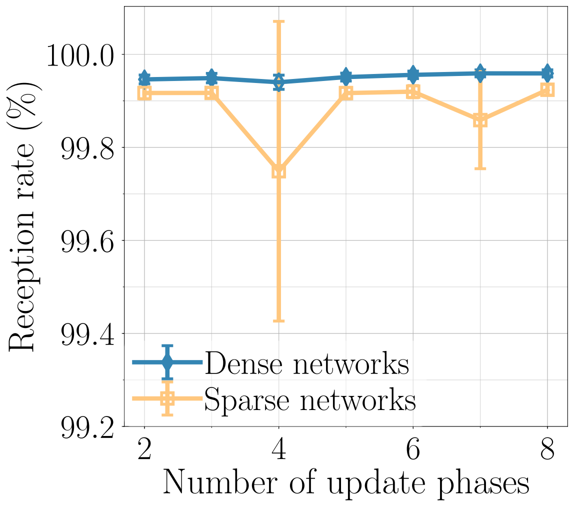
observe that the average number of transmissions per data reception decreased as the interval size increased. This result was expected, as longer communication intervals lead to less frequent update intervals and, consequently, to fewer control transmissions.

We can also observe lower average end-to-end delays and energy consumptions at longer communication intervals. In dense networks, the average end-to-end delay recorded was about 20% lower in intervals with 10240 slots than in those with 1024 slots. Again, we believe these results are related to less frequent update intervals.

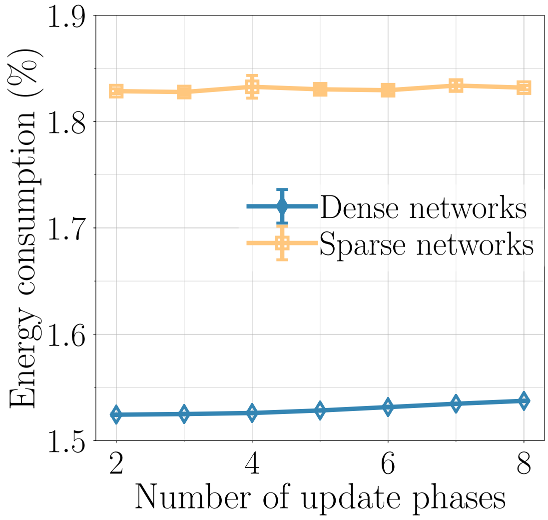
We note that we do not expect the use of long communication intervals to lead to lower delays in mobile networks. As the simulated networks were static, the firsts



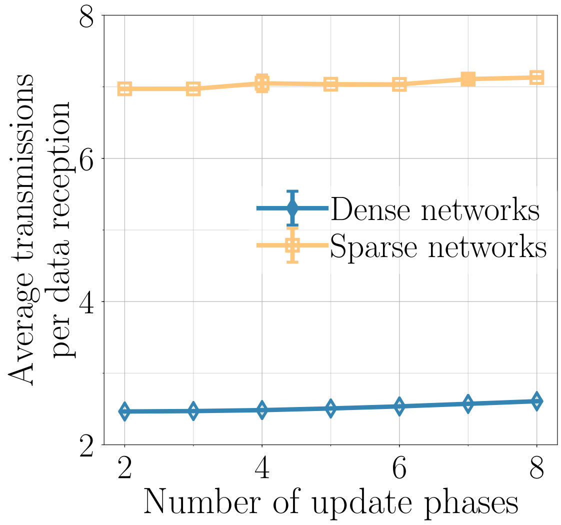
(a)



(b)



(c)



(d)

Figure 5.13: UW-SEEDEX performance as a function of the number of update phases in dense and sparse networks.

update intervals must have been enough to nodes to disseminate their information. However, mobility may change network topology and make less accurate the information that the nodes have, which can negatively affect transmission planning and, as a result, increase network delay.

5.4.2 Comparing UW-SEEDEX with other protocols

After evaluating how UW-SEEDEX's parameters affect its performance, we verified how it performs against two other MAC protocols: UW-Aloha and Slotted FAMA (S-

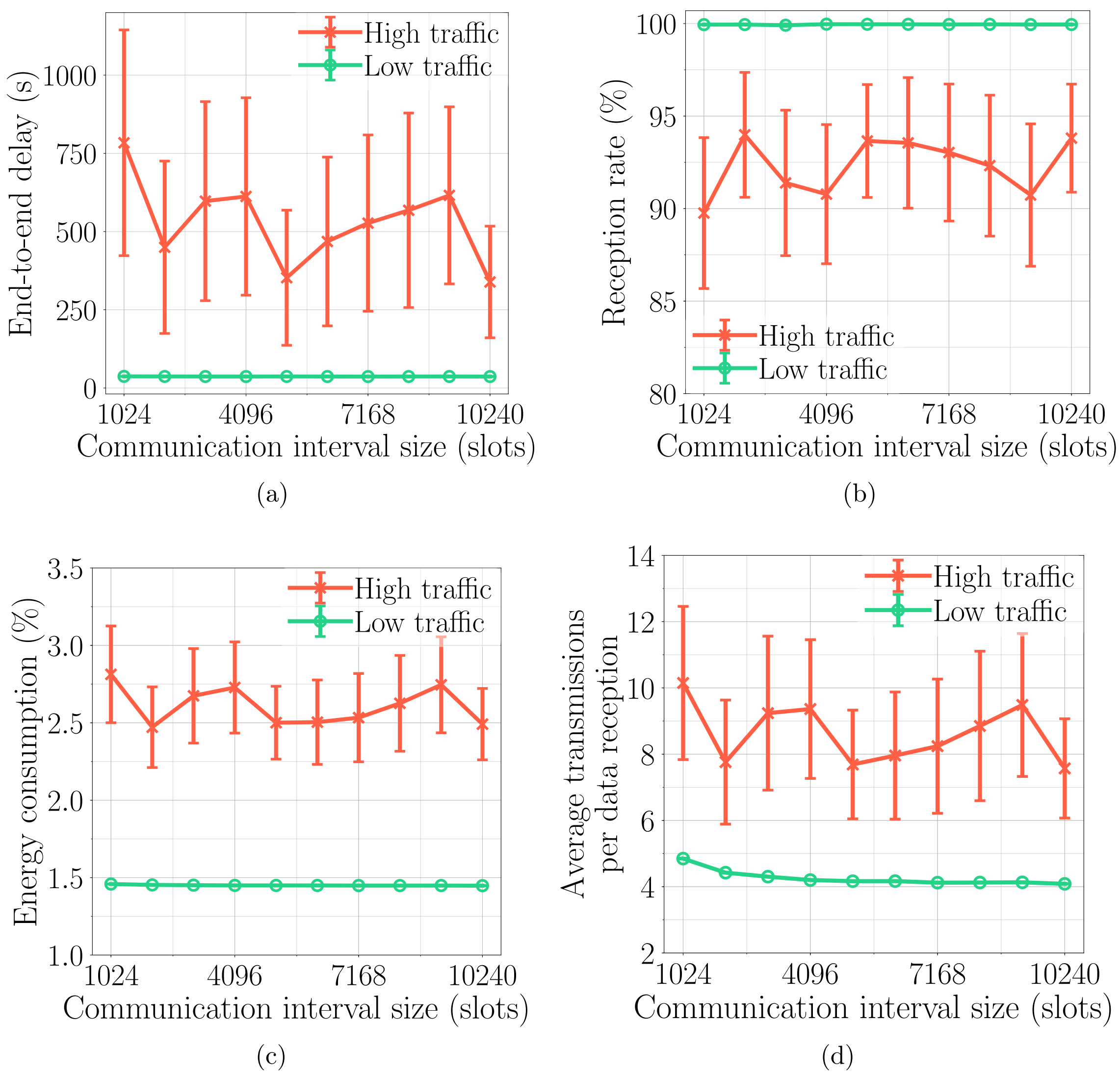


Figure 5.14: UW-SEEDEX performance as a function of the number of slots per communication interval in both high and low traffic conditions.

FAMA). Table 5.2 shows the default values used in the simulations for parameters of the three MAC protocols. We defined these values based on preliminary tests, where we searched for combinations of parameter values that lead to good performances of the protocols in different scenarios.

5.4.2.1 Network traffic

We first compared the three MAC protocols under different network traffic conditions and using the two methods for deploying the nodes mentioned in Section 5.3 (random and grid deployments). We varied the average period for nodes to generate new data messages from 5 to 30 minutes. Figure 5.16 and show the results we observed when

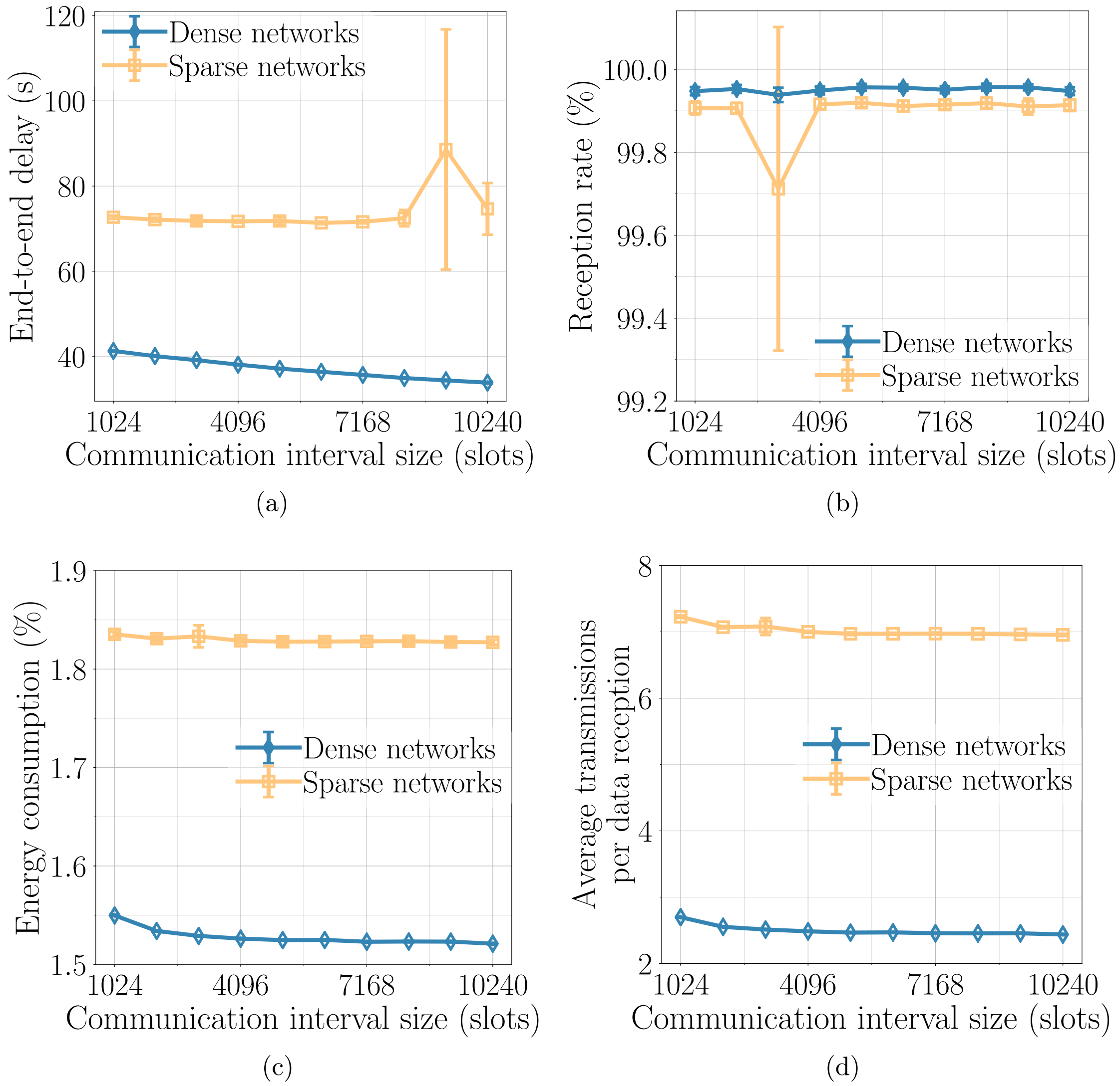


Figure 5.15: UW-SEEDEX performance as a function of the number of slots per communication interval in dense and sparse networks.

nodes were deployed in grids, while Figure 5.17 presents the results using the random deployment.

UW-SEEDEX was able to deliver almost the same number or more messages than both UW-Aloha and S-FAMA in both deployment scenarios using, on average, fewer transmissions. Its reception rate was very close to 100% in all traffic conditions, and its average number of transmissions per data reception was lower than those of the two other protocols. Meanwhile, UW-Aloha and S-FAMA also had high reception rates, but they were considerably lower than UW-SEEDEX's when packets were generated more frequently. For example, the difference between UW-SEEDEX and UW-Aloha's reception rate was up to 60% in scenarios with an average data generation period of

MAC Parameter	**Value**
UW-SEEDEX	
p	0.4
α	7
Update interval length	25 s
Number of update phases	7
Communication interval size	10240 slots
Guard time	0.2 s
Time slot size	$\approx$ 4.386 s
Slotted FAMA	
Maximum retransmissions	5
Maximum backoff	12 slots
Guard time	0.2 s
Time slot size	2.376 s
UW-Aloha	
Maximum retransmissions	5

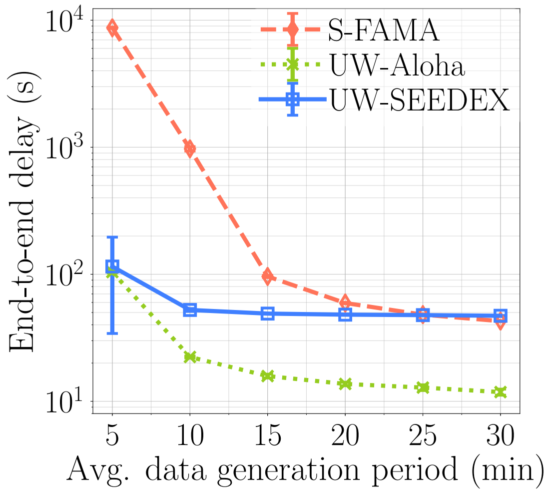
Table 5.2: Default values used for comparing the MAC protocols.

5 minutes. When the network traffic is high, the S-FAMA's use of RTS-CTS-DATA-ACK sequences seems to be too much overhead and the UW-Aloha strategy of almost sending packets as soon as possible seems to result in many collisions.

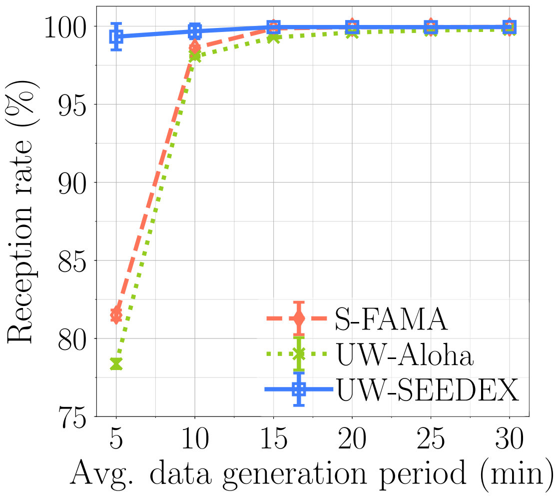
Simulations also showed that UW-SEEDEX and S-FAMA spent less energy than UW-Aloha. The two protocols had similar energy consumption in the test scenarios, but UW-SEEDEX delivered more packets than S-FAMA in some of them. In some cases, it delivered 15% more messages than S-FAMA using, on average, less than half the number of transmissions per delivery. We note that, as we did not consider the energy costs of a synchronization service required by both UW-SEEDEX and S-FAMA, UW-Aloha may consume less energy than both of them if the energy cost of using such a service in a real-life scenario is too high.

Figure 5.16a shows that UW-SEEDEX produced the lowest average end-to-end delay in networks with grid node distribution only in the scenarios with the highest frequency of data generation (average of 1 packet per 5 minutes). Meanwhile, Figure 5.17a shows that it produced the lowest average delays for networks with random deployment when the data generation period was lower than 10 minutes. In all other scenarios, UW-Aloha was the protocol that produced the lowest delays. This result is mainly due to the greedy nature of the Aloha protocol, which demands nodes to send new packets almost as soon as they are available.

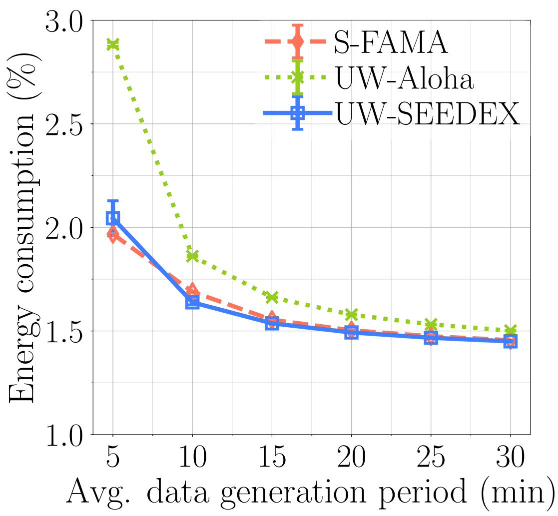
We did not expect UW-SEEDEX to have the lowest delay in all scenarios since



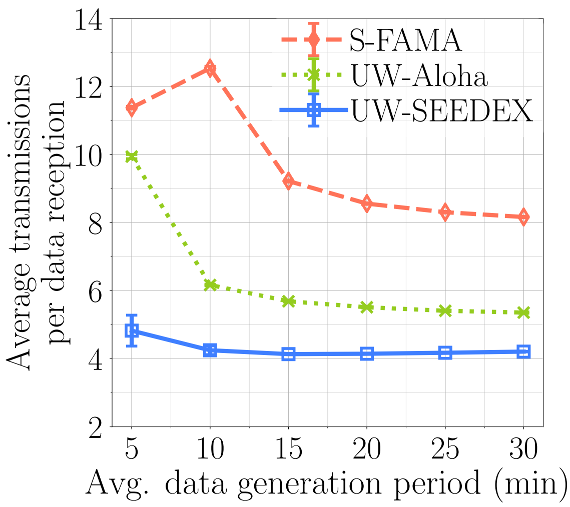
(a)



(b)



(c)



(d)

Figure 5.16: Performance comparison of the three MAC protocols as a function of the network traffic in networks with grid deployment ($4 \times 3 \times 3$ nodes).

it defines that nodes only have chances to transmit packets on some slots of the schedule(the ones where they are in PT state and their destinations are in L state). If we look at the results, we can observe that while the increase in the data generation period caused a great drop in the average end-to-end delay for both S-FAMA and UW-Aloha, it had little influence when using UW-SEEDEX.

5.4.2.2 Network density

Finally, we evaluated the MAC protocols using networks with different node density. To do so, we first set the average time to generate new packets as 10 minutes. Then, we

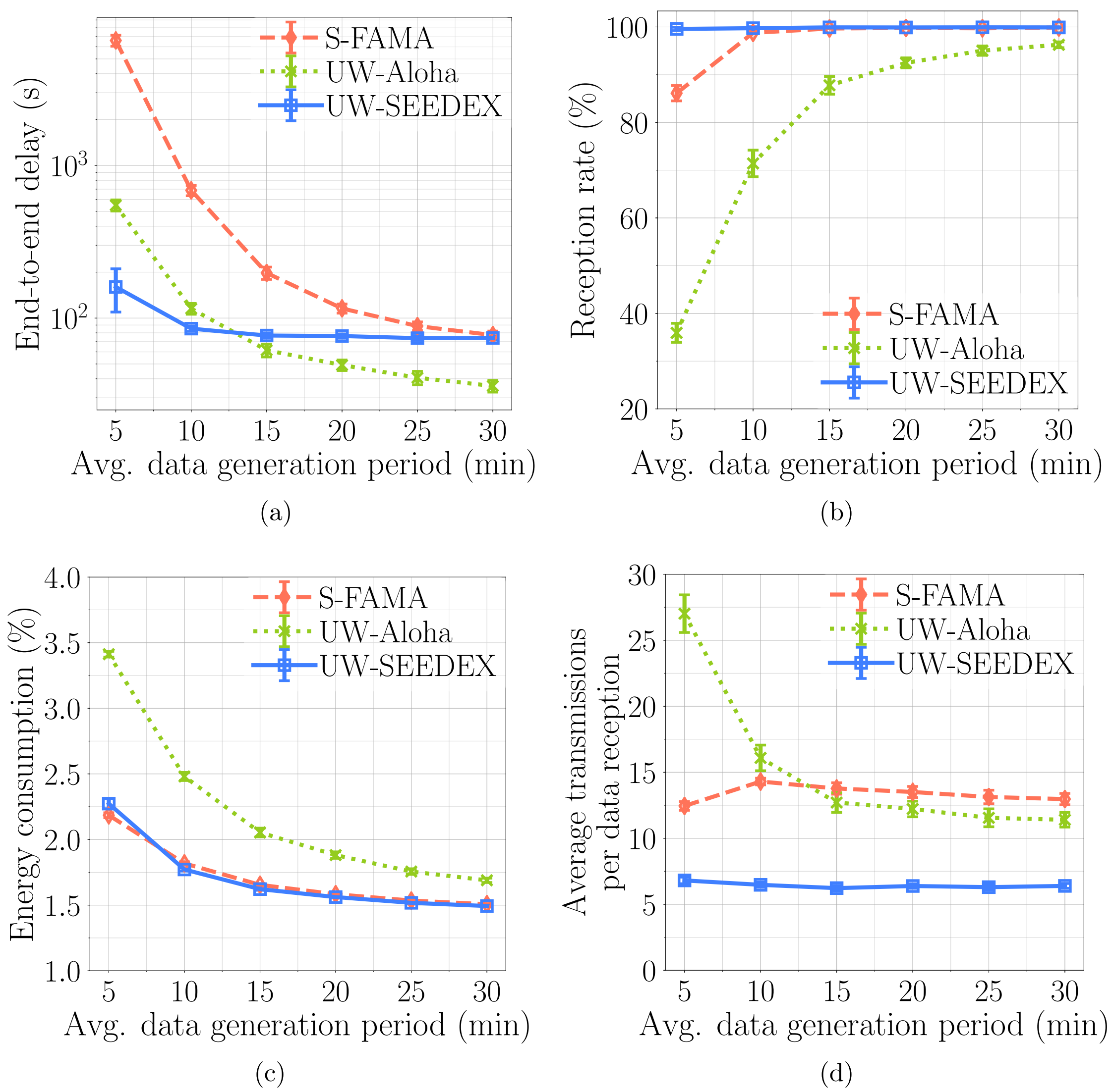


Figure 5.17: Performance comparison of the three MAC protocols as a function of the network traffic in networks with random deployment.

used the grid deployment method and, by using different grid spacings, we generated networks with different densities. We varied the grid spacing between 0.5 and 2.5 km, which produced network densities between 0.005 (for 2.5 km) and 0.667 (for 0.5 km) nodes per km^3. While the use of a 2.5 km spacing resulted in sparse networks, the use of 0.5 km spacing resulted in fully connected networks. Figure 5.18 shows the results we observed.

UW-SEEDEX had similar or better performance than the other protocols in most of the tested network densities for the different metrics. Starting with the end-to-end delay, we can observe that UW-Aloha achieved the lowest values in all scenarios, except the one with grid spacing equal to 2.5 km, where the protocol proposed here presented

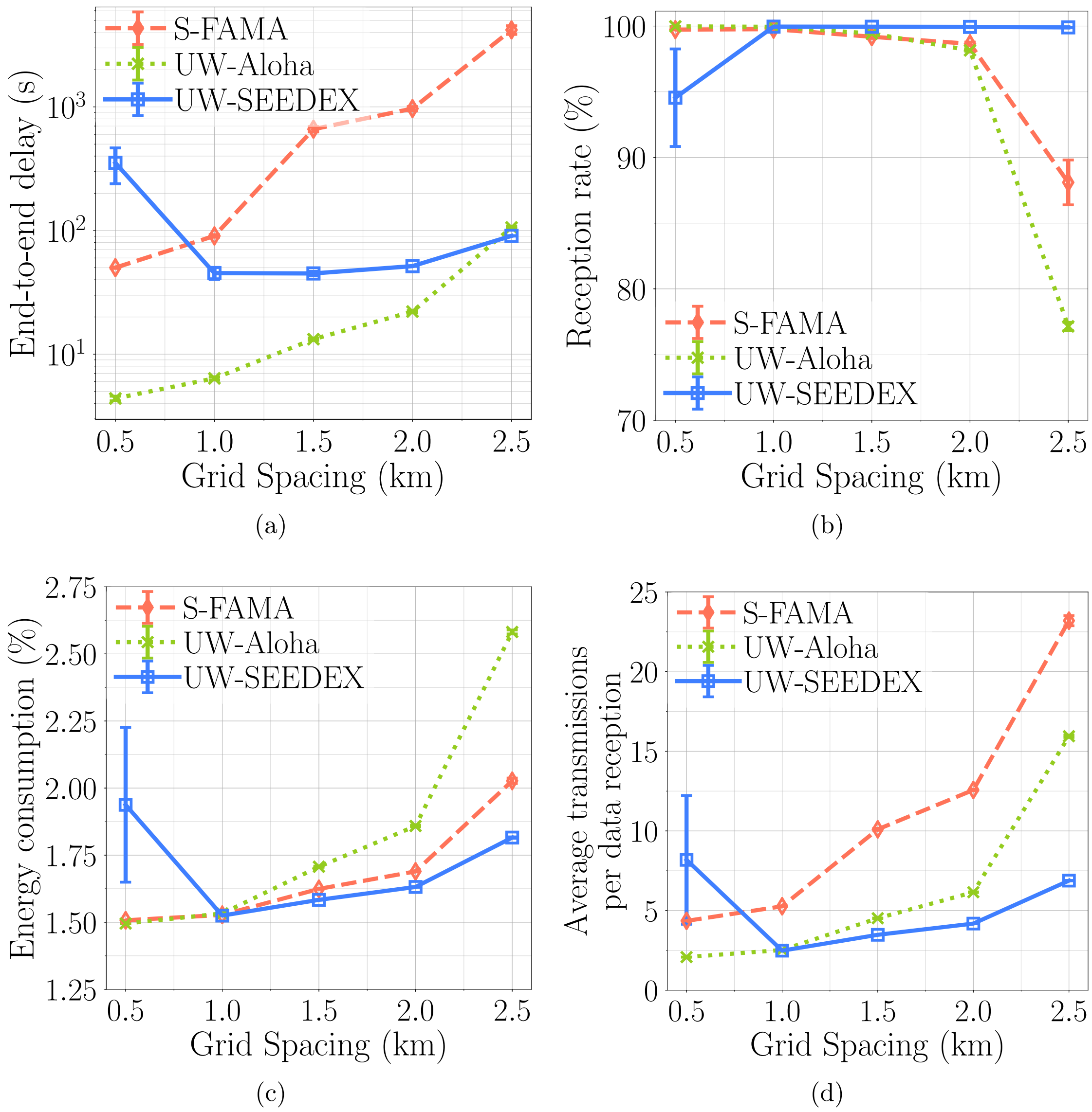


Figure 5.18: Performance comparison of the three MAC protocols as a function of the network density.

the lowest delay. Moving to the next metric, Figure 5.18b shows that UW-SEEDEX's reception rate was close to 100% in almost all scenarios, while those of UW-Aloha and S-FAMA started close to 100% and then dropped as the grid spacing increased. Figures 5.18c and 5.18d show that UW-SEEDEX outperforms the other protocols in networks that are not so dense. When the grid spacing was greater than 1 km, the protocol consumed less energy and lead to fewer transmissions per data reception than both UW-Aloha and S-FAMA. UW-SEEDEX consumed up to 10% less energy than S-FAMA and 30% less than UW-Aloha. The use of the protocol also achieved values of average transmissions per data reception that were up to 70% and 57% lower than those of S-FAMA and UW-Aloha, respectively.

All protocols had weaker performances in sparse networks than in dense ones, but UW-SEEDEX deterioration was lower than the others. Increasing the grid spacing resulted in the necessity of some messages to go through multiple hops before reaching their destinations, which in turn increased the network traffic. Thus, these results are consistent with those shown in Figures 5.16 and 5.17.

We can also note in Figure 5.18 that UW-SEEDEX performance had a wide variation in fully connected networks, and their averages were not so good as those of UW-Aloha and S-FAMA. This result seems to be related to the low transmission probabilities pt that nodes must have obtained due to the high network connectivity, as the transmission probability decreases with the number of neighbors in the PT state of the destination node.

Chapter 6

Conclusion and Future Work

This chapter first presents a summary of this dissertation in Section 6.1. Then, Section 6.2 describes future research directions for the proposed works.

6.1 Summary of this Dissertation

In this dissertation, we proposed solutions for two problems in underwater wireless sensor networks (UWSNs). The first solution addresses the data collection problem in underwater optical-acoustic sensor networks (UOASNs). The second solution is a medium access control (MAC) protocol for underwater acoustic sensor networks (UWASNs).

We first proposed the data collection solution. It is named CAPTAIN, and it can be seen as a three-phase algorithm. In its first phase, CAPTAIN clusters networks, classifying their nodes as either cluster heads or members according to (i) their remaining energy level and (ii) the proportion of their neighbors that are reached through optical communication. Its second phase builds a routing three from the sink node, connecting heads to other heads and members to other nodes in their clusters. The third and last phase is about maintaining the routes and relaying the collected data. When forwarding data, cluster members transmit using their optical systems. Meanwhile, heads aggregate data from their clusters and utilize acoustic communication to send the results.

We evaluated CAPTAIN using simulations. We compared it to the shortest path algorithm (SPA), and the results showed that our proposed solution was able to save more energy, principally in dense networks. CAPTAIN was able to save more energy in many scenarios, consuming up to approximately 70% less than SPA in dense networks. This result is very important for underwater sensor networks since their nodes' life are very restricted. We could also observe that CAPTAIN could take more advantages of clusters in the networks than the SPA, resulting in more optical transmissions and less acoustic transmissions. Also, the average latency was generally lower when using CAPTAIN than when using SPA in networks with clusters. CAPTAIN also achieved rates of data collection per hour close to the ideal ones in these networks, using, on average, fewer acoustic transmissions than SPA. Therefore, CAPTAIN is suitable for UOASNs and has better performance when nodes are deployed in clusters.

Next, we proposed the MAC protocol called UW-SEEDEX. The protocol employes random timeslot schedules composed of two possible states: "Listening" (L) and "Possible Transmitting" (PT). Nodes must only listen for other packets when in L states, while they can also transmit when in PT states. The key idea of UW-SEEDEX is to produce the timeslots from seeds and using pseudorandom number generators. By doing so, nodes can predict other schedules just by knowing their seeds, and so they can plan their transmissions to avoid collisions.

UW-SEEDEX was inspired by an existing MAC protocol for terrestrial networks named SEEDEX. However, to cope with the characteristics of the underwater acoustic channel, UW-SEEDEX (i) adopts a slightly different scheme for seed sharing, and (ii) considers propagation delay and acknowledgments to determine the timeslot length.

We also used simulations to evaluate UW-SEEDEX. First, we evaluated the protocol parameters. Then, we compared it to Slotted FAMA and UW-Aloha. Results showed that UW-SEEDEX could deliver more messages within the same time window while using fewer transmissions and consuming less energy than the other protocols.

6.2 Future Work

For CAPTAIN, our future research directions are the following. We intend to evaluate the use of other score functions in its phase of cluster formation. We also expect to investigate how much aspects, such as the number of clusters, the average number of members, and the average cluster diameter, affect CAPTAIN performance. Finally, we intend to evaluate cross-layer designs that could decrease the latency caused by the TDMA system we adopted at the MAC layer.

Moving to UW-SEEDEX, we intend to evaluate the insertion of a new state where nodes must sleep instead of transmitting or listening. This new state could save energy, which is a precious resource for UWNs, and so it is worth investigating how its addition would affect the protocol performance. We also intend to test the use of adaptive α and p, as they could be useful for nodes to adjust their transmissions according to variations in the network traffic conditions.

Bibliography

Akyildiz, I. F., Pompili, D., and Melodia, T. (2004). Challenges for efficient communication in underwater acoustic sensor networks. *SIGBED Rev.*, 1(2):3–8.

Akyildiz, I. F., Pompili, D., and Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257 – 279. ISSN 1570-8705.

Anguita, D., Brizzolara, D., Parodi, G., and Hu, Q. (2011). Optical wireless underwater communication for AUV: Preliminary simulation and experimental results. In *OCEANS, 2011 IEEE-Spain*, pages 1--5. IEEE.

Aval, Y. M., Han, Y., Tu, A., Basagni, S., Stojanovic, M., and Fei, Y. (2016). Testbed-based performance evaluation of handshake-free MAC protocols for underwater acoustic sensor networks. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–7. ISSN null.

Azad, S., Casari, P., Petrioli, C., Petroccia, R., and Zorzi, M. (2011). On the impact of the environment on MAC and routing in shallow water scenarios. In *OCEANS 2011 IEEE - Spain*, pages 1–8. ISSN null.

Brekhovskikh, L. and Lysanov, Y. (1982). *Fundamentals of Ocean Acoustics*. Springer.

Campagnaro, F., Favaro, F., Guerra, F., Calzado, V. S., Zorzi, M., and Casari, P. (2015). Simulation of multimodal optical and acoustic communications in underwater networks. In *OCEANS 2015-Genova*, pages 1--7. IEEE.

Caruso, A., Paparella, F., Vieira, L. F. M., Erol, M., and Gerla, M. (2008). The meandering current mobility model and its impact on underwater mobile sensor networks. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 221–225. ISSN 0743-166X.

Chen, K., Ma, M., Cheng, E., Yuan, F., and Su, W. (2014). A Survey on MAC Protocols for Underwater Wireless Sensor Networks. *IEEE Communications Surveys Tutorials*, 16:1433–1447. ISSN 2373-745X.

Coates, R. F. (1989). *Underwater acoustic systems*. Halsted Press.

Corinaldesi, C. (2015). New perspectives in benthic deep-sea microbial ecology. *Frontiers in Marine Science*, 2:17. ISSN 2296-7745.

Coutinho, R. W., Boukerche, A., Vieira, L. F., and Loureiro, A. A. (2016). On the design of green protocols for underwater sensor networks. *IEEE Communications Magazine*, 54(10):67--73.

Coutinho, R. W. L. (2017). *Topology control and opportunistic routing in underwater sensor networks*. PhD thesis, Universidade Federal de Minas Gerais.

Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., and Loureiro, A. A. F. (2014). GEDAR: geographic and opportunistic routing protocol with depth adjustment for mobile underwater sensor networks. In *Communications (ICC), 2014 IEEE International Conference on*, pages 251--256. IEEE.

Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., and Loureiro, A. A. F. (2017). EnOR: Energy balancing routing protocol for underwater sensor networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. ISSN 1938-1883.

Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., and Loureiro, A. A. F. (2018). Underwater wireless sensor networks: A new challenge for topology control–based systems. *ACM Comput. Surv.*, 51(1). ISSN 0360-0300.

Coutinho, R. W. L., Vieira, L. F. M., and Loureiro, A. A. F. (2013). DCR: Depth-controlled routing protocol for underwater sensor networks. In *2013 IEEE Symposium on Computers and Communications (ISCC)*, pages 000453--000458. IEEE.

Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2019a). 3DVS: Node scheduling in underwater sensor networks using 3D voronoi diagrams. *Computer Networks*, 159:73 – 83. ISSN 1389-1286.

Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2019b). Topology control in underwater sensor networks using cellular automata. *Journal of Cellular Automata*, 14.

Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2020a). CAPTAIN: A data collection algorithm for underwater optical-acoustic sensor networks. *Computer Networks*, 171:107145. ISSN 1389-1286.

Câmara Júnior, E. P. M., Vieira, L. F. M., and Vieira, M. A. M. (2020b). UW-SEEDEX: a pseudorandom-based MAC protocol for underwater acoustic networks. *IEEE Transactions on Mobile Computing.*

da Silva Santos, E. R., Câmara Júnior, E. P. M., Vieira, M. A. M., and Vieira, L. F. M. (2019). Aplicações de monitoramento de tráfego utilizando redes programáveis ebpf. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 417--430. SBC.

De, S., Mandal, P., and Chakraborty, S. S. (2011). On the characterization of Aloha in underwater wireless networks. *Mathematical and Computer Modelling*, 53(11):2093 – 2107. ISSN 0895-7177.

Demirkol, I., Ersoy, C., Alagoz, F., et al. (2006). MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115--121.

Erol, M., Vieira, L. F., Caruso, A., Paparella, F., Gerla, M., and Oktug, S. (2008). Multi stage underwater sensor localization using mobile beacons. In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pages 710--714. IEEE.

Erol, M., Vieira, L. F., and Gerla, M. (2007). Localization with Dive'N'Rise (DNR) beacons for underwater acoustic sensor networks. In *Proceedings of the second workshop on Underwater networks*, pages 97--100. ACM.

EvoLogics (2014). Evologics S2CR 18/34 product information. Accessed: Nov. 2017 [Online]. Available: `https://goo.gl/b36NVB`.

EvoLogics (2020). USBL Positioning and Communication Systems. Accessed: March 2020 [Online]. Available: `https://evologics.de/web/content/15977`.

Faheem, M., Tuna, G., and Gungor, V. C. (2018). QERP: Quality-of-service (qos) aware evolutionary routing protocol for underwater wireless sensor networks. *IEEE Systems Journal*, 12(3):2066–2073. ISSN .

Farr, N., Bowen, A., Ware, J., Pontbriand, C., and Tivey, M. (2010). An integrated, underwater optical /acoustic communications system. In *OCEANS'10 IEEE SYDNEY*, pages 1–6. ISSN .

Farr, N., Chave, A., Freitag, L., Preisig, J., White, S., Yoerger, D., and Titterton, P. (2005). Optical modem technology for seafloor observatories. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 928--934. IEEE.

Farr, N., Chave, A. D., Freitag, L., Preisig, J., White, S. N., Yoerger, D., and Sonnichsen, F. (2006). Optical modem technology for seafloor observatories. In *OCEANS 2006*, pages 1–6. ISSN .

Freitag, L., Grund, M., Singh, S., Partan, J., Koski, P., and Ball, K. (2005). The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 1086–1092 Vol. 2. ISSN 0197-7385.

Fullmer, C. L. and Garcia-Luna-Aceves, J. J. (1995). Floor Acquisition Multiple Access (FAMA) for Packet-radio Networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '95, pages 262--273, New York, NY, USA. ACM.

Goyal, N., Dave, M., and Verma, A. K. (2017). Data aggregation in underwater wireless sensor network: Recent approaches and issues. *Journal of King Saud University-Computer and Information Sciences*.

Guerra, F., Casari, P., and Zorzi, M. (2009). World Ocean Simulation System (WOSS): A Simulation Tool for Underwater Networks with Realistic Propagation Modeling. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, WUWNet '09, pages 4:1--4:8, New York, NY, USA. ACM.

Gussen, C., Diniz, P., Campos, M., Martins, W., Costa, F., and Gois, J. (2016). A survey of underwater wireless communication technologies. *Journal of Communication and Information Systems*, 31(1).

Han, S., Noh, Y., Lee, U., and Gerla, M. (2019). Optical-acoustic hybrid network toward real-time video streaming for mobile underwater sensors. *Ad Hoc Networks*, 83:1--7.

Han, S., Noh, Y., Liang, R., Chen, R., Cheng, Y.-J., and Gerla, M. (2014). Evaluation of underwater optical-acoustic hybrid network. *China Communications*, 11(5):49--59.

Heidemann, J., Stojanovic, M., and Zorzi, M. (2012). Underwater sensor networks: applications, advances and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):158--175.

Heidemann, J., Ye, W., Wills, J., Syed, A., and Li, Y. (2006). Research challenges and applications for underwater sensor networking. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 228--235. IEEE.

Hu, T. and Fei, Y. (2012). MURAO: A multi-level routing protocol for acoustic-optical hybrid underwater wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, pages 218--226. IEEE.

Jiang, S. (2018). State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Communications Surveys Tutorials*, 20(1):96–131. ISSN 2373-745X.

Jiejun Kong, Jun-hong Cui, Dapeng Wu, and Gerla, M. (2005). Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications. In *MILCOM 2005*, volume 3, pages 1535–1541. ISSN 2155-7586.

Johnson, L. J., Green, R. J., and Leeson, M. S. (2014). Hybrid underwater optical/acoustic link design. In *2014 16th International Conference on Transparent Optical Networks (ICTON)*, pages 1--4. IEEE.

Karn, P. (1990). MACA - A new channel access method for packet radio. In *ARRL/CRRL Amateur radio 9th computer networking conference*, volume 140, pages 134--140. London, Canada.

Kaushal, H. and Kaddoum, G. (2016). Underwater optical wireless communication. *IEEE Access*, 4:1518–1547.

Khan, A., Ali, I., Ghani, A., Khan, N., Alsaqer, M., Rahman, A., and Mahmood, H. (2018). Routing protocols for underwater wireless sensor networks: Taxonomy, research challenges, routing strategies and future directions. *Sensors*, 18(5):1619.

Kosowsky, R. P., Jacobs, I. M., and Gilhousen, K. S. (1988). ARNS: a new link layer protocol. In *MILCOM 88, 21st Century Military Communications - What's Possible?'. Conference record. Military Communications Conference*, pages 515–519 vol.2. ISSN null.

Kredo II, K., Djukic, P., and Mohapatra, P. (2009). STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol. In *IEEE INFOCOM 2009*, pages 2961–2965. ISSN 0743-166X.

Kumar, P., Kumar, P., Priyadarshini, P., and Srija (2012). Underwater acoustic sensor network for early warning generation. In *2012 Oceans*, pages 1–6. ISSN 0197-7385.

Lanzagorta, M. (2012). Underwater communications. *Synthesis lectures on communications*, 5(2):1--129.

Lee, U., Wang, P., Noh, Y., Vieira, L. F., Gerla, M., and Cui, J.-H. (2010). Pressure routing for underwater sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1--9. IEEE.

Liao, W. and Huang, C. (2012). SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks. *IEEE Sensors Journal*, 12(6):1686–1694. ISSN 2379-9153.

Lima, F. H. M. B., Vieira, L. F. M., Vieira, M. A. M., Vieira, A. B., and Nacif, J. A. M. (2019). Water ping: ICMP for the internet of underwater things. *Computer Networks*, 152:54 – 63. ISSN 1389-1286.

Liu, J., Zhou, Z., Peng, Z., Cui, J., Zuba, M., and Fiondella, L. (2013). Mobi-Sync: Efficient Time Synchronization for Mobile Underwater Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(2):406–416. ISSN 2161-9883.

Lloret, J., Sendra, S., Ardid, M., and Rodrigues, J. J. (2012). Underwater wireless sensor communications in the 2.4 ghz ism frequency band. *Sensors*, 12(4):4237--4264.

Lu, Q., Liu, F., Zhang, Y., and Jiang, S. (2017). Routing protocols for underwater acoustic sensor networks: A survey from an application perspective. *Advances inUnderwater Acoustics*, page 23.

Luo, H., Guo, Z., Dong, W., Hong, F., and Zhao, Y. (2010). LDB: Localization with directional beacons for sparse 3D underwater acoustic sensor networks. *Journal of networks*, 5(1):28.

Ma, Y., Guo, Z., Feng, Y., Jiang, M., and Feng, G. (2009). C-MAC: A TDMA-Based MAC Protocol for Underwater Acoustic Sensor Networks. In *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, volume 1, pages 728–731. ISSN null.

Manjula, R. and Manvi, S. S. (2011). Issues in underwater acoustic sensor networks. *International Journal of Computer and Electrical Engineering*, 3(1):101.

Manjula, R. B. and Manvi, S. S. (2012). Cluster based data aggregation in underwater acoustic sensor networks. In *India Conference (INDICON), 2012 Annual IEEE*, pages 104--109. IEEE.

Mao, J., Chen, S., Liu, Y., Yu, J., and Xu, Y. (2015). LT-MAC: A location-based TDMA MAC protocol for small-scale underwater sensor networks. In *2015 IEEE*

International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), pages 1275–1280. ISSN null.

Melodia, T., Kulhandjian, H., Kuo, L.-C., and Demirors, E. (2013). Advances in underwater acoustic networking. *Mobile ad hoc networking: Cutting edge directions*, pages 804--852.

Mhatre, V. and Rosenberg, C. (2004). Homogeneous vs heterogeneous clustered sensor networks: a comparative study. In *Communications, 2004 IEEE International Conference on*, volume 6, pages 3646--3651. IEEE.

Molins, M. and Stojanovic, M. (2006). Slotted FAMA: a MAC protocol for underwater acoustic networks. In *OCEANS 2006 - Asia Pacific*, pages 1–7. ISSN .

Moriconi, C., Cupertino, G., Betti, S., and Tabacchiera, M. (2015). Hybrid acoustic/optic communications in underwater swarms. In *OCEANS 2015 - Genova*, pages 1–9. ISSN .

NOAA. (2018). How much of the ocean have we explored? National Ocean Service website. Accessed: March 2020 [Online]. Available: `https://oceanservice.noaa.gov/facts/exploration.html`.

Noh, Y., Lee, U., Lee, S., Wang, P., Vieira, L. F. M., Cui, J.-H., Gerla, M., and Kim, K. (2016). Hydrocast: Pressure routing for underwater sensor networks. *IEEE Trans. Vehicular Technology*, 65(1):333--347.

Othman, A.-K., Adams, A., and Tsimenidis, C. C. (2006). Node discovery protocol and localization for distributed underwater acoustic networks. In *Telecommunications, 2006. AICT-ICIW'06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, pages 93--93. IEEE.

Partan, J., Kurose, J., and Levine, B. N. (2007). A survey of practical issues in underwater networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4):23–33. ISSN 1559-1662.

Peng, Z., Zhou, Z., Cui, J., and Shi, Z. J. (2009). Aqua-Net: An underwater sensor network architecture: Design, implementation, and initial testing. In *OCEANS 2009*, pages 1–8. ISSN 0197-7385.

Pompili, D. and Akyildiz, I. F. (2009). Overview of networking protocols for underwater wireless communications. *IEEE Communications Magazine*, 47(1):97–102. ISSN 1558-1896.

Preisig, J. (2007). Acoustic propagation considerations for underwater acoustic communications network development. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4):2–10. ISSN 1559-1662.

Qiu, S., Guo, W., Li, B., Wu, Y., Chu, X., Wang, S., and Dong, Y. Y. (2015). Long range and long duration underwater localization using molecular messaging. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 1(4):363–370.

Rappaport, T. S. et al. (1996). *Wireless communications: principles and practice*, volume 2. Prentice Hall PTR New Jersey.

Ribeiro, F. J. L., Pedroza, A. d. C. P., and Costa, L. H. M. K. (2015). Underwater monitoring system for oil exploration using acoustic sensor networks. *Telecommunication Systems*, 58(1):91--106.

Riley, G. F. and Henderson, T. R. (2010). The ns-3 network simulator. In Wehrle, K., Güneş, M., and Gross, J., editors, *Modeling and Tools for Network Simulation*, pages 15--34. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rozovsky, R. and Kumar, P. R. (2001). SEEDEX: A MAC Protocol for Ad Hoc Networks. In *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '01, pages 67--75, New York, NY, USA. ACM.

Shah, G. A. (2009). A survey on medium access control in underwater acoustic sensor networks. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 1178--1183. IEEE.

Shakir, M., Khan, M. A., Malik, S. A., and ul Haq, I. (2012). Design of underwater sensor networks for water quality monitoring. In *World Applied Sciences Journal,vol.17,no.11*, page 1441–1444.

Sonardyne (2016). Datasheet BlueComm 200 - optical communications system. Accessed: Nov. 2017 [Online]. Available: `https://goo.gl/YaXpid`.

Stojanovic, M. (2007). On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):34--43.

Stojanovic, M. and Preisig, J. (2009). Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE communications magazine*, 47(1):84--89.

Syed, A. A. and Heidemann, J. (2006). Time synchronization for high latency acoustic networks. In *Proceedings IEEE INFOCOM 2006*, pages 1–12. ISSN 0743-166X.

Syed, A. A., Ye, W., and Heidemann, J. (2008). T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 231–235. ISSN 0743-166X.

Tan, H. and Seah, W. K. G. (2007). Distributed CDMA-based MAC Protocol for Underwater Sensor Networks. In *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, pages 26–36. ISSN 0742-1303.

Tavares, R. C., Carvalho, M., Câmara Júnior, E. P. M., de Britto e Silva, E., Vieira, M. A. M., Vieira, L. F. M., and Krishnamachari, B. (2019). FWB: Funneling Wider Bandwidth algorithm for high performance data collection in Wireless Sensor Networks. *Computer Communications*, 148:136 – 151. ISSN 0140-3664.

Tennenbaum, A., Dyakiw, M., Cui, J.-H., and Peng, Z. (2014). Application of low cost optical communication systems to underwater acoustic networks. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 755--758. IEEE.

Vasilescu, I., Detweiler, C., and Rus, D. (2007). Aquanodes: an underwater sensor network. In *Proceedings of the second workshop on Underwater networks*, pages 85--88. ACM.

Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., and Corke, P. (2005). Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, page 154–165, New York, NY, USA. Association for Computing Machinery.

Vieira, L., Loureiro, A., Fernandes, A., and Campos, M. (2010). Redes de sensores aquáticas. *XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Gramado, RS, Brasil*, 24.

Vieira, L. F. M., Almiron, M. G., and Loureiro, A. A. (2011). 3D MANETs: Link probability, node degree, network coverage and applications. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 2042--2047. IEEE.

Vieira, L. F. M., Kong, J., Lee, U., and Gerla, M. (2006). Analysis of aloha protocols for underwater acoustic sensor networks. *Extended abstract from WUWNet*, 6.

Vieira, L. F. M., Lee, U., Noh, Y., and Gerla, M. (2009). LPS: Laser positioning system for underwater networks. In *ACM International Workshop on Underwater Networks, WUWNet'09*.

Vieira, L. F. M., Vieira, M. A. M., Nacif, J. A., and Vieira, A. B. (Jan. 2018). Autonomous wireless lake monitoring. In *Computing in Science & Engineering (Print)*.

Vieira, M. A. M., Castanho, M. S., Pacífico, R. D. G., Santos, E. R. S., Júnior, E. P. M. C., and Vieira, L. F. M. (2020). Fast Packet Processing with EBPF and XDP: Concepts, Code, Challenges, and Applications. *ACM Comput. Surv.*, 53(1). ISSN 0360-0300.

Wang, J., Shi, W., Xu, L., Zhou, L., Niu, Q., et al. (2017). Design of optical-acoustic hybrid underwater wireless sensor network. *Journal of Network and Computer Applications*.

Wei, X., Zhao, L., Li, X., and Zou, C. (2008). A Distributed Power Control Based MAC Protocol for Underwater Acoustic Sensor Networks. In *2008 4th IEEE International Conference on Circuits and Systems for Communications*, pages 688–692. ISSN null.

Xiao, Y. (2010). *Underwater acoustic sensor networks*. CRC Press.

Xie, P., Cui, J.-H., and Lao, L. (2006). VBF: vector-based forwarding protocol for underwater sensor networks. In *International conference on research in networking*, pages 1216--1221. Springer.

Yan, H., Shi, Z. J., and Cui, J.-H. (2008). DBR: depth-based routing for underwater sensor networks. In *International conference on research in networking*, pages 72--86. Springer.

Zeng, Z., Fu, S., Zhang, H., Dong, Y., and Cheng, J. (2017). A survey of underwater optical wireless communications. *IEEE Communications Surveys Tutorials*, 19(1):204–238. ISSN .

Zhou, Z., Cui, J.-H., and Zhou, S. (2007). Localization for large-scale underwater sensor networks. *Networking 2007. Ad hoc and sensor networks, wireless networks, next generation internet*, pages 108--119.