Università degli Studi di Padova

Department of Information Engineering

Ph.D. Course in Information Engineering
Science and Information Technology Curriculum

XXXIV Series

Underwater Acoustic Networks: Protocols Design and Security Issues

Ph.D. Candidate
Alberto Signori

Supervisor
Prof. Michele Zorzi

Coordinator
Prof. Andrea Neviani

Academic Year 2020/2021

Abstract

Although underwater networks have been developed for considerable time now, the harshness of the acoustic channel still poses a lot of challenges for wireless communications. In addition, the development of low cost modems paves the way for new applications in different scenarios, such as future smart ports. Nevertheless, for the time being, the use of underwater networks is still mainly related to military and public safety applications, whose relevance arises further challenges related also to network security. Indeed, a network failure caused by an attack could lead to serious consequences. However, security problems have not been deeply investigated so far, and the implementation of terrestrial network countermeasures in the underwater field is not always feasible. In this thesis we focus on both open challenges previously described: the design of efficient communication protocols, and the analysis of attacks and countermeasures in underwater networks. First, we consider a data collection service in a smart port scenario, designing a Medium Access Control (MAC) protocol for this service and analyzing the End-To-End (E2E) transmission, from sensor nodes to the shore. We evaluate the performance through both simulations and field tests. Afterward, we analyze security issues in underwater networks. We start by analyzing attackers with limited knowledge of the protocol stack, studying the effect of jamming and replay attacks. We then move to the design of a countermeasure based on a trust metric to be employed against more sophisticated attacks in which the malicious node can exploit the protocol's behavior and vulnerabilities to carry out the attacks. In both cases we specifically focus on the peculiarities of the acoustic channel that differentiate attacks and countermeasures in underwater networks from their terrestrial counterpart.

Sommario

Nonostante le reti sottomarine siano ormai studiate da tempo, esistono ancora molte difficoltà da affrontare, causate dell'avversità del canale acustico, per poter stabilire una comunicazione wireless affidabile. Inoltre, lo sviluppo di nuovi modem a basso costo apre la strada a nuove applicazioni per le reti sottomarine, come, ad esempio, applicazioni per i futuri porti *intelligenti*. Ad oggi, comunque, i principali ambiti di utilizzo delle reti sottomarine rimangono quello militare e di pubblica sicurezza, la cui rilevanza pone inoltre ulteriori sfide legate anche alla sicurezza delle reti. Infatti, un'interruzione dei servizi forniti dalle reti causata da un attacco potrebbe portare a serie conseguenze. Tuttavia fino ad ora, gli aspetti di sicurezza delle reti sottomarine non sono stati adeguatamente studiati e l'utilizzo di contromisure progettate per reti terrestri non sempre risulta possibile. In questa tesi ci occuperemo sia della progettazione di protocolli di comunicazione efficaci, che dell'analisi di attacchi e contromisure nelle reti sottomarine. Prima di tutto, progetteremo un protocollo MAC per la raccolta di dati da sensori sottomarini in ambiente portuale e analizzeremo le prestazioni complessive della rete, dalla trasmissione dei dati dai sensori fino alla ricezione sulla costa. L'analisi delle prestazioni verrà effettuata sia attraverso simulazioni che attraverso test sperimentali. Successivamente, ci dedicheremo all'analisi della sicurezza nelle reti sottomarine. Cominceremo analizzando degli scenari in cui gli attaccanti dispongono di una conoscenza limitata dei protocolli di rete, studiando gli effetti di attacchi di tipo *jamming* e *replay*. Progetteremo, poi, contromisure basate sulla reputazione di un nodo, a protezione di attacchi più sofisticati in cui il nodo attaccante conosce il funzionamento e le vulnerabilità dei protocolli e sfrutta queste conoscenze per attaccare la rete. In entrambi i casi, ci concentreremo sulle peculiarità delle comunicazioni acustiche che rendono gli attacchi e le contromisure differenti rispetto a quelli in ambito terrestre.

Contents

List of Figures

List of Tables

List of Acronyms

8PSK Eight Phase-Shift Keying

AC APOLL Controller

AMQP Advanced Message Queuing Protocol

AODV Ad-hoc On-demand Distance Vector

ARQ Automatic Repeat Request

ASV Autonomous Surface Vehicle

AUV Autonomous Underwater Vehicle

AWGN Additive White Gaussian Noise

BER Bit Error Rate

BNE Bayesian Nash Equilibrium

BPSK Binary Phase-Shift Keying

CRC Cyclic Redundancy Check

CSMA Carrier-Sense Multiple Access

CSS Chirp Spread Spectrum

CTS Clear-To-Send

DACAP Distance-Aware Collision Avoidance Protocol

DNS Domain Name System

DoS Denial of Service

DQPSK Differential Quadrature Phase Shift Keying

DSR Dynamic Source Routing

DTN Delay Tolerant Network

E2E End-To-End

ECDF Experimental Cumulative Density Function

ED End Device

ELF Extremly Low Frequency

FEC Forward Error Correction

GBN Go-Back-N

GUWMANET Gossiping in Underwater Acoustic Mobile Ad-hoc Networks

GW Gateway

HF High Frequency

HMM Hidden Markov Model

JFI Jain's Fairness index

LED Light Emitting Diode

LF Low Frequency

LoS Line of Sight

LPWAN Low Power Wide Area Network

LUT Look-Up-Table

MAC Medium Access Control

MANET Mobile Ad Hoc Networks

MC Markov Chain

MCS Modulation and Coding Scheme

MF Medium Frequency

MI Magneto-Inductive

MU Mobile Unit

NE Nash Equilibrium

NS Network Server

ns2 Network Simulator 2

PDD Packet Delivery Delay

PDR Packet Delivery Ratio

PER Packet Error Rate

pmf probability mass function

PoC Proof of Concept

PPP Poisson Point Process

QPSK Quadrature Phase-Shift Keying

RF Radio Frequency

ROV Remotely Operated Vehicle

RS Reed-Solomon

RTS Request-To-Send

SBC Single-Board Computer

SF Spreading Factor

SINR Signal to Interference plus Noise Ratio

SNR Signal to Noise Ratio

TAODV Trusted AODV

TCM Trellis Coded Modulation

TDMA Time Division Multiple Access

UAN Underwater Acoustic Network

USN Underwater Sensor Network

UV Unmanned Vehicle

VLF Very Low Frequency

WSN Wireless Sensor Network

Chapter 1

Introduction

Water covers 71% of the Earth surface and underwater communications can be used to exploit this huge amount of space with new applications and services. Protocols and techniques used for above water networks cannot be directly applied and employed in the underwater domain. Indeed, the Radio Frequency (RF) waves only propagate for few meters and are only employed when a broadband communication link needs to be established between two nodes separated by few meters or even centimeters [3, 4]. To establish a long range underwater wireless communication, instead, acoustic waves are mostly used. Acoustic links can reach up to tens of kilometers but with a limited bitrate up to tens of kbit/s [5, 6]. A third solution that can be used to communicate underwater are optical waves. In this case the transmission range increases up to few hundreds of meters with a relatively high data rate (in the order of Mbit/s) [7, 8].

The applications for which underwater networks can be exploited, therefore, depend on the technology employed for the communication, and the design of new applications needs to face and consider all the challenges originating from these technologies, such as low bitrate or short communication range. Underwater networks are mostly employed for military applications, in which acoustic networks can be deployed for surveillance and intruder detection along the coast or for mine counter-measure [9]. These networks can also be used for public safety such as tsunami prevention, for industrial applications such as in smart port scenario or in the oil and gas industry, and for environmental and scientific monitoring such as water pollution analysis [10, 11]. For example, with the development and improvement of technologies for the underwater communications and the advance in the development of underwater vehicles, both unmanned and remotely operated, the working activities inside a harbor can be enhanced and improved using these new technologies towards a safer and more efficient port. Among others, one of the activities required in a smart port is the continuous monitoring of water in the port area, to detect possible anomalous values in water pollution and thus to promptly and properly react [12]. The data can be collected by means of an AUV which patrols the port area while asking for new data to the underwater sensor nodes.

In all these applications, a node of the underwater network can be equipped with

either a modem that exploits a single communication technology, or multiple modems with different technologies to be combined creating a multimodal network. In this last case the goal is to exploit the peculiarities and the advantages of each technology to overcome the limits of the others [13]. Multimodal networks can also include those networks which employ the same technology, e.g., the acoustic one, but with different non-overlapping bandwidths obtaining different bitrates and communication ranges [11].

Communicating underwater is still a relevant challenge due to the characteristics of the underwater acoustic channel such as multipath, high delay spread and the presence of the ambient noise (caused by waves, wind and also vessel propellers) [5, 14, 15], which may result in long packet error bursts. For these reasons, so far most of the effort by the research community has been focused on the development of both communication protocols, e.g., MAC [16, 17] or routing [18, 19, 20], to mitigate the effect of the underwater channel, and self-adapting physical layer, e.g., changing the Modulation and Coding Scheme (MCS) based on the link quality [21, 22], to improve the communication performance.

Despite all these challenges in the underwater communications, another important element that needs to be carefully taken into account is security in this type of networks. Indeed, even simple attacks such as Denial of Service (DoS) or sinkhole attacks can lead to disastrous consequences when performed in mission critical scenarios [23, 24], such as military networks for surveillance or public safety networks employed for tsunami prevention. However, albeit the relevant scenarios in which underwater networks can be deployed, network security has not been widely investigated so far and is still an open issue in the underwater field.

The rest of this chapter is organized as follows. Section 1.1 provides an overview of the available technologies for underwater communications, describing the operational advantages and limits of each of them. It also provides a detailed survey on the available modems with their performance in terms of communication range and bitrate, to better understand the applications in which these technologies can be exploited and how they can be combined together. Section 1.2 then introduces the problem of security in underwater networks, providing the state of the art in this field and a comparison with security in the terrestrial domain, analyzing the differences between the two and the limits of applying the terrestrial countermeasures and solutions to underwater networks. Finally, Section 1.3 presents the structure of the thesis.

1.1 Wireless Communication Techniques

Nowadays, the technologies used for underwater wireless communications are acoustics, RF, Magneto-Inductive (MI) and optical[25]. In this section, we analyze these technologies by presenting pros and cons of each system, some details on their expected performance and the lessons learnt from the actual performance we experienced during sea trials.

As a summary, Figure 1.1 shows the comparison between some of the modems and

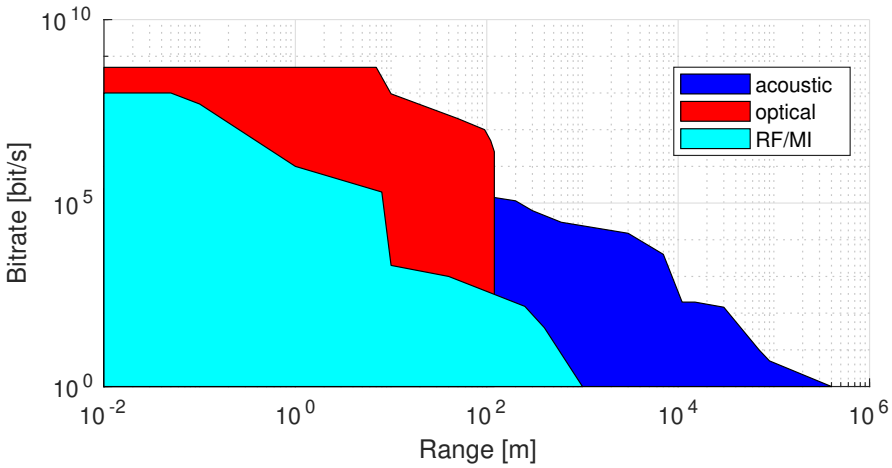


Figure 1.1: Nominal bitrate vs. range for the best among the technologies presented in Section 1.1.

the technologies presented in this section and summarized in Tables 1.1, 1.2 and 1.3.

1.1.1 Acoustic Communications

Developed in the late 1950's [26], acoustic communication is certainly the most mature underwater telecommunication technology available so far. It provides long transmission ranges, up to tens of kilometers, depending on the carrier frequency and the environmental conditions [5]. The modem bandwidth and its consequent communication rate depend on the carrier frequency, the characteristics of the modem transducers, as well as on external conditions, such as the noise caused by ships, wind and marine life, the multipath and the Doppler effect caused by the movements of the submerged nodes. For these reasons, the communication rate of an acoustic link ranges from few tens of bits per second for long range communications, up to few tens of kilobits per second for short range links. The main disadvantages of acoustic communications are the long propagation delay caused by the low speed of sound (1500 m/s, on average), the time variation of noise and of the channel impulse response, the presence of asymmetric acoustic links, and the poor performance in shallow water (i.e., when the water column is less than 100 m) due to signal reflections. In a mobile network deployed in shallow water, the multipath caused by signal reflections often results in link disruption, where, for instance, the communication between two nodes deployed at a depth of 1 m is established in the range of 0 to 110 m, lost in the range of 110 to 220 m, and established again in the range of 220 to 290 m (Figure 1.2a). The link, instead, is definitely more stable in a deep water scenario, where the communication can be established up to the maximum range of the modem, without link disruption (Figure 1.2b). Depending on the expected conditions and the user needs, there is a wide set of acoustic modems in the market, that can be employed in a variety of specific scenarios.

For example, to achieve a communication range of more than 4 km, a modem with a

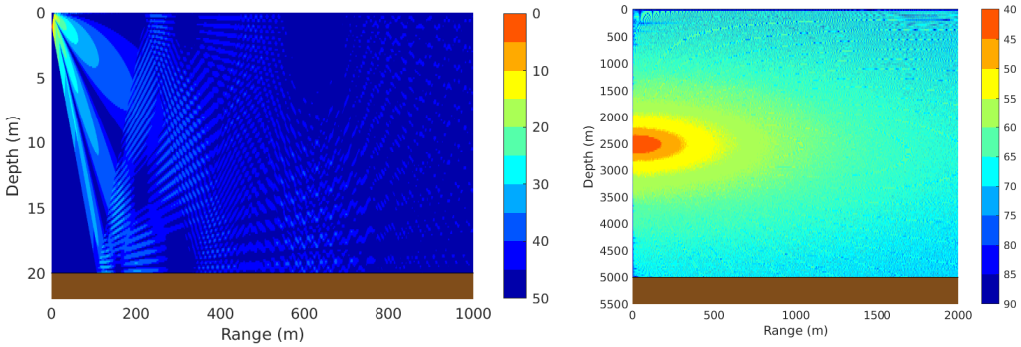


Figure 1.2: Example of multipath effect in shallow (a) and deep water (b) obtained through the Bellhop ray tracer [1].

carrier frequency below 12 kHz should be used, such as the Benthos ATM 960 in the Low Frequency (LF) band [27], the EvoLogics S2C 7/17 [6], the LinkQuest UWM3000 [28], the AQUATEC AQUAmodem1000 [29], the Develogic HAM.NODE [30], the Sercel MATS3G 12 kHz [31] modem, the WHOI MicroModem [32], or the Kongsberg cNODE LF [33]. The transducer of most of these devices can be customized to the geometry of the channel, and also the modem bitrate can be adapted accordingly. For this reason, all LF modems can achieve a communication rate up to few kilobits per second in a vertical link in deep water, where the multipath is negligible, while in a horizontal link in very shallow water they can reach a maximum rate of few hundreds of bits per second. Among the others, WHOI demonstrated that, in good channel conditions, it was possible to achieve a communication link of 70 to 400 km at 1 to 10 bit/s in the Arctic [34] by employing a carrier frequency of 900 Hz.

For communication ranges from 1 to 3 km, instead, a Medium Frequency (MF) modem is most suitable, as it can provide a higher bitrate. The carrier frequency of an MF modem is selected between 20 and 30 kHz, depending on the manufacturer. All the aforementioned manufacturers that produce LF devices also develop MF acoustic modems. In addition to them, other companies also supply commercial off-the-shelf products in this range, such as the Popoto Modem [35], the Applicon Seamodem [36], the Sonardyne 6G [37], the DSPComm Aquacomm Gen2 [38], the SubNero [39] and the Blueprint Subsea [40] acoustic modem. In low noise scenarios, a vertical communication link with an MF modem can reach a bitrate up to tens of kbit/s [41], depending on the modem manufacturer, while in horizontal communications in very shallow water they can provide a communication link with a bitrate spanning from 500 bit/s up to a few kbit/s.

Due to the high cost of good quality LF and MF transducers with a wide bandwidth (the cost of the single transducer can easily exceed 2 kEUR), research and industrial prototypes of high performance LF and MF acoustic modems are mostly developed by navy-related research institutes and companies, such as TNO [42], FOI [43], FFI [44],

Wärtsilä ELAC [45], and L3HARRIS [46], that are specifically interested in very long range communications for surveillance applications [47]. Also the JANUS NATO standard focuses on LF and MF frequency bands [48]. In this frequency domain, some universities and civil research institutes developed some low-cost low-power products for medium and short range (few hundreds of meters) low rate (few hundreds of bits per second) modems for internet of things (IoT) applications [49, 50, 51], by employing low cost narrow band transducers. Some commercial low cost MF acoustic modems (with a price of less than 2 kEUR) are also available off-the-shelf. For instance, the modem recently launched by DSPComm [38] has a maximum transmission rate of 100 bit/s, and a nominal range of 500 m. With the same range, the Micron Data Modem developed by Tritech [52] is a low power compact modem with a maximum data rate of 40 bit/s. Similar performance can be obtained with the Desert Star SAM-1 modem [53]. Finally, DeveNET, a company that mainly produces communication and localization equipment for divers, supplies Sealink [54], an affordable and low power acoustic modem that provides a range up to 8 km at a datarate of 80 bit/s.

To establish communication links of less than 500 m, a High Frequency (HF) acoustic modem, with a carrier frequency of at least 40 kHz, can be used. The market of HF modems is divided into two segments: low-rate acoustic modems and high-rate acoustic modems. While the former have a price below 2 kEUR, and are suitable for low rate (less than 200 bit/s) communication in shallow water up to a distance of 200 m [55, 56, 57], the latter have the same price of MF and LF acoustic modems (between 8 and 10 kEUR, depending on depth rate requirements), and, for example, can be used for sending a still-frame slide show-like video feedback from an underwater vehicle [42, 58, 59, 60, 61] to a control station, as they can perform transmissions with a bitrate of more than 30 kbit/s [6].

Although several high rate acoustic modems are available in the literature, only few of them are commercial off-the-shelf products. The maximum rate of a LinkQuest [28] modem is 35.7 kbit/s (with a directional beam pattern), while EvoLogics S2CM HS [6] is the off-the-shelf modem that provides the highest bitrate (62.5 kbit/s up to 300 m in good channel conditions, although its actual maximum throughput would typically be about 30 kbit/s). This last modem has been used in [60] to perform an in-tank low quality video streaming, where the transmission bitrate selected by the modem during its initial handshake phase [22] was 31.25 kbit/s, and the actual throughput obtained 20 kbit/s. In good channel conditions, this performance can also be obtained with other EvoLogics models, such as S2C 48/78 and S2C 42/65. While the former is optimized for horizontal communications, the latter is suitable for vertical links. The S2C HS, instead, has an omnidirectional transmitter. Modems that can provide a higher bitrate are either University non-commercial systems [58, 61, 62, 63], or company prototypes waiting for a bigger market demand before becoming available off-the-shelf [64]. The Marecomms Robust Acoustic Modem (ROAM), recently developed in partnership with Geospectrum, achieves a throughput of 26.7 kbit/s at a distance of 600 m while operating in the HF band in shallow water. The test has been performed in the presence of Doppler as the nodes were floating randomly with a speed between 0.5 and

1 knot [65]. With the new version of the modem, the manufacturer expects to achieve 50 kbit/s within a range of 1 km. The ROAM modem can also operate in the MF band, providing a bitrate of 13 kbit/s.

The most representative underwater acoustic modems with omnidirectional beam pattern that can also be employed for communications in shallow water scenarios are summarized in Table 1.1.

1.1.2 Optical Communications

Although acoustic modems are the typical solution for underwater communications, their bandwidth is very limited. The need for high speed communications in the underwater environment has pushed the realization of optical devices that can transmit data within short distances at a bitrate on the order of one or more Mbit/s (up to few Gbit/s at very short ranges, depending on the model and the water conditions). Indeed, unlike acoustic communications, optical communications are more suitable for ranges up to 100 m, especially in deep dark waters, and are not affected by multipath, shipping and wind noise, as their performance mainly depends on water turbidity and sunlight noise [7]. In fact, high turbidity scatters and attenuates the optical field, whereas ambient light may become a significant source of noise, making transmissions close to the sea surface more difficult. The turbidity coefficient, called attenuation coefficient, is composed by the sum of scattering and absorption coefficients [67]. The former depends on the quantity of particles, such as plankton, dissolved in the water. The plankton exists due to chlorophyll effect, that happens only where the solar light reaches the medium, i.e., in shallow water, up to a depth of 100 m. The latter, instead, is an inherent optical property of the medium. Blue and green lights, which have a wavelength of 470 and 530 nm respectively, are the most widely used for underwater optical communication [68], as these wavelengths are the least attenuated in deep and shallow water, respectively. Intuitively, in order to understand which of the two wavelengths is less attenuated in a certain scenario, we just need to observe the color of the water in the presence of white light (e.g., the sunlight). If the color of the water is blue, the best wavelength to use is around 470 nm (that is the typical case of a deep water deployment), otherwise, if the water color is green the wavelengths around 530 nm should be employed (that is the typical case of a deployment close to the surface).

Similarly to acoustic modems, also optical transceivers are designed to perform best in some scenarios, therefore an optical modem that outperforms all the others in all possible conditions does not exist. Specifically, we can divide the optical modems in models composed by a Light Emitting Diode (LED)-based transmitter designed for hemispherical communications [69, 70, 71, 72], and models composed by a high directional laser-based transmitter [73, 74, 75, 76, 77, 78]. Although the latter achieves a throughput from 10 to 100 times higher than the former it also requires perfect alignment between transmitter and receiver, a condition that can be difficult to obtain in some case (e.g. when mobile vehicles are involved in the network).

We can also divide the optical modems in models tailored for dark water medium range (MR) communications (up to 100 meters) [8, 69], and devices designed for short

Table 1.1: Performance figures of some acoustic modems with omnidirectional beam pattern

	Manufacturer and model	Max Range	Bit rate
LF	Sercel MATS3G 12 kHz [31]	15 km	{20, 200} bit/s secure data with coding, {0.8,7.4} kbit/s high rate with no coding
	EvoLogics S2C R 7/17 [6]	7 km	{1, 7} kbit/s raw bitrate with no coding, {0.6,4} kbit/s net datarate with coding
	WHOI MicroModem [32]	[6, 11] km	{0.2, 5.4} kbit/s in vertical links [66]
	LinkQuest UWM10000 [28]	7 km	5 kbit/s raw bitrate, 2 kbit/s payload datarate
	Kongsberg cNODE LF [33]	few kms	up to 4.5 kbit/s
	Benthos ATM 960 LF [27]	6	{0.08, 2.4} kbit/s
	AQUATEC AQUAmodem1000 [29]	10 km	{0.1, 2} kbit/s
	DiveNET: Sealink R [54]	2.5 km	{0.56, 1.2} kbit/s in shallow water
	Develogics HAM.NODE [30]	30 km	145 bit/s
	DiveNET: Sealink {C,S} [54]	8 km	80 bit/s in shallow water
MF	Sercel MATS3G 34 kKHz [31]	5 km	{20, 300} bit/s secure data with coding, {1, 24.6} kbit/s high rate with no coding
	LinkQuest UWM2000 series [28]	1 km	17.8 kbit/s raw bitrate, {0.3, 6} kbit/s payload datarate
	Subnero WNC [39]	[3, 5] km	15 kbit/s
	EvoLogics S2C R 18/34 [6]	3.5 km	{1, 13.9} kbit/s raw bitrate with no coding, {0.6, 9} kbit/s net datarate with coding
	Popoto Modem [35]	[1, 8] km	{0.08, 10} kbit/s
	Sonardyne 6G [37]	1.5 km	{0.2, 9} kbit/s
	Kongsberg cNODE MF [33]	1 km	up to 4.5 kbit/s
	Benthos ATM 960 MF, band C [27]	1.5 km	{0.08, 2.4} kbit/s, 15 kbit/s possible in very quiet deep ocean environments
	Applicon Seamodem [36]	100s of m	{0.75, 2} kbit/s
	DSPComm Aquacomm Gen2 [38]	8 km	{0.1, 1} kbit/s
	Blueprint Subsea X150 USBL Beacon [40]	1 km	100s of bit/s
	DiveNET: Sealink M [54]	1 km	80 bit/s
HF	Rutgers MIMO modem [61]	10s of m	{100, 250} kbit/s
	Northeastern SEANet prototype [63]	10s of m	{41, 250} kbit/s
	BaltRobotics Prototype [64]	[100, 200] m	{1, 115} kbit/s
	MIT Prototype [58]	200 m	100 kbit/s
	FAU Hermes modem prototype [62]	150 m	87.7 kbit/s
	EvoLogics S2C M HS [6]	300 m	{2, 62.5 kbit/s raw bitrate with no coding, {1.2, 35} kbit/s net datarate with coding
	Marecomms ROAM Prototype [65]	0.6 km 1 km expected	26.7 kbit/s actual throughput in shallow water 50 kbit/s expected in the new version
	AHOI modem [55]	200 m	200 bit/s
	Waterlinked M64 [56]	200 m	64 bit/s

range (SR) communications (up to 10 meters) in high ambient light environments [79]. In the MR class, we can find the Sonardyne BlueComm 200 [8], equipped with a very sensitive receiver based upon a photomultiplier. This modem achieves a hemispherical transmission rate of 10 Mbit/s up to a distance of 100 m, but only in deep, dark waters. The same modem would perform poorly in the presence of light noise, due to the saturation of the receiver: for this reason, Sonardyne designed an ultraviolet version of this modem, able to achieve a maximum range of 75 m even in the presence of some ambient light. Still, from our experience both models are unable to establish a communication link when deployed few centimeters above the sea surface during day time. Similar issues have been experienced with the Hydromea Luma 500ER [69], able to cover, in good conditions, more than 50 m with a bitrate of 500 kbit/s (beam pattern 120°), the Ifremer optical modem [71], that can communicate at a similar range with a bitrate of 3 Mbit/s, and the early-stage version of the ENEA Proof of Concept (PoC) prototype [80]. Also the AquaOptical modem developed by MIT [70] can perform MR communications in low solar noise conditions, by reaching a maximum range of 50 m with a rate of 4 Mbit/s. The beam pattern of both Ifremer and MIT modems is 100°, while the ENEA optical modem is omnidirectional. Customized LED-based MR optical modems are developed by Penguin ASI [81, 82]; the maximum performance of their system is in the order of 100s of Mbit/s at hundreds of meters, but comes at the price of very bulky and expensive modems that are only suitable for extremely specialized applications, such as deployment in heavy size working class Remotely Operated Vehicles (ROVs) or similar vehicles.

Models designed for SR communications, instead, typically overcome the ambient light noise issue by employing a noise compensating mechanism to avoid receiver saturation, at the price of lower bitrate and range. This mechanism typically consists in measuring the average noise at the receiver, and in injecting a signal with equal intensity but opposite sign at the receiver unit. The BlueComm 100 [8], for instance, can be used in all water conditions, including shallow water in daytime, to transmit at a rate of 5 Mbit/s in SR, at a maximum distance of 15 m. Its beam pattern is 120°. Similarly, the Sant'Anna OptoCOMM modem [72] can establish a 10 meter communication link at a speed of 10 Mbit/s, when both receiver and transmitter are deployed just half a meter below the sea surface. They use optical lenses to reduce the beam aperture angle to 20°, and reduce the receiver field of view to 70° to limit the sunlight noise. Also ENEA developed a solar light noise cancellation mechanism for their new version of the optical modem: preliminary results declared by ENEA proved that their new prototype can now communicate in high ambient light conditions, at the price of a reduced bitrate. Another commercial off-the-shelf optical modem for SR is the AQUAmodem Op1 [79], which achieves 80 kbit/s at 1 m, with a beam pattern of 34°. The company declares that the modem is affected by direct sunlight noise but is generally robust to low sources of ambient light noise, and can be used in the presence of ROV lights without compromising the communication link. The same happens for the CoSa optical modem [83], able to reach 2 Mbit/s at up to 20 m, with a transmitting beam aperture angle of 45°, and a receiver field of view of 90°. Another low-cost modem that is quite

Table 1.2: Performance figures for representative optical modems

	Manufacturer and model	Max Range	Bit rate
Laser	FUDAN modem [75]	34.5 m	2.70 Gbit/s
	Oceanit ultra [77]	100 m	1 Gbit/s
	USTC modem [76]	100 m	500 Mbit/s
	Sonardyne BlueComm 5000 [74]	7 m	500 Mbit/s
	SA Photonics Neptune [85]	200 m	200 Mbit/s customized, with beam steering capabilities
	MC100 [73]	10 m	95 Mbit/s
	JAMSTEC modem [78]	50 m	20 Mbit/s
LED SR	Sant'Anna OptoCOMM [72]	10 m	10 Mbit/s
	Sonardyne BlueComm 100 [8]	15 m	5 Mbit/s
	CoSa optical [83]	20 m	2 Mbit/s
	ENEA PoC [80]	1 m	2 Mbit/s
	IST Medusa Optical Modem [84]	10 m	{20, 200} kbit/s
	Aquatec AQUAmodem Op1 [79]	1 m	80 kbit/s
LED MR	Penguin Automated Systems [81]	[10, 300] m	{1.5, 100} Mbit/s custom
	Sonardyne BlueComm 200 [8]	120 m	10 Mbit/s
	Sonardyne BlueComm 200 UV [8]	75 m	10 Mbit/s
	MIT AquaOptical modem [70]	50 m	4 Mbit/s
	Ifremer optical modem [71]	60 m	3 Mbit/s
	Hydromea Luma 500ER [69]	50 m	500 kbit/s

robust to sunlight noise is the optical modem developed by IST [84], that can reach 200 kbit/s at a maximum distance of up to 10 m and, conversely to the other modems presented so far, uses green instead of blue LEDs, as it is specifically tailored for shallow water operations. This modem uses optical lenses to reduce the beam aperture angle to 12°, and an optical filter to reduce the sunlight noise.

The most representative underwater optical modems are summarized in Table 1.2.

1.1.3 Radio Frequency and Magneto-Inductive Communications

Also electromagnetic radio frequency and magneto-inductive communications can be used underwater. Compared with acoustic and optical waves, the RF waves can perform a relatively smooth transition through the air/water interface [86, 87]. This benefit can be used to achieve cross-boundary communication: for instance the authors in [88] used this concept to pilot an ROV deployed up to 45 cm below the water surface. Another advantage is that RF and MI are almost unaffected by water turbulence, turbidity, misalignment between transmitter and receiver, multipath, acoustic and solar noise, that are the main causes of poor performance of either optical or acoustic modems when used in practical scenarios. For these reasons, when in range, RF and MI can provide a much more stable link than optical and acoustic communications, with less disruptions, thus,

in our opinion, they should be preferred to the other media whenever the bitrate and the range required by the application can be supported. However, their communication range is usually limited to no more than a few meters. Inductive modems [89, 90, 91], for instance, are often deployed in mooring systems [92, 93], as they enable communication over jacketed mooring lines, and can be used to retrieve data from instruments such as Conductivity, Temperature, and Pressure sondes (CTDs) and Acoustic Doppler Current Profilers (ADCPs), by substituting physical connectors and the need for dedicated cables for communication. These modems generate a low frequency signal that travels in the mooring line, and can only substitute mechanical connectors for low rate communications (up to 5 kbit/s). Also RF modems [83, 94] can be used to replace the mechanical connector of cables in very short range, but they provide broadband communications (up to 100s of Mbit/s), and thus can support high rate-demanding applications, such as real-time control and high quality video streaming. For example, Hydromea uses an RF connector in the umbilical cable of the EXRAY ROV [95], where the vehicle's tether can be disconnected to perform an autonomous mission, before being reconnected again. Also in this case the communication range is in the order of few centimeters. WiSub supplies the Maelstrom connector [96], able to support both power and data transfer via RF. The communication, based on a microwave link, has a rate of 100 Mbit/s up to a distance of 5 cm between the connectors. Similar devices are sold also by Blue Logic [97]. Broadband RF modems can also be employed in docking stations to quickly download data from an AUV [98]. For instance, the WFS Seatooth S500 [94] RF modem provides a bit rate up to 100 Mbit/s up to a range of 10 cm, and the Lubeck University of Applied Science developed the CoSa underwater WiFi [83], with a rate of 10-50 Mbit/s up to 10 cm.

These examples prove how RF communications can achieve high transmission bitrates underwater, although their communication range is very limited. Indeed, RF communications suffer from RF interference and are prone to very strong attenuation in salty waters, where the conductivity of the medium is larger than in fresh waters. A range up to few meters (SR) can be reached with RF modems, at the price of a lower bitrate. For example, INESC Tec developed a dipole antenna prototype [3] to support 1 Mbit/s communication at 1 m, and the Lubeck University of Applied Sciences developed a dipole [83] antenna, to communicate with a rate of 0.2 to 1 Mbit/s and a range of 1-8 m, depending on the water conditions (i.e., 1 meter in salty water, 8 meters in fresh water).

Although in air MI communication is outperformed by RF modems, as the latter can achieve a higher bitrate and a longer range, underwater MI modems are almost unaffected by the change of medium, while the electrical field is strongly attenuated. Indeed MI modems are proved to reach a bitrate of few kbit/s at tens of meters, both in air and underwater [99]. Dalhousie University developed an MI prototype that achieves 8 kbit/s at 10 m [100], to perform low-rate low-latency communications. With MI modems, longer distances can be achieved at the price of a lower bitrate. For instance, the authors in [99] established a directional link with a maximum range of 41 m, and an omnidirectional link with a range of 26 m. Both links were providing a datarate of

512 bit/s. With their new modem design recently presented in [101] they were able to achieve 1 kbit/s at a 40 m distance with an omnidirectional beam pattern. Nearly 20 years ago, the authors in [102], instead, demonstrated a 153 bit/s communication link at a distance of 250 m, and 40 bit/s at a distance of 400 m.

Very Low Frequency (VLF) and Extremly Low Frequency (ELF) radio signals have been extensively used during the cold war to communicate from inland control stations to submarines. The drawbacks of these systems are the low rate and the need for a very large and high-power consuming inland antenna. Indeed, VLF can provide a 300 bit/s one way communication link from shore to the submarine up to a distance of 20 m below the sea surface, and requires a broadcast inland antenna with a size between 300 m and 2 km. For example, the Sweden Grimeton Radio Station [103] uses a set of antennas 1.9 km long, each with an RF power peak of 200 kW.

ELF can also be used to communicate from land to submarines (one way): they reach up to 1 bit/s [104] at a range of several hundreds of meters below the sea surface, but require a grounded wire inland antenna (ground dipole) with a size of several tens of kilometers, and a transmission power in the order of millions of watts. Due to the high cost of the deployment, US, Russia, India, and China are the only nations known to have constructed ELF communication facilities. For instance, the US ELF system employed a ground dipole antenna 52 km long [105], while the Russian system used an antenna 60 km long [106]. This system has been typically used to signal one way coded messages to the submarine's commander to resurface to receive more information via other means.

The most representative underwater MI and RF communication systems are summarized in Table 1.3.

1.2 Security in Underwater Communications

As previously mentioned, the areas of application of underwater networks can range from industry to the military field, and, depending on the scenario, network security can be an important issue. As for terrestrial networks, also underwater networks are prone to malicious attacks due to the wireless nature of the communication medium. Different types of DoS attacks can be performed, from simple attacks in which the malicious node needs neither specific knowledge of the protocols nor special computational capabilities, to more complex attacks in which the attacker exploits the behavior of the protocol to damage the network. Attacks as jamming [107, 108], where a malicious node injects a signal into the channel to interfere with the ongoing transmission and degrade the communication quality, and replay attack [109, 110], where a node intentionally reinjects the received packets into the network to saturate it or to lead the protocol to some inconsistent state, belong to the first category. Instead, other DoS attacks such as sinkhole [111, 112], in which a node attracts the largest possible amount of packets and then selects which to forward and which to drop, and resource exhaustion [113], in which a malicious node exploits some vulnerabilities of the protocol to exhaust a resource such as the node's battery or the channel bandwidth, need a deeper knowledge

Table 1.3: Performance figures for representative RF/MI underwater modems

	Manufacturer and model	Max Range	Bit rate
(E)VLF-RF	VLF (3-30) kHz [103]	20 m below the sea surface, inland antenna size = 2 km	300 bit/s one way, from land to underwater
	ELF (3-300) Hz [105, 106]	hundreds of meters below the sea surface, inland ground dipole size = [52, 60] km	1 bit/s one way, from land to underwater
MI	Dalhousie Univ. Prototype [100]	10 m	8 kbit/s
	inductive modems for mooring lines [89, 90, 91]	up to few kilometers from tx using the mooring line, few cm from the mooring line	{1, 2} kbit/s
	MST Prototype [101]	40 m	1 kbit/s
	CSS/MISL Prototype [102]	[250, 400] m	{153, 40} bit/s
RF	WiSub Maelstrom [96]	5 cm	100 Mbit/s
	CoSa WiFi [83]	10 cm	{10, 50} Mbit/s
	WFS Seatooth S500 [94]	10 cm	10 Mbit/s
	INESC TEC Dipole [3]	1 m	1 Mbit/s
	CoSa EF Dipole [83]	[1, 8] m	{0.2, 1} Mbit/s
	WFS Seatooth Mark IV SR [4]	[5, 7] m	2.4 kbit/s
	WFS Seatooth Mark IV MR [4]	[30, 45] m	100 bit/s

of the protocol behavior and therefore belong to the second category.

However, despite these threats related to network security, most of the effort in underwater communication has been addressed towards the design of new communication protocols and modulations to increase the reliability of underwater communications. Security aspects of underwater wireless acoustic networks, instead, have not been widely studied so far and require a dedicated effort, due to the fact the countermeasures used in wireless terrestrial networks cannot be directly applied to the underwater domain. For instance, a freshness index based on the generation time of a data packet, used to check if a malicious node is performing a replay attack [114], is a valid countermeasure in a wireless network, where the packet header already contains the packet generation timestamp, but could not always be a valid countermeasure in underwater networks, where only few bytes are used for the packet header, as the communication overhead needs to be minimized due to the low datarate. In addition, such countermeasure, that restricts the time validity of a packet transmitted in an underwater network, needs to be carefully investigated since it could result in the drop of legitimate packets, as underwater networks may be characterized by a very large Packet Delivery Delay (PDD), of the order of up to a few minutes. Moreover, in other attacks such as jamming, the countermeasures should be designed taking into account the large propagation delay due to the low propagation speed of acoustic waves, which is a negligible factor in terrestrial networks but that could lead to different trade-offs in acoustic networks.

In order to defend against a DoS attack there are two possible strategies. The first consists in providing a countermeasure to specific attacks, analyzing how the network behaves when these attacks are performed [114, 115]. On the one hand, this solution is very effective, as it usually allows to both detect an attacker and avoid the DoS; on the other hand, it requires the knowledge of the attacker setup, and an extended simulation study where different variations of the attack are performed and analyzed. In case a different attack is applied, a different countermeasure needs to be taken. The second strategy, instead, consists in using a trust mechanism, based on the reputation of the nodes [116, 117], to identify whether or not a node can be expected to follow the protocol rules of the underwater network. While this strategy is limited to identifying the attacker rather than limiting its effect, its main advantage is that it can be applied to many different attacks, thus providing a general defense solution.

In the remainder of this section, we present the state of the art of the main attacks and countermeasures that will be analyzed in this thesis.

1.2.1 Related Work

1.2.1.1 Jamming Attacks and Countermeasures

Physical layer jamming is a well-studied and common DoS attack technique, in both terrestrial and underwater networks [118]. Its basic principle is particularly intuitive: in order to block the reception of a packet, the attacker increases the noise level at the receiver by injecting a single-tone or white Gaussian noise signal [119] into the channel in the same frequency band as the transmitter. More sophisticated attacks can target the packet preamble [120] in order to affect the synchronization and metadata decoding processes, which can be vulnerable to more targeted jamming attacks, saving the attacker energy as it does not need to jam the longer body of the packet. All these techniques can be adapted to prevent the transmitter from escaping the jamming using spread spectrum modulation or frequency hopping [121], and more flexible approaches can involve power and modulation control from the jammer to maximize the effectiveness of the attack. For a more comprehensive taxonomy of jamming attacks and defenses, we refer the reader to [122].

Defense techniques have evolved in parallel with jamming attacks, as network designers adapt their solutions to potential attacks, which are then updated in a continuous arms race. While jamming attacks can make easy prey of unaware transmitters that use simple duty cycling to save energy [123], the literature on jamming countermeasures involves active reactions such as power control [124] and channel-hopping [125]. Game theory is a tool that can be employed to model a scenario in which both the jammer and the transmitter have some adaptive capabilities and can adapt to each other. The main drawback of active defense strategies is that they require more energy, reducing the lifetime of the nodes. So-called “vampire” attackers can then exploit this to deplete the transmitter’s battery. In this case, the game theoretical model needs to include energy consumption, either as an explicit constraint [126] or by considering nodes with a limited battery [127]. The latter case can be solved by applying dynamic

programming techniques [128], as there is a limited number of possible energy states, which can only decrease if the nodes have no energy harvesting capabilities.

Some recent works have analyzed jamming with game theory in an underwater context, exploiting the peculiar nature of Underwater Acoustic Networks (UANs). For example, [129] applies a reinforcement learning deep Q-network-based transmission scheme, using movement as a countermeasure against a jamming attack in a mobile underwater acoustic network. The jammer sends acoustic signals with the same band as the transmitter, and each agent can decide its own transmission power level. The problem is modeled as a dynamic game in which all nodes are power-constrained; the winner of the game is the last node to completely deplete its battery. The results are proven via both simulation and a pool test, in short range. [130, 131] investigate friendly jamming as a potential help for the transmitter in preventing eavesdropping, and in [132] the authors study the effect of different types of jamming models, such as random, reactive, constant and white noise jammers, using real commercial and prototype modems. Other works in this area are described in [24], which surveys the recent literature on underwater jamming. However, none of them considers the long propagation delays that characterize acoustic transmission as a sort of natural defense against jamming. In [133], the authors address the effects of propagation delay on the detection of reactive jamming without however analyzing the impact of this delay on defensive countermeasures.

While reactive jamming, in which the attacker only transmits the jamming signal if it senses a packet being transmitted, is the standard assumption in terrestrial wireless networks, the long propagation delay in acoustic networks might make it impossible for the jammer to sense the packet and jam it in time. In this case, the jammer might need to adopt a blind strategy, jamming at random instants and hoping to block a packet transmission.

1.2.1.2 Replay Attacks and Countermeasures

A replay attack is an attempt to perform a malicious action by recording valid data transmissions and repeating or delaying them in order to impersonate a valid user in a network. The replay attacks can be classified in straight replays, where the packets are intended for the same destination but delayed, and deflections, where the packets are directed to other than the intended recipient, e.g., reflected back to the sender, or deflected to a third node [109]. In the classical replay attack, the intruder records the data transmitted in the channel for a certain amount of time, and then replays the whole recorded signal as it is [134]. A smarter attacker can also identify the packets from the signal, decode them and decide whether to transmit all or a part of them, selected in order of arrival or chosen at random, one or multiple times [135].

Replay attack countermeasures for wireless terrestrial networks have been discussed for a long time in the literature. Traditional security methods (e.g., cryptography) do not provide complete protection against replay attacks [136]. Some attempts to use timestamp methods in the packets have provided some benefits [137]. The use of timestamps, however, could not be applicable in case of lack of synchronization in the

network [136], however, given the low bandwidth and the large packet delivery delay experienced in underwater acoustic networks, in the underwater scenario a large validity period can be set, by overcoming the synchronization issue and thus indicating that a timestamp-based solution should be investigated. The authors in [137] provided an authentication protocol for preventing replay attacks. The protocol gave a mechanism to inspect the message freshness (e.g., serial number, timestamp). Nevertheless, this system requires the exchange of several messages for sharing the keys used for the authentication of the legitimate nodes, and may not be directly applicable to a UAN. The authors in [138] and [139] have studied the effects of replay attacks in secure ZigBee networks. They showed that ZigBee networks are vulnerable to replay attacks also when using encrypted payloads and a frame counter. Authors thus suggest a full timestamp scheme as replacement of the frame counter mechanism. The authors in [140], instead, demonstrate that a protection for the replay attack based on the hash value of the bits of the packet outperforms the frame counter method. This strategy has the advantage that no additional bits need to be added to the packet, and fits well WiFi networks, where the packet header already includes a timestamp and the packet size is usually very large compared to the size of the packets transmitted in an acoustic networks. Indeed, applying this solution directly to the bits of the short packets transmitted in acoustic networks, where the packet header size is minimized and may not contain a timestamp, would lead to a high HASH collision probability. This problem could be mitigated with the addition of a timestamp to the bits used as input to the HASH function.

1.2.1.3 Trust Models

Trust is a measure of the belief that a given subject will behave according to what is expected. This measure can be applied to many different fields [141], from sociology to science, academia, journalism, economics, medicine and, finally, wireless terrestrial and underwater networks, the last being the focus of this section. The trust of a node can be based either on authentication certificates and cryptographic keys [142], or on a reputation-based system, the latter being more relevant to Mobile Ad Hoc Networks (MANET) [141]. Many works in the literature propose a reputation-based system for terrestrial wireless networks [116, 117, 143], but only a few papers address the aspects of underwater acoustic networks, and most of them only propose a preliminary analysis [144, 145].

The authors in [143] demonstrate how a watchdog-based reputation system applied to the Ad-hoc On-demand Distance Vector (AODV) routing protocol in a wireless mesh network provides significant benefits in terms of network performance when the network is under attack. This reputation extension of the AODV protocol, called AODV-REX, has been tested against malicious nodes performing blackhole and grayhole attacks. The reputation of a node computed by one of its neighbors increases when it correctly forwards received packets according to the network protocol. Conversely, if this does not happen within a certain time interval, the reputation decreases. This observation can be performed by means of the watchdog mechanism [146], i.e., the neighbors of a node

can overhear the packets it transmits even if these packets are not for them. The more interactions a node A performs with a node B, the more the reputation of B computed by A is considered solid. The reputation of a certain node is finally shared among the nodes of the network: the more the reputation values for that node differ, the less the node is trusted. While the watchdog mechanism can also be applied to acoustic networks, the proposed reputation system cannot be directly applied to underwater networks due to the disruptive nature of the acoustic channel and the overhead introduced by the signaling of the AODV-REX routing.

The authors in [116] propose an extension of the AODV routing protocol, called Trusted AODV (TAODV), where the trust of the nodes is performed using watchdog. The protocol has been designed for secure MANET. In this work, the trust among nodes is represented by opinion, which is an item derived from subjective logic [147]. An opinion can be interpreted as a probability measure containing secondary uncertainty: specifically, a node may be uncertain about another node's trustworthiness because it does not collect enough evidence. For this reason, in subjective logic an opinion is modeled using belief, disbelief and uncertainty. Subjective logic is also used in the trustworthiness model presented in [117], where the authors used the uncertainty to model the error probability of the channel, that is assumed to be constant. They also use federated learning for distributed model training using local datasets from large-scale nodes, but this method applies well to terrestrial networks where large datasets can easily be collected by observing the traffic of cellular networks, rather than to acoustic networks where only a few network deployments can be observed in reality. Conversely, subjective logic can be applied to underwater networks, in order to model the case the transmission of a forwarded packet is not observed due to adverse conditions of the acoustic channel rather than the intentional misbehavior of a node. In this case, the model should consider the nature of the acoustic channel, where the error probability is not constant in time but changes during the day.

Sharing the trust metrics among nodes can help build a reputation system in a cooperative way. The drawback of this solution is that it is prone to attacks where the malicious node transmits wrong reputation scores of the other nodes, causing severe damages to the network. However, in [148] the authors prove that, as soon as all nodes share enough reputation information, the effect of a malicious node sharing wrong information on purpose is mitigated. A Bayesian approach is used to update the reputation, taking into account the possibility that the reputation value may be received from a malicious node. In their paper they also use a discount factor to weigh recently observed events more than events occurred in the past, thus addressing the case when a node changes its behavior after a certain amount of time.

Security aspects of underwater acoustic networks have been partially addressed, since only recently researchers have started focusing their work on these aspects. The simulation study in [145], for instance, uses trust in underwater networks to enhance location privacy rather than to detect intruders and malicious nodes. ITrust [144], instead, is an anomaly-resilient trust model based on isolation forest for underwater acoustic sensor networks. ITrust is composed of two sequential stages: data fusion

(by aggregating various trust metrics) and defective node detection through the trust model. The model has been evaluated via simulation, with the simplistic assumption that the acoustic noise power spectral density can be computed with the analytical formulas presented in [5].

1.3 Main Contributions and Thesis Structure

The thesis tackles the problem of the design of underwater protocols to enable the usage of a large number of low cost underwater modems in industrial scenarios, such as the smart port introduced at the beginning of this chapter. Specifically, I designed a polling-based MAC protocol to retrieve data by means of underwater vehicles in a harbor scenario. I also extended the DESERT Underwater network simulator [149], by including the polling protocol into the simulator, and the WOSS Framework [150], adding the Hamburg port bathymetry. The goal was to simulate the propagation of acoustic waves in the harbor obtaining an accurate channel model for the network simulation. Finally, I participated in the design and realization of the tests carried out to validate the polling protocol through a lake trial. In addition, the thesis deals with the still little studied problem of security in underwater networks introduced in Section 1.2, proposing new analyses and tools to counteract possible attacks in those networks. Specifically, I wrote the code of a game-theoretic framework employed for the analysis of jamming attacks, and I also participated in the design of security mechanisms for replay attacks and a reputation system for underwater networks. Finally, I designed a channel-based trust model, analyzing it through both an analytical formulation of the problem and the implementation of the trust mechanism in the DESERT Underwater network simulator. One of the goals achieved in this thesis is the extension of the DESERT Underwater network simulator with new modules to study security issues in underwater networks.

The thesis is organized as follows. The first part is about the employment of underwater networks into a smart port scenario. Specifically, we focus on the design of a polling-based MAC protocol (UW-POLLING) for the collection of data from underwater sensors. We then describe a lake test performed employing the UW-POLLING protocol. Finally, we analyze how to forward the data from the underwater network deployment to the shore.

- Chapter 2 describes the polling-based MAC protocol designed for the collection of data from submerged nodes in the port environment. We compare the performance of a commercial modem and a low-cost prototype modem. We then assess the results with a multimodal solution in the scenario of the port of Hamburg, using the bathymetry of the harbor to perform the simulations. We also describe the test performed at lake Kreidesee in Hemmoor (Germany), where we tested the polling protocol in a lake environment with an ASV collecting the data from buoys equipped with the AHOI modem. We analyzed the performance of the polling protocol with different topologies. The chapter is based on **[J1, J4]**.

- Chapter 3 tackles the problem of the transmission of the collected data from the underwater network to the shore. We analyze the feasibility of using a low-cost radio solution, such as LoRaWAN, to forward the data. We assess the performance considering different acoustic modems and a different number of sink nodes (used to both collect the data from the AUV and communicate to the shore). We study whether the bottleneck for the transmission is due to the underwater network or to LoRaWAN. The chapter is based on **[C6]**.

The second part of the thesis debates about security issues in underwater networks, starting from simple attacks that do not require any knowledge about the employed protocols, and then moving to more sophisticated attacks. First, we analyze through a game-theoretical framework jamming attacks in acoustic communications. We then assess the effect of simple attacks, such as replay attacks, and analyze the security of the polling-based protocol presented in the previous part. Finally, we move to the design of a trust model for underwater acoustic networks.

- Chapter 4 presents a game-theoretical framework for the analysis of jamming attacks in acoustic communications. First we analyze the effect of the jammer as a function of the distance between the jammer and the receiver in a complete information scenario, in which each node knows everything about their opponents. We then perform a bayesian analysis relaxing the assumption of complete information by making the distance between the jammer and the receiver unknown to the transmitter, and comparing the obtained results with the complete information scenario. Finally, we analyze the effect of the geometry of the network deployment on the jamming effectiveness, i.e., we asses the impact of the low propagation speed of acoustic waves on the performance of jamming attacks. The chapter is based on **[C4, J2, C5, J6]**.

- Chapter 5 describes different types of replay attacks and some possible countermeasures based on either a timestamp or a HASH mechanism. In this chapter the attacker does not have any particular computational capability, and there are not assumptions about the knowledge by the attacker of the protocol stack used in the network. The chapter is based on **[C8]**.

- Chapter 6 moves a step forward by allowing the attacker to know the protocol stack of the network, and thus exploiting its knowledge while performing the attack. First, we test the effectiveness of the replay attack against the UW-POLLING protocol described in Chapter 2 with an attacker replaying the signaling packets of the protocol. Then, we present a reputation system in which each node has a score about its neighbors. The score is decreased each time the neighbor is not accomplishing a task expected by that node. When the score goes below a threshold, the neighbor is inserted into a black list and no longer considered for future interactions. The system has been evaluated with the UW-POLLING protocol and the SUN routing protocol [151]. The chapter is based on **[C10]**.

- Chapter 7 describes a channel-based trust model specifically tailored for underwater acoustic networks. The trust model can be used to detect malicious nodes not acting according to the protocol rules, e.g., not participating in the forwarding process of the routing protocol. The model takes into account the peculiarity of the underwater channel by considering the channel state when evaluating the behavior of a node. We test the model both analytically and through simulations. The chapter is based on **[J6, C12]**.
- Chapter 8 draws some conclusions.

Note

Part of this chapter is based on **[J3,J5,J6,C8]**

Part I

Data Collection in a Smart Port Scenario

Chapter 2

UW-POLLING: a MAC Protocol for Data Muling

2.1 Introduction: The Smart Port Scenario

Waterborne handles about 90% of goods transportation [152]. New technologies can enhance the shipping activities in a port environment, to pursue both cost reductions and improvements in safety of human operators and equipment. In this context, the goal of a smart port is to revolutionize the shipping and near-shore operations by offering robotic aided services via interconnected Unmanned Vehicles (UVs), equipped with specialized sensor technology, a reliable data transfer cloud network for above water and underwater communication, a monitoring station, and a real-time web-based user interface. An example of smart port is envisioned in the RoboVaaS project [12], which aims to address the challenges needed towards this new harbor concept. The high level of autonomy implied in a smart port is expected to be reached by exploiting the most innovative communication technologies and advanced robotic vehicles to perform both autonomous missions and tasks remotely controlled by a human operator with the aim to improve shipping operations, offering on-demand and robotic-aided services. To enhance harbor activities, inspection services for quay walls and ship hulls, anti-grounding service, bathymetry and environmental data collection can be provided to create a smart port environment. An example of a smart port scenario is depicted in Figure 2.1.

In the quay wall and ship hull inspection, for example, an ROV tethered to an ASV connected to the shore through an RF link inspects the wall or the hull through a high resolution camera and sends back the video through the ASV [153]. In the anti-grounding service, an ASV equipped with a sonar sends back real-time bathymetry data to a ship incoming into the harbor to notify the vessel in case of grounding warning. This service can help in ports in which the bathymetry data is outdated or not available or in case of frequent change in the bottom shape caused by ship passage (as for the port of Hamburg) [154]. The data collection service, also known as data muling, consists in

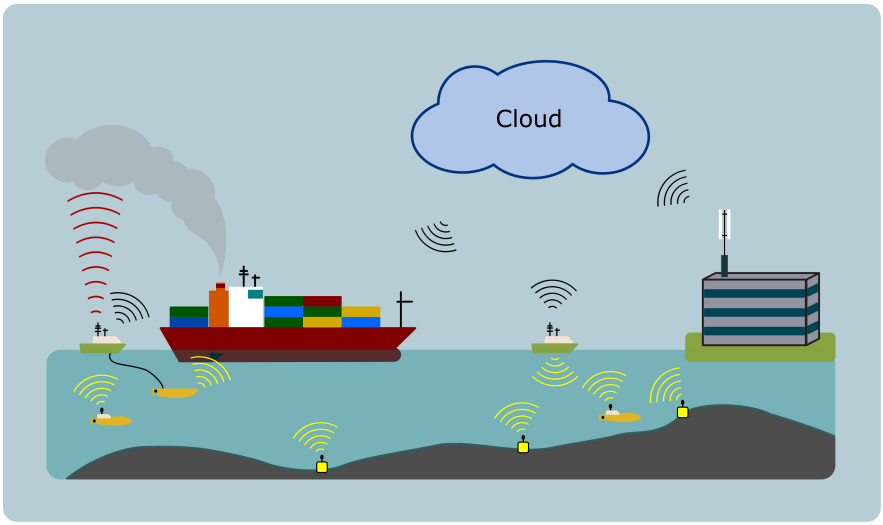


Figure 2.1: Example of possible smart port services.

collecting data from underwater sensor nodes deployed in the port area by means of an ASV or an AUV which follows a predefined path and retrieves the data from the sensors through acoustic communications. The vehicle can then be in charge of forwarding the collected data directly to the shore or transmitting it to one or more buoy nodes (sinks) that will take care of the transmission to the shore. The data retrieved from the Underwater Sensor Network (USN) can be environmental data typically employed to monitor the water quality and the water characteristics in order to inspect the impact of human activities and to monitor the level of pollution in the port area and promptly react in case of anomalous values [11]. Figure 2.2 shows the data muling service. The remainder of this chapter will focus on this scenario, designing and studying a MAC protocol to be employed in the data collection service.

The data collection service involves the use of underwater communications to collect data from the sensor nodes. In this chapter we design and analyze a polling-based MAC protocol (UW-POLLING) to collect the data from the USN, tailored to the number of nodes in the network and the challenges of the underwater acoustic channel [5]. In addition, the protocol has the ability to automatically adapt the choice of the protocol parameters according to the estimated number of nodes in the AUV neighborhood: the optimal choice of the network parameters aims to maximize the throughput of the network. We also present the design of a protocol stack for a hybrid multimodal USN, composed by two different models of HF short-range acoustic modems: the smartPORT Acoustic Underwater Modem (AHOI modem) [55], employed to communicate from the nodes of the USN to the AUV, and the high rate EvoLogics HS modem [6], used for the communication between the AUV and the sink. While the latter is a well-known commercial modem, the AHOI modem is a small, low-power and low-cost acoustic underwater modem developed by the Technical University of Hamburg. The selection of the proper modem to use for each transmission is performed thorough a multimodal

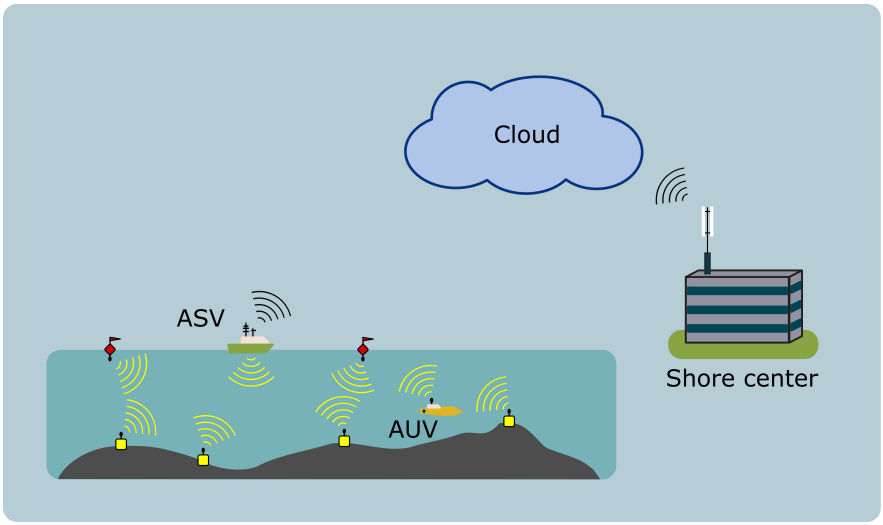


Figure 2.2: Underwater data collection service, where both an ASV and an AUV collect data from static underwater sensor nodes.

layer, called UW-MULTI-DESTINATION, that decides which technology is used to transmit the data.

The data muling scenario has been evaluated with simulations based on field measurements: to this aim, both the AHOI modem performance figures and the bathymetry of the port of Hamburg have been included into the DESERT Underwater network simulator [149].

2.1.1 Related Work on Data Muling Protocols

In the context of data collection from sensor nodes, and more in general in the acoustic underwater environment, the design of the MAC protocol plays an important role, since packet collisions and subsequent retransmissions may have a strong impact on the network. Indeed, due to the typical low bitrate of the acoustic modems and the high propagation delay, each retransmission significantly increases the channel occupancy.

In such an environment, in [155] the authors compare two MAC random access protocols against a polling-based MAC protocol in the data muling scenario. Specifically, the first random access protocol is a CSMA-based protocol called CSMA-Aloha-Trigger, the second, called Distance-Aware Collision Avoidance Protocol (DACAP), is based on Request-To-Send (RTS) and Clear-To-Send (CTS) signaling. In the CSMA-based protocol the nodes transmit their packets to the AUV in a CSMA-like fashion only after the vehicle notifies its presence with the transmission of a trigger packet. In DACAP sensor nodes are allowed to transmit their data only after the transmission of an RTS followed by the correct reception of a CTS packet. The authors show that the polling-based MAC protocol in the considered scenario always outperforms the analyzed random access protocols in terms of throughput and packet delivery ratio.

In [156] the authors describe APOLL, a polling-based MAC protocol that groups all the nodes of the network in two classes: the APOLL Controller (AC) and the Mobile Unit (MU). In APOLL, the AC can decide to schedule a REGISTRATION period in which the MU randomly selects a transmission opportunity to transmit its REGISTRATION packet to the AC. The number of opportunities and the length of each opportunity in the REGISTRATION period are decided by the AC. After the registration, the POLL-REPORT phase starts. In this phase, the AC sends a poll to the intended MU followed by data for the MU, if any, and the MU sends a report back to the AC. The POLL-REPORT phase goes on until the the AC decides to schedule another REGISTRATION phase.

2.1.2 Chapter Structure

The rest of the chapter is organized as follows: Section 2.2 describes in detail the UW-POLLING protocol, Section 2.3 evaluates the performance of the MAC protocol when employing the EvoLogics S2C HS modem, while Section 2.4 analyzes the scenario with the low-cost AHOI modem. Section 2.5 assesses the performance with the multimodal solution in the scenario of the port of Hamburg. Section 2.6 presents the lake test performed to validate the polling protocol, and finally Section 2.7 draws some conclusions.

2.2 UW-POLLING Description

This section presents a hybrid multimodal network, composed by sensor nodes, an AUV and a sink node. The sensor nodes are equipped with the AHOI modem, while the sink uses the EvoLogics S2C HS acoustic modem. The AUV collects the data from the sensors: it can behave as the sink itself, by resurfacing and transmitting the data to shore via radio link, or convey the data to a sink deployed from a buoy or an ASV. In the former case, the AUV can be equipped only with the AHOI modem, while in the latter case it has to use an EvoLogics S2C HS modem as well. In order to handle different modems the AUV requires a dedicated multimodal layer, that selects which modem to employ. This layer, called UW-MULTI-DESTINATION [157], is deployed between the routing layer and the MAC layer. For each incoming packet from the routing layer, UW-MULTI-DESTINATION first checks the packet's next hop address, and then chooses the right physical layer technology to use for the transmission. This selection is performed through a technology per node map, where the UW-MULTI-DESTINATION stores the list of physical layers available at each node. In our case this list is assumed to be known at network deployment, however, a periodic topology discovery mechanism [158] might be employed to update the list. In Section 2.5, a protocol stack that employs the UW-MULTI-DESTINATION layer will be evaluated via simulation.

UW-POLLING, the polling-based MAC protocol used in this USN, is presented in Section 2.2.1.

Table 2.1: Meaning of the most important symbols used to described the UW-POLLING protocol.

Symbol	Meaning
TrP	TRIGGER packet, sent by the AUV to the nodes
PrP	PROBE packet, sent by the nodes to the AUV
PoP	POLL packet, sent by the AUV to the nodes
$T_{b_{min}}$	Minimum backoff time a node waits before transmitting a PrP
$T_{b_{Max}}$	Maximum backoff time a node waits before transmitting a PrP
Pks_i	Number of data packets node i is going to transmit
Pks_{MAX}	Maximum number of packets a node can transmit in a polling phase
Pr_{Max}	Maximum number of PrP AUV can receive in a discovery phase
Prx, i	Total number of packets received by AUV from node i
N_{Max}	Maximum number of packets AUV can transmit to sink in a polling phase
λ	Node density, i.e., the average number of nodes deployed in 1 km^2

2.2.1 A Polling-Based MAC Protocol for Underwater Acoustic Networks

The MAC layer employed in the data muling scenario is a polling-based MAC protocol. The general behavior of the protocols is described by the state machines depicted in Figure 2.3. In particular, most of the protocol logic is implemented in the AUV (Figure 2.3a), which collects the data from the sensor nodes and, optionally, forwards the data to a sink node. All the operations performed by the sensor nodes (Figure 2.3b) and the sink node (Figure 2.3c) are executed in reaction to the AUV indications.

The UW-POLLING protocol works in two phases: the discovery phase identifies how many surrounding sensor nodes have data packets to send, and the polling phase collects the data from the nodes. In case a sink node is employed, in the polling phase the data is also forwarded to the sink. The discovery phase starts when the AUV sends a TRIGGER packet (TrP) in broadcast, revealing itself to all the surrounding nodes. Each node that correctly received the TrP and has data to transmit replies with a PROBE packet (PrP) after a random backoff time. The AUV collects all the PrPs and then starts the polling phase sending to one node at a time a POLL packet (PoP), and waiting for the data from the selected node. If the SINK receives a TrP, it replies with a PrP as well; afterwards, in the polling phase, the AUV will forward the data to the sink. A detailed description of the UW-POLLING protocol is reported in the rest of this section.

At the beginning of the discovery phase the AUV transmits the TrP to all the surrounding nodes. The TrP contains information that will be used by the nodes to send the PrP. In particular, the minimum ($T_{b_{min}}$) and the maximum ($T_{b_{Max}}$) backoff time to use for the transmission of the PrP are inserted in the TrP. This mechanism enables the AUV to possibly adapt the backoff time choice based on the network density.

The nodes that correctly received the TrP and have data packets to send reply

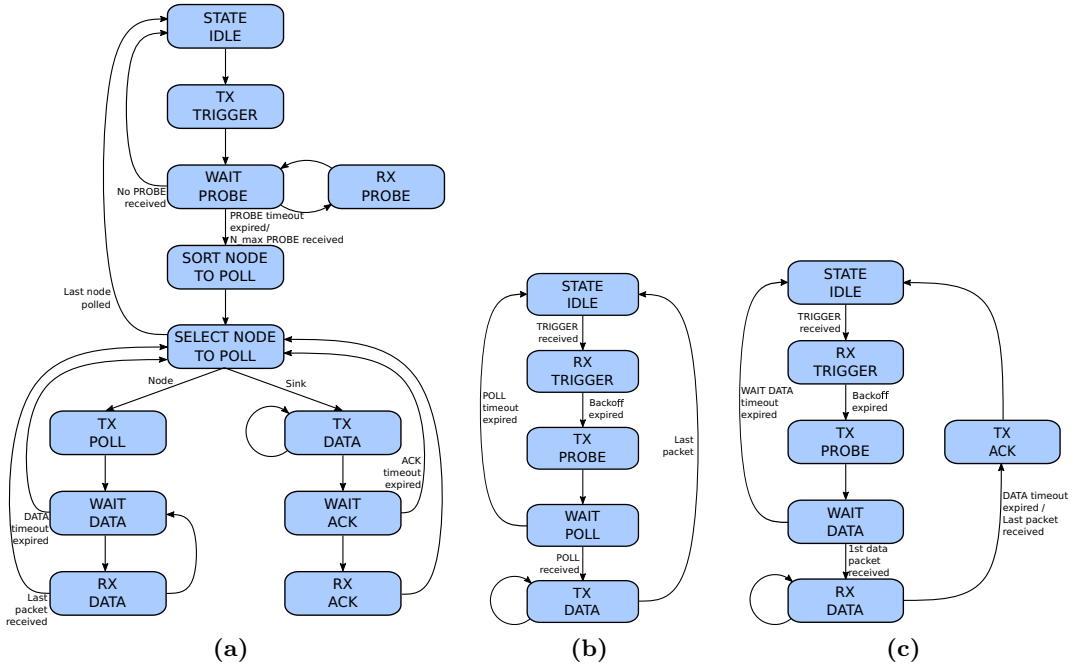


Figure 2.3: From left to right: state machine of the UW-POLLING protocol for an AUV (a), for a node (b) and for a sink (c), respectively.

with the PrP. Each node transmits the PrP after a random backoff time uniformly chosen between $T_{b_{min}}$ and $T_{b_{Max}}$. In the PrP, each node i inserts the number of data packets that is going to transmit (Pks_i) in the polling phase. This information is needed by the AUV to choose the order in which to poll the nodes. Pks_i is bounded by a maximum value Pks_{MAX} equal for each node of the network, in order to avoid that a node which holds many packets acquires the access to the channel for a long time, thereby preventing all the surrounding nodes from transmitting. If the sink received the TrP, it takes part in the contention, transmitting a PrP as well. The AUV waits for the reception of all the PrPs until either a timeout T_{probe} expires, or the maximum number of PrP (Pr_{Max}) is received.

Once the PrP is transmitted, the node waits for the PoP from the AUV. If the PoP is not received before a timeout T_{poll}, the node considers a failure in the transmission of either the PrP or the PoP, and waits for the reception of another TrP. Similarly, the sink waits for T_{data}^{SINK} for the reception of the first data packet from the AUV. If the data is not received within T_{data}^{SINK}, the sink waits for the reception of another TrP.

If the AUV does not receive PrPs, the discovery phase starts again with the transmission of another TrP. Instead, if at least one PROBE is received, the polling phase starts. Firstly, the AUV creates the POLL list, i.e., the ordered list of the nodes to be polled. The UW-POLLING protocol sorts the nodes according to a proportional fair scheduling, trying to obtain fairness in the number of packets each node transmits to the AUV. In particular, for each node i, the AUV computes a weight $w_i = \frac{Pks_i}{Prx,i}$, where

$P_{rx,i}$ is the number of data packets received by the AUV from node i so far [159]. Then the nodes are sorted according to this weight, in order to select first the nodes with the highest weight. If two or more nodes have the same weight, they are ordered depending on the time of arrival. If a PrP from the sink is received, the sink is inserted in the list as well. The sink is inserted in the first position such that the sum of the packets that the AUV expects to receive from the previous nodes in the list, and the number of data packets that are already in the AUV's queue (N_q), is greater than or equal to the maximum number of packets the AUV can transmit to the sink in each polling phase (N_{Max}). Thus, if m is the position in the POLL list, the sink is placed in position m' such that

$$m' = \min_{m} \left(N_q + \sum_{k=1}^{m-1} Pks_k \geq N_{Max} \right). \tag{2.1}$$

If this condition is never reached, the sink is inserted at the end of the POLL list.

After the creation of the POLL list, the AUV starts to POLL all the nodes in the list. The AUV sends a PoP to the first node in the list, and waits for the reception of the data packets from the selected node. Once the AUV receives all the expected data packets from node i, it polls the next one in the list. If some packets are lost, the AUV waits until the timeout $T_{data}^{AUV,i}$ expires before going to the next node in the list. $T_{data}^{AUV,i}$ is automatically computed based on the number of data packets Pks_i the AUV expects to receive from node i and the round-trip-time (RTT) between the AUV and the polled node, measured during the discovery phase.

In the PoP the AUV inserts the expected amount of time it needs to poll all the nodes in the list. This value is used by all the nodes that received the POLL packet to refine the timeout T_{poll}. Similarly, the sink can use the value inserted in the POLL to adapt the timeout T_{data}^{SINK} as well. When node i receives the PoP intended for itself, it starts to transmit Pks_i data packets to the AUV. After the transmission of the last packet, the node moves to the IDLE STATE waiting for the reception of another TrP.

If the selected node in the list is the sink, the AUV starts to transmit up to N_{Max} data packets to the sink. Once the sink receives the first data packet, it refines the T_{data}^{SINK} timeout to a value within which it expects to receive all the data packets from the AUV. The timeout is needed for the transmission of an acknowledgement (ACK) to the AUV. Indeed, if the last packet is lost, the sink needs to wait until T_{data}^{SINK} expires before sending the ACK. Instead, if the last packet is correctly received the ACK is sent immediately after the reception of this packet. For this purpose, before sending the data, the AUV inserts in each packet a unique ID (UID) and the UID of the last packet it is going to transmit in that polling phase. The lost packets are sent again in the next polling phase, according to a Selective Repeat (SR) Automatic Repeat Request (ARQ) mechanism: after the data transmission, the sink sends to the AUV an ACK containing either the UIDs of the lost packets or the UID of the next expected packet, in case no packets are lost. The AUV waits for the reception of the ACK until T_{ACK} seconds from the transmission of the last packet and, if the ACK is not received within this timeout, it considers the ACK lost and starts to POLL the next node in the list. Moreover, the sink employs a second mechanism for the packet acknowledgment. This second ACK is

inserted in the PrP and consists of a Go-Back-N (GBN) ARQ mechanism. Indeed, the PoP transmitted by the sink contains the UID of the first lost packet in the previous polling phase. If no packets were lost, the UID of the next expected packet is inserted. The purpose of the second ACK is twofold: *i*) to avoid the retransmission of the whole block of data packets in case the first ACK is lost; *ii*) to handle a scenario in which the AUV goes out of range when it is transmitting the data to the sink, allowing to retrieve the UID of the last correctly received packet as soon as the AUV comes back in range with the sink and correctly receives a PrP from the sink itself.

When all the nodes in the POLL list are polled by the AUV, the polling phase ends and the discovery phase starts again.

2.2.1.1 Timeout Setting

As described in Section 2.2.1, the behavior of the UW-POLLING protocol depends on the timeout settings. In this section we describe more in detail how these timeouts are computed.

T_{probe} used by the AUV has to be properly set to a value such that all the PrPs can reach the AUV before the timeout expires. In particular, the timeout has to be set to a value $T_{probe} \geq T_{b_{Max}} + RTT_{Max}$, where RTT_{Max} is the maximum RTT of a node in the network.

Another timeout employed in the protocol is T_{poll}. Each node updates its own timeout every time a PoP is received. The value of T_{poll} is inserted in the PoP and is computed each time a new PoP is transmitted. The value of the timeout is computed as:

$$T_{poll} = \sum_{i=1}^{N_{list}} \left(Pks_i \cdot T_{data} + RTT_i + T_g \right) + N_{pkts}^{SINK} \cdot T_{data} + T_g \ , \tag{2.2}$$

where N_{list} is the number of remaining sensor nodes in the POLL list (the sink, if any, is not considered), T_{data} is the time needed to transmit a DATA packet, RTT_i is the RTT between node i and the AUV measured in the discovery phase, and T_g is a guard time, used to take into account the processing delay and errors in the RTT measurement. The last term of the equation takes into account the time the AUV should employ to transmit N_{pkts}^{SINK} packets to the sink. The value of T_{poll} inserted in the PoP is also used by the sink to update T_{data}^{SINK}.

The timeout T_{data}^{SINK} is also updated when the first data packet is received by the sink. The timeout is updated based on the number of data packets the sink is going to receive by the AUV. Since each data packet contains the UID of the last transmitted packet in that polling phase, the sink can easily compute the expected number of packets it will receive. The time needed to receive the data is $T_{rx} = N_{pkts}^{SINK} \cdot T_{data}$ and the timeout is set to:

$$T_{data}^{SINK} = T_{rx} + 0.5 \cdot T_{rx} \ . \tag{2.3}$$

Finally, $T_{data,i}^{AUV}$ is the timeout used by the AUV to stop waiting for the reception of

the data packets from a polled node i, and is computed as:

$$T^{AUV}_{data,i} = Pks_i \cdot T_{data} + RTT_i + T_g \ . \tag{2.4}$$

2.2.1.2 Choice of the Maximum Backoff Time

For each node density λ, defined as the number of nodes deployed in 1 km^2, an optimal value for the maximum backoff time ($T^*_{b_{Max}}$) that maximizes the throughput exists (see Section 2.3 and 2.4). If the AUV is not aware of the network node density, or if the density is not constant in all the network area, the AUV has to employ some algorithm to estimate the value of λ, and, therefore, adapt the best maximum backoff time to use time by time. In this section, we present a reactive system based on the estimated number of neighbors NN. This estimate is performed by the AUV at the end of the discovery phase, by observing the number of probe packets received. Two separate cases are analyzed: the case where we assume full knowledge (FC) information about the number of packets arrived at the physical layer, and the realistic case (RC), where only the packets received by the MAC layer are considered.

1. FC assumes full knowledge of the number of PrPs that have been correctly received in the k-th discovery phase $PrP_{rx}(T_k)$, and the number of packets discarded (LsP) by the physical layer at any time during the simulation. While the former is a parameter well known by the MAC layer, the latter can rarely be determined with high accuracy by a real modem: for this reason the FC algorithm is used as an upper-bound benchmark. With FC, the estimated number of neighbors during the k-th discovery phase T_k is calculated as

$$NN_{FC}(T_k) = PrP_{rx}(T_k) + LsP(T_{k,e}) - LsP(T_{k,b}) \ , \tag{2.5}$$

   where $LsP(T_{k,b})$ is the total number of lost packets at the beginning of the k-th discovery phase, and $LsP(T_{k,e})$ the total number of lost packets at the end of the same discovery phase.

2. RC assumes knowledge of $PrP_{rx}(T_k)$ and $PrP_f(T_k)$, i.e., the number of probe packets that have been correctly received and discarded by the MAC layer in the k-th discovery phase, respectively. In this case, both parameters are known by the MAC layer of realistic modems, therefore we are not introducing any assumptions that may favor our solution [1]. The estimated number of neighbors during the k-th discovery phase T_k is calculated as

$$NN_{RC}(T_k) = PrP_{rx}(T_k) + 2PrP_f(T_k) \ . \tag{2.6}$$

A packet is discarded by the MAC layer due to the failure of the CRC checksum, either because of low SNR, or because of interference. A packet is not received by the

[1] This is particularly true for both the AHOI and the EvoLogics modems, although the latter has to be employed with the extended notification activated.

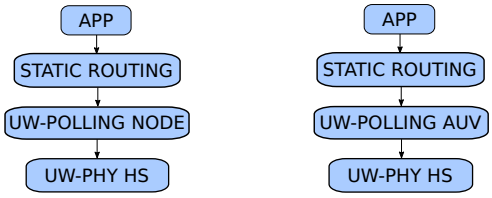


Figure 2.4: Protocol stacks used by the sensor nodes (a) and the AUV (b), respectively, during the single mode scenario with the EvoLogics S2C HS modem.

physical layer (and, consequently, not even passed to the MAC), instead, if its signal strength is below the sensitivity threshold of the transducer, due to a failure during the per-packet synchronization caused by a low SNR or interference, or if the packet reaches the destination when the physical layer is already receiving another packet. The last case is true only if the modem does not employ any interference cancellation mechanism [160] and preamble re-synchronization, as in the case of the AHOI modem. In order to perform a fair comparison, in this chapter we assume that also the EvoLogics HS modem does not employ any interference cancellation system, as we do not have any information about it.

As presented in Equation 2.6, NN_{RC} is calculated by accounting two times the number of packets discarded by the MAC layer: this is because during the discovery phase, where many nodes compete to access the channel, there is a high probability that if a packet P_k is lost at the MAC layer this is due to the interference with another packet P_i arrived after the beginning of the reception of P_k. In this case, both P_i and P_k are discarded, but only P_k is detected. In the results we will analyze when this consideration is true, and how it affects the system performance.

2.3 UW-POLLING With High Data Rate Acoustic Modem

In this section, we evaluate the protocol in the case where both AUV and sensor nodes are equipped with EvoLogics S2C HS acoustic modems. In this evaluation, the AUV acts as the sink itself. The performance of the protocol is evaluated by varying the node density λ, defined as the number of nodes deployed in 1 km^2. The settings used for our simulations are presented in Section 2.3.1, while Section 2.3.2 reports the simulation results.

2.3.1 Simulation Scenarios Description

The protocol stack implemented in the DESERT Underwater Network Simulator [149] for the sensor nodes and the AUV is depicted in Figures 2.4a and 2.4b, respectively. The physical layer employed in these simulations is the default physical layer of DESERT,

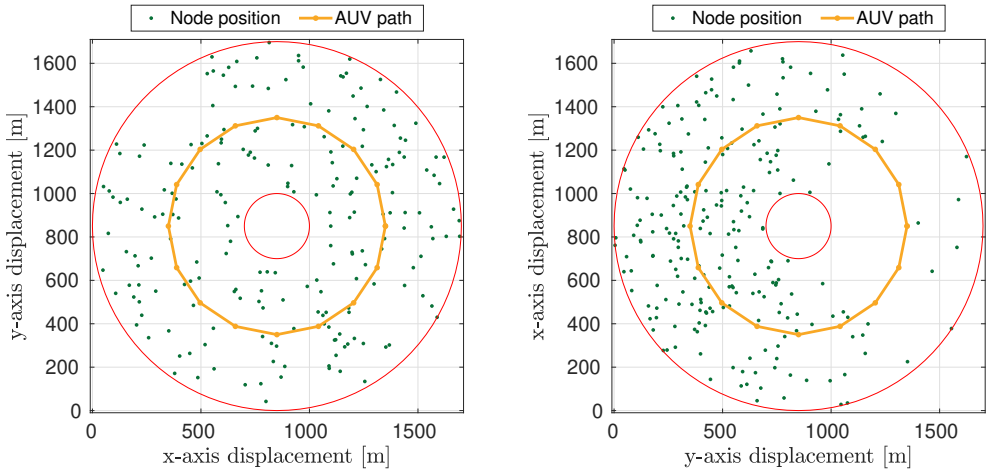


Figure 2.5: Examples of scenarios with the high speed modem: node deployment with a fixed node density λ= 100 nodes/km^2 (a), and node deployment with variable λ ranging from 10 to 200 nodes/km^2 (b).

which employs the propagation model presented in [5]. In order to simulate the behavior of the EvoLogics S2C HS modem [6], the center frequency has been set to 150 kHz, the bandwidth to 60 kHz, the transmission power to 156 dB re 1μPa@1m, and the bitrate to 7 kbit/s. The payload packet length has been set to $L_{S2C} = 125$ Byte (plus additional 8 Byte needed for the headers of the protocol stack presented in Figure 2.4). With this configuration, and considering a shipping factor of 1, wind speed of 5 m/s, and a geometrical spreading factor of 1.75, the maximum coverage range of the modem in our simulations is 490 m. This range is actually higher than the nominal range of this EvoLogics modem (that is 350 m in shallow water), because in this first simulation no multipath has been considered. A more realistic model has been considered in Section 2.5.

With this network configuration, two different scenarios are considered. In both scenarios, the AUV moves in a circular path of diameter D=1000 m at a fixed speed of 2 m/s. In the first scenario, depicted in Figure 2.5a, the nodes are uniformly distributed in a 2D space according to a homogeneous Poisson point process (PPP) with density λ. In this case, the overall throughput of the network is analyzed by varying λ and $T_{b_{Max}}$, in order to find the value of $T_{b_{Max}}$ that maximizes the throughput for each value of λ. These values are mapped in a Look-Up-Table (LUT) and used as an input for the simulations related to the second scenario (Figure 2.5b), where the nodes deployed along the AUV path have a variable density. In this case the value of λ ranges from 10 to 200 nodes/km^2. Indeed, in the second scenario the behavior of the adaptive backoff algorithm is analyzed, comparing the two approaches (FC and RC) described in Section 2.2.1.2 with the fixed backoff case.

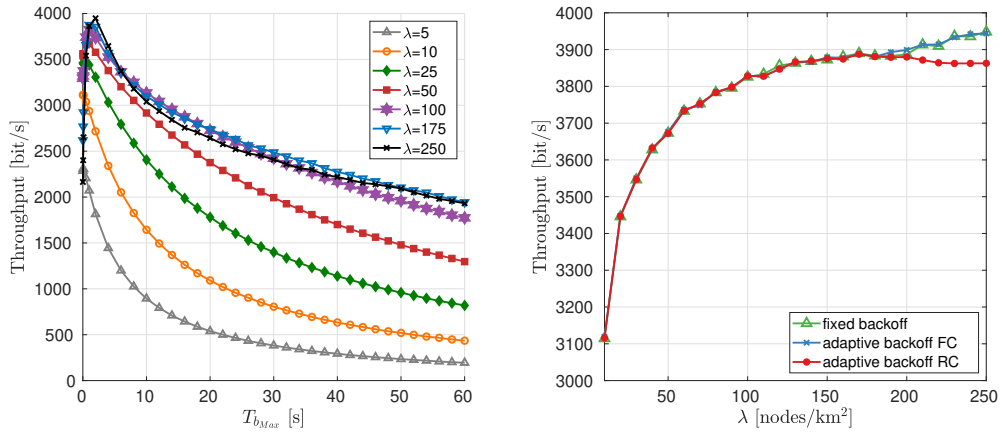


Figure 2.6: Simulation results in the first scenario with high speed modem: overall throughput as a function of the maximum backoff time for different values of λ (a), throughput as a function of λ comparing adaptive backoff approaches and fixed backoff case (b).

2.3.2 Results

In this section we report the results obtained with the high speed modem in the two scenarios described above. First of all, we analyzed the UW-POLLING protocol considering a fixed maximum backoff time and then we compared the different approaches employed to estimate the number of neighbors in the case of adaptive backoff time. All the results have been obtained averaging over 20 independent simulation runs.

In the first scenario with a fixed value of λ throughout a simulation run (Figure 2.5a), we analyzed how $T_{b_{Max}}$ impacts the network performance. Figure 2.6a depicts the overall throughput of the network as a function of the maximum backoff time for different values of λ, from 5 to 250 nodes/km^2. The throughput is computed as:

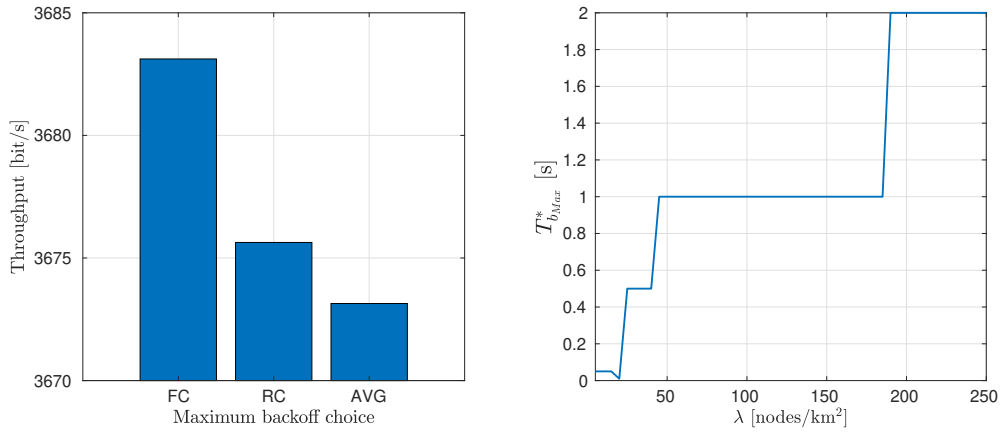
$$Thr = \frac{N_{rx}^{AUV} \cdot L_{S2C}}{T_{sim}} , \tag{2.7}$$

where N_{rx}^{AUV} is the overall number of packets received by the AUV during the simulation and T_{sim} is the duration of the simulation. With a value of $\lambda < 25$ nodes/km^2 the overall throughput decreases as the maximum backoff time increases. With these values of node density, the increase of the maximum backoff time leads to an increase of the time the AUV waits for the reception of the PrPs, without significantly decreasing the collision probability, and therefore without increasing the number of PrPs correctly received. Conversely, for $\lambda \geq 25$ nodes/km^2 the overall throughput is a trade-off between the time the AUV waits for the reception of the PrP and the number of PrPs correctly received. With $\lambda = 25$ nodes/km^2, the maximum throughput value is equal to 3476 bit/s and is obtained with $T_{b_{Max}} = 0.5$ s. With $\lambda = 50, 100, 175$ nodes/km^2 the maximum throughput (equal to 3673, 3826, 3876 bit/s, respectively) is reached with $T_{b_{Max}} = 1$ s.

Considering a density $\lambda = 250$ nodes/km^2, the overall maximum throughput is equal to 3948 bit/s and is obtained with $T_{b_{Max}} = 2$ s. We want to highlight that the optimal maximum backoff time is always smaller than 2 s in this scenario, where the high speed modem is employed, because the propagation time plays an important role in avoiding collisions. Indeed, the time needed to transmit a PrP is equal to 16 ms and in this amount of time acoustic waves cover a distance equal to 24 m, therefore 2 nodes whose distance from the receiver differs more than 24 m can transmit simultaneously without colliding at the receiver.

As a second step, we compared the maximum throughput obtained for each value of λ employing a fixed maximum backoff time with the throughput obtained choosing the maximum backoff time based on the estimate of the number of neighbors, considering the two approaches described in Section 2.2.1.2. As before, in this scenario we considered a network deployment with a fixed node density throughout each simulation run. The results are depicted in Figure 2.6b. The green line has been obtained taking the maximum value of the throughput for each λ from the previous results where a fixed $T_{b_{Max}}$ was used. The blue line has been obtained considering the full knowledge (FC) approach in the estimate of the number of neighbors. This approach is used as a benchmark for the realistic case (RC) approach (red line), where only the packets discarded by the MAC layer are known. As we can observe, the results in the 3 cases are very similar with a node density $\lambda \leq 200$ nodes/km^2. With higher values of λ, the RC line starts to diverge from the other two cases. This is due to the error in the estimate of the number of neighbors in the realistic case. Indeed, in our algorithm we supposed that a packet is discarded at the MAC layer if a collision with another packet occurs. However, with such values of λ this assumption is no longer realistic, because there is a higher probability that a collision occurs between more than two packets.

As a last step, we analyzed the protocol in the second scenario (Figure 2.5b) where the value of λ changes in space. In this scenario we compared the throughput in 3 different cases: using a variable $T_{b_{Max}}$ with the FC approach for the network density estimate, using a variable $T_{b_{Max}}$ with the RC approach, and using a constant value for the maximum backoff time lasting for the entire simulation run (AVG). In this last case, the value of $T_{b_{Max}}$ is set as the optimal maximum backoff time for the average density of the network. The results, in terms of the overall throughput of the network, are depicted in Figure 2.7a. The difference between the full knowledge case (FC) and the AVG case in this scenario is only 10 bit/s. Indeed, the $T_{b_{Max}}$ for the average case is equal to 1 s, that is a value suitable for the majority of the network densities analyzed in the previous scenario, i.e., for all the λ between 45 and 175 nodes/km^2. Figure 2.7b reports the optimal backoff time $T^*_{b_{Max}}$ obtained via simulation as a function of the network density λ. This means that the adaptation of the maximum backoff time with the high speed modem is effective only with very low or very high network density. Moreover, as mentioned before, the RC approach is not able to properly estimate the number of nodes in a high density scenario, therefore, the results of the RC are closer to the AVG case rather than the to FC approach.



(a) **(b)**

Figure 2.7: Throughput in the variable density scenario with the adaptive backoff approaches (FC and RC) and the fixed backoff case (AVG) (a). Optimal backoff time as a function of the network density obtained via simulation considering the first scenario with fixed node density (b).

2.4 UW-POLLING With AHOI Modem

In this section, we evaluate the protocol in the case where both AUV and sensor nodes are equipped with AHOI acoustic modems. Similarly to Section 2.3.2, the AUV acts as the sink itself, and the network performance is evaluated by varying the node density λ. Section 2.4.1 briefly describes the AHOI modem, the settings used for our simulations are presented in Section 2.4.2, while Section 2.4.3 reports the simulation results.

2.4.1 AHOI Modem

The AHOI modem is a small, low-power and low-cost acoustic underwater modem, developed to be integrated into micro AUVs or USNs. The modem consists of three stacked Printed Circuit Boards (PCB) with an overall size of $50 \times 50 \times 25$ mm^3 and approximately €200 component cost. The first PCB includes a CortexM4 microcontroller, power supply and external connections. The second PCB works as the receiver and involves amplifiers, a bandpass filter and an analog-to-digital converter. The bandpass filter is formed by a highpass filter with cut-off frequency $f_{c,h} = 50$ kHz and a lowpass filter with $f_{c,l} = 75$ kHz. The AHOI modem supports different frequency bands, which can be adapted by the user (tuning the bandwidth of the analog bandpass filter). The bandwidth between 50 and 75 kHz is the default one. The transducer used for signal reception and transmission is the Aquarian Audio AS1 hydrophone [161] at the cost of €400. To counter time-dependent attenuation caused by multipath propagation, each symbol is repeated three times. In addition, frequency hopping (FHSS) is applied using a hopping scheme, which is adapted to the symbol repetitions. The modem has 25 kHz

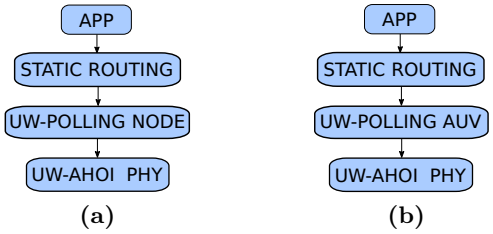


Figure 2.8: Protocol stacks used by the sensor nodes (a) and the AUV (b), respectively, during the single mode scenario with the AHOI modem.

bandwidth around a central frequency of 62.5 kHz. The default setup, in combination with an extended Hamming code, leads to a net data rate of 260 bit/s. However, the actual firmware allows a net data rate up to 4.7 kbit/s using six bits per symbol, 1.28 ms symbol duration, and without repetitions and Hamming coding [162]. The experienced transmission range can reach up to 150/200 m [11].

2.4.2 Simulation Scenarios Description

The protocol stack implemented in the DESERT Underwater Network Simulator [149] for the sensor nodes and the AUV is depicted in Figures 2.8a and 2.8b, respectively. The physical layer employed in these simulations is the AHOI physical layer, designed to mimic the modem performance based on real-field data. In this case, the central frequency has been set to 62.5 kHz, the bandwidth to 25 kHz, the transmission power to 156 dB re 1μPa@1m, and the bitrate to 200 bit/s. The maximum coverage range of the AHOI modem is roughly 150 m. The payload packet length has been set to $L_{AHOI} = 24$ Byte (plus additional 8 Byte needed for the headers of the protocol stack presented in Figure 2.8).

With this network configuration, two different scenarios are considered. In both scenarios, the AUV moves in a circular path of diameter $D = 520$ m at a fixed speed of 2 m/s. In the first scenario, depicted in Figure 2.9a, the nodes are uniformly distributed in a 2D space according to a homogeneous PPP, with a node density of λ nodes per square kilometer. Also in this case, the overall throughput of the network is analyzed by varying λ and $T_{b_{Max}}$, in order to find the value of $T_{b_{Max}}$ that maximizes the throughput for each value of λ. These values are mapped in a LUT and used as an input for the simulations related to the second scenario (Figure 2.9b), where the number of nodes deployed along the AUV path has a variable density. In this case the value of λ ranges from 50 to 400 nodes/km^2. Finally, in the second scenario the behavior of the adaptive backoff algorithm is analyzed comparing the FC and RC approaches (described in Section 2.2.1.2) with the fixed backoff case.

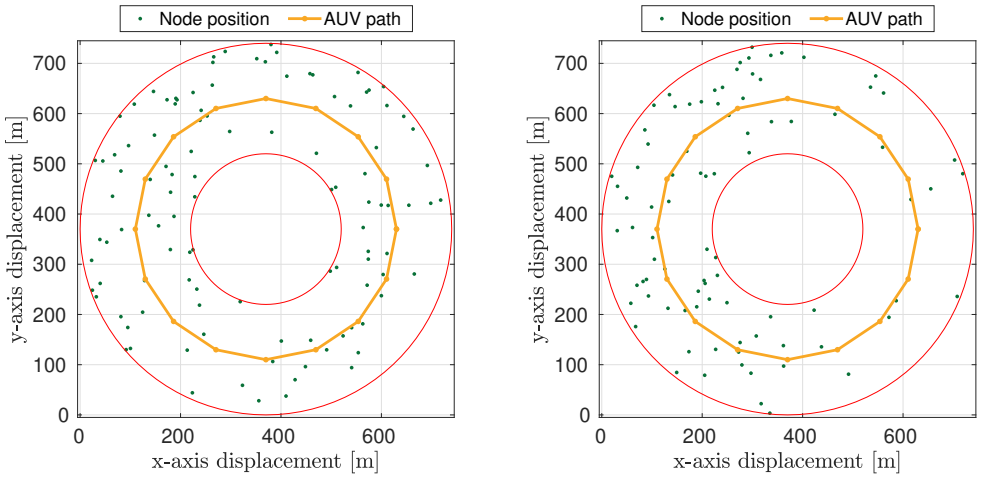


Figure 2.9: Examples of the two scenarios considered in the low rate modem simulations: node deployment with a fixed node density λ= 300 nodes/km^2 (a), and node deployment with variable λ ranging from 50 to 400 nodes/km^2 (b).

2.4.3 Results

In this section we present the performance of the UW-POLLING protocol in the first and second scenarios, when the low rate modem is employed. As for the high speed modem, we first analyzed the overall throughput of the network considering a fixed network density and a constant backoff time throughout a simulation run, and then compared the different approaches to estimate the number of neighbors with both a fixed and a variable node density.

Figure 2.10a reports the overall throughput of the network as a function of the maximum backoff time for different values of λ. The throughput is computed as:

$$Thr = \frac{N_{rx}^{AUV} \cdot L_{AHOI}}{T_{sim}} , \tag{2.8}$$

where N_{rx}^{AUV} is the overall number of packets received by the AUV during the simulation and T_{sim} is the duration of the simulation. In the low rate modem scenario the effectiveness of the maximum backoff time is more significant with respect to the high speed modem scenario. In this case the time needed to transmit a PROBE is equal to 0.56 s, that is bigger than the maximum propagation time experienced in this scenario (i.e., 0.1 s). This means that the distance between nodes is no longer sufficient to avoid collision unlike in the high speed scenario presented in Section 2.3. In this scenario, the maximum throughput is a trade-off between the number of PROBE packets correctly received and the time spent by the AUV waiting for the reception of the PROBE. With $\lambda = 25$ the maximum throughput is obtained with $T_{b_{Max}} = 2$ s. The optimal $T_{b_{Max}}$,

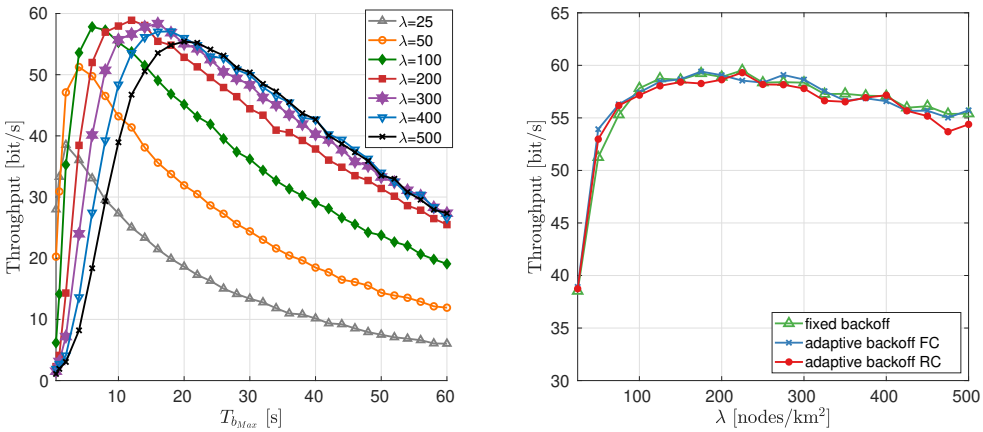


Figure 2.10: Simulation results in the first scenario with low rate modem: overall throughput as a function of the maximum backoff time for different values of λ (a), throughput as a function of λ comparing adaptive backoff approaches and fixed backoff case (b).

i.e., the maximum backoff time corresponding to the maximum throughput, increases as the network density increases.

In the scenario with fixed node density, we also compared the results obtained with the adaptive mechanisms for the choice of the maximum backoff time with the fixed $T_{b_{Max}}$. In the adaptive choice of the backoff we simulated both the FC approach, used as the upper-bound benchmark, and the RC approach. Figure 2.10b depicts the results, in terms of throughput, in the three cases: fixed backoff (green line), FC approach (blue line) and RC approach (red line). The choice of the fixed maximum backoff has been obtained from the previous simulations with fixed density and fixed $T_{b_{Max}}$, selecting the maximum throughput for each λ. We can observe that the results in the three cases are almost equivalent for all values of λ. This means that the node estimate in the RC approach is accurate enough to not degrade significantly the performance with respect to the FC approach, and with respect to the case where λ is known (fixed backoff case).

As a last step we analyzed the protocol performance in the scenario with a variable node density (Figure 2.9b). Also in this scenario we compared the throughput in three different cases: using a variable $T_{b_{Max}}$ with the FC approach for the network density estimate, using a variable $T_{b_{Max}}$ with the RC approach, and using a constant value for the maximum backoff time lasting for the entire simulation run (AVG). In the AVG case, the value of $T_{b_{Max}}$ is set as the optimal maximum backoff time for the average node density of the network. The results, in terms of the overall throughput of the network, are depicted in Figure 2.11. In this scenario, where the low rate modem is employed, the throughput of the RC approach is very close to the benchmark obtained with the FC approach. Moreover, the improvement in the adaptive backoff case with respect to the AVG case is about 8%

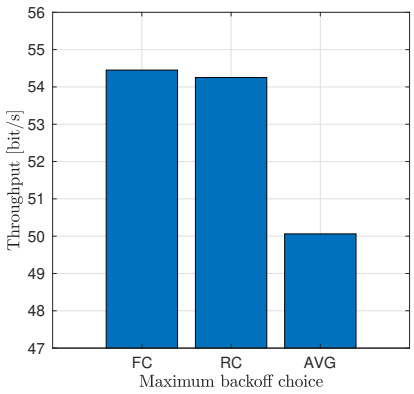


Figure 2.11: Throughput in the variable density scenario with the adaptive backoff approaches (FC and RC) and the fixed backoff case (AVG).

2.5 Multimodal Solution in the Hamburg Port Scenario

In this section we analyze the UW-POLLING protocols in the scenario of the port of Hamburg. Differently from the previous scenarios, in this deployment we considered a multimodal setting where both the AHOI modem and the EvoLogics S2CM HS modems are employed. Moreover, we considered a sink node placed in a fixed position and different from the AUV. The scenario and the setting employed in these simulations are presented in Section 2.5.1. Section 2.5.2 reports the results.

2.5.1 Simulation Scenario Description

In this scenario we considered a multimodal network where both the AHOI and EvoLogics modems are employed. Figure 2.12 reports the protocol stack for all the nodes in the network. In particular, sensor nodes are equipped with the AHOI modem (Figure 2.12a), the sink node is equipped with the high speed modem (Figure 2.12c), and the AUV is equipped with both modems (Figure 2.12b). The AUV protocol stack includes the UW-MULTI-DESTINATION layer (described in Section 2.2) for selecting which modem to employ at each packet transmission. Specifically, in the AUV the AHOI modem is used to collect the data from the sensor nodes deployed along the AUV path, and the Evologics HS modem to forward the data to the sink, if in range. The AHOI modem has been simulated as described in Section 2.4, while for the EvoLogics HS modem, the acoustic propagation has been simulated with the Bellhop ray tracer, importing the Hamburg port bathymetry (Figure 2.13b) in the WOSS Framework [150]. Two instances of the UW-POLLING protocol have been used in the AUV protocol stack: one related to the AHOI modem and the other to the high speed modem.

In this scenario, we deployed the nodes as in Figure 2.13a and we let the AUV move at 2 m/s, performing 10 laps of the orange path. We considered 63 sensor nodes deployed in an area of 0.6 km^2 (average node density $\lambda = 105$ nodes/km^2), creating

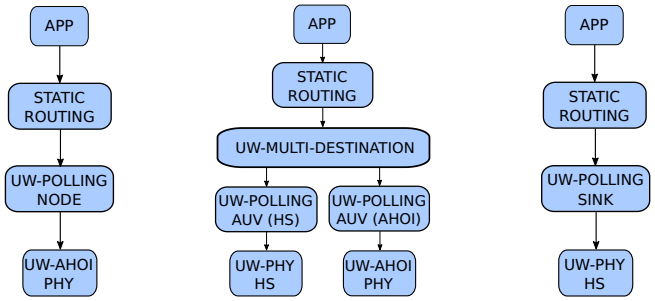


Figure 2.12: Protocol stack used by the sensor nodes (a), the AUV (b) and the sink (c), during the complete multimodal scenario, where the AUV delivers the collected data to the sink.

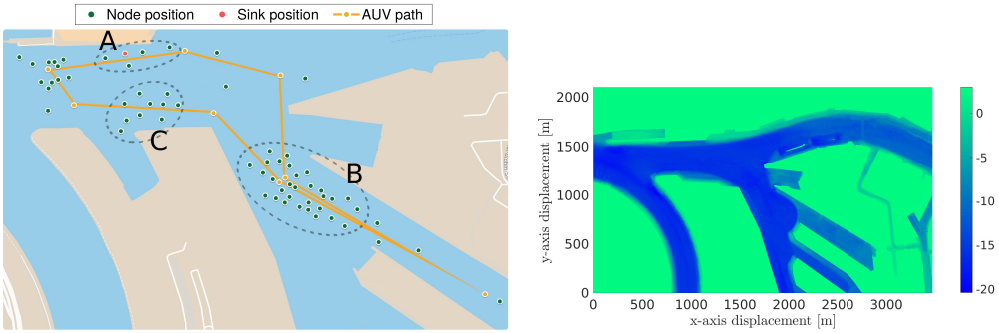


Figure 2.13: Node deployment along the Elbe river in the port of Hamburg (a). Three clusters of nodes are identified with the letters A, B and C. (b) represents the port of Hamburg bathymetry related to the zone depicted in (a).

high density areas interspersed with low density areas. The sink (red node) is placed in a fixed position.

The UW-POLLING protocol related to the AHOI modem has been analyzed both with the adaptive backoff mechanisms (FC and RC) and in the fixed backoff case. In the fixed backoff case we use as $T_{b_{Max}}$ the optimal value for the average node density obtained with simulations in the scenario described in Section 2.4.2. The optimal value with $\lambda = 105$ nodes/km^2 is equal to $T_{b_{Max}} = 6$ s. The instance of UW-POLLING related to the EvoLogics modem has been analyzed considering only a fixed $T_{b_{Max}} = 20$ s and with $Pr_{Max} = 1$.

2.5.2 Results

In this section we present the results obtained in the scenario of the port of Hamburg described above.

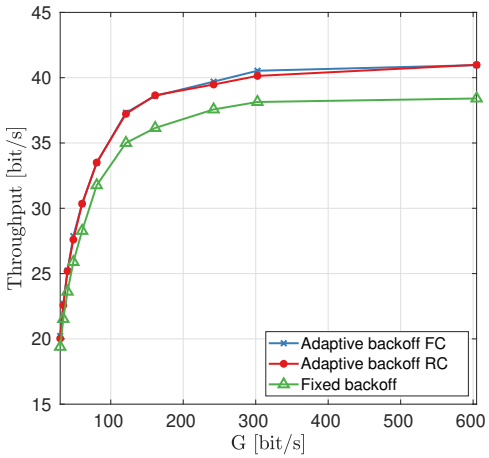


Figure 2.14: Overall throughput in the port of Hamburg scenario as a function of the offered traffic.

Figure 2.14 reports the overall throughput of the network as a function of the overall offered traffic of the network G. The throughput is computed as:

$$Thr = \frac{N_{rx}^{SINK} \cdot L_{AHOI}}{T_{sim}} , \tag{2.9}$$

where N_{rx}^{SINK} is the overall number of packets received by the SINK during the simulation and T_{sim} is the duration of the simulation. We assessed the performance of UW-POLLING considering the adaptive backoff mechanisms (FC and RC) and the fixed backoff case. In the adaptive backoff cases, the FC approach and the RC approach, used for the neighbors estimate, are almost equivalent. As mentioned in Section 2.4.3, with the AHOI modem the neighbors estimate performed with the RC approach is accurate enough to not degrade the performance in terms of throughput. The throughput increases as the offered traffic increases up to $G = 300$ bit/s. Considering $G \geq 300$ bit/s, the overall throughput of the network remains almost constant at 40 bit/s. For low values of G the difference between offered traffic and throughput is mainly due to packet losses due to bad channel conditions. Increasing G, the maximum achievable throughput is also limited by the bitrate of the modem and the presence of the discovery phase: for this reason, some of the nodes are not able to transmit all the generated packets to the AUV. In particular, in the area with high density nodes, such as cluster B, most of the packets remain in the node queues. In the low density areas this fact is less marked but still present. Thanks to multimodality, the transmission of the packets from the AUV to the sink is no longer a bottleneck, unlike what happened in [163] where only the AHOI modem was employed. Moreover, also the fairness of the nodes is enhanced with respect to the results in [164]. In that paper, nodes close to the sink were not able to transmit their data packets, because in their proximity the AUV was busy forwarding the data to the sink. With a multimodal approach, instead, we are able to both collect

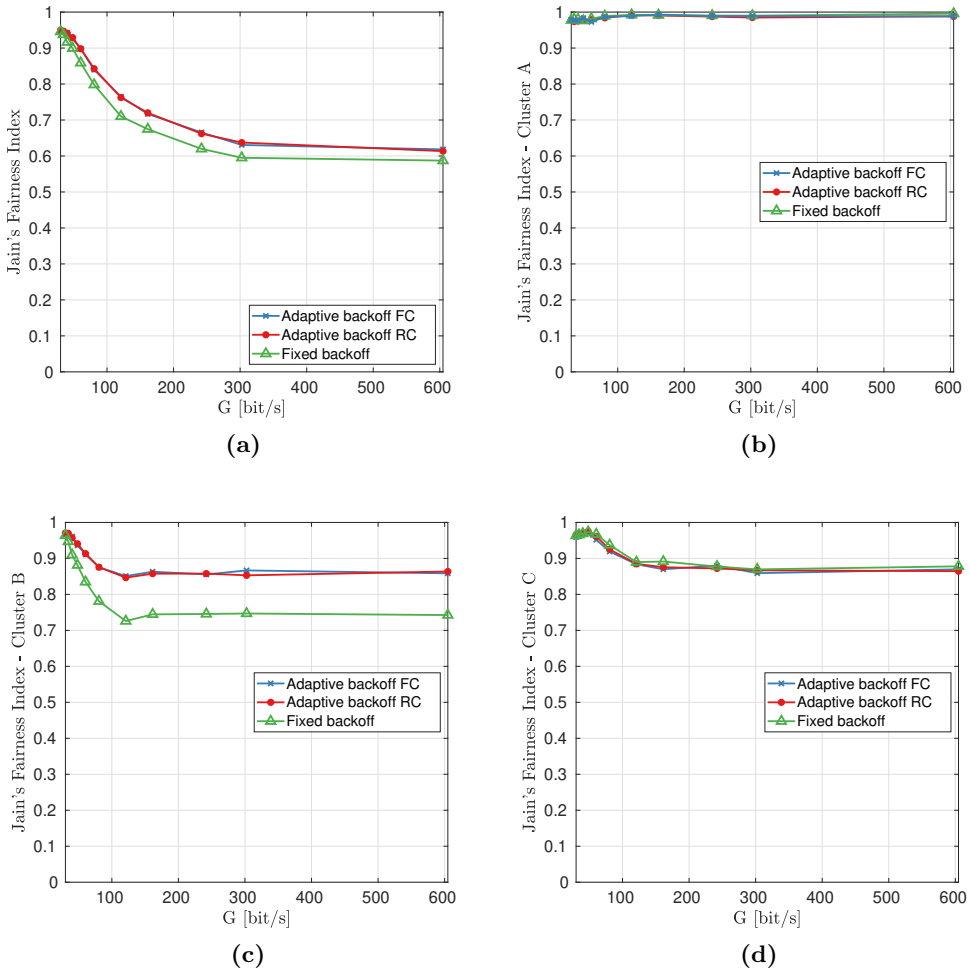


Figure 2.15: Jain's Fairness Index for the whole network (a), cluster A (b), cluster B (c) and cluster C (d).

the data from sensor nodes and forward packets to the sink at the same time. In the fixed backoff case, the behavior is the same as for the adaptive mechanisms, but the throughput is much lower than in the former cases: in particular, using an adaptive mechanism we can improve the performance up to 9%.

Figure 2.15 reports Jain's Fairness Indexes (JFIs) [165] as a function of the offered traffic, computed for both the overall network and the clusters of nodes depicted in Figure 2.13a. Jain's Fairness index is computed as

$$JFI = \frac{\left(\sum_{i=1}^{N_{nodes}} P_{rx,i}\right)^2}{N_{nodes} \sum_{i=1}^{N_{nodes}} P_{rx,i}^2}, \tag{2.10}$$

where N_{nodes} is the number of nodes in the network (or in the considered cluster) and $P_{rx,i}$ is the number of DATA packets received by the AUV from node i. Figure 2.15a reports JFI for the whole network for both the adaptive backoff cases and the fixed backoff mechanism. In the adaptive cases JFI is slightly higher than in the fixed case, however in all cases, as the offered traffic increases JFI decreases, down to 0.6. Since the node density is not constant in space, we have to highlight that JFI of the whole network cannot be close to 1, especially for the largest values of offered traffic. Indeed the number of packets transmitted by a node in a high density area is smaller than the packets transmitted by a node in a low density area. For this reason we computed JFI separately for nodes in each of the clusters presented in Figure 2.13a. Figure 2.15b depicts JFI for the nodes in cluster A. Cluster A is representative of a low density scenario and JFI is close to 1 for both the adaptive backoff mechanisms and the fixed backoff case. Cluster B is a high density area and JFI in this area is depicted in Figure 2.15d. In this area JFI with both the FC and RC approaches is equal to 0.85 for $G \geq 100$ bit/s. As G decreases, JFI further increases up to 1. Therefore, also in high density scenarios we are able to obtain a good level of fairness. In the fixed backoff case, with $G \geq 100$ bit/s JFI decreases down to 0.75. The difference between the adaptive cases and the fixed backoff case is because the $T_{b_{Max}}$ used in the fixed case is not suitable for this area. Indeed, the node density in this area is about 235 nodes/km^2, and the maximum backoff time is too small, causing a higher collision probability for the PrPs, and therefore giving less chance to the nodes to transmit their packets. Finally, Figure 2.15d reports JFI computed for the nodes in cluster C. In this case JFI is similar for all the three assessed cases. JFI decreases for G smaller than 150 bit/s and then remains constant to 0.86 for bigger values of the offered traffic.

2.6 Experimental Analysis of UW-POLLING

In the smart port system described at the beginning of this chapter, users, e.g., port authorities, ship owners, or other port clients, will be able to book different services and monitor their progress through a user interface, for example by a standard web browser. In such a system, the operators will then be responsible for choosing the UV available to perform the service [166]. Among the others, the data collection service has been studied through the design of the polling-based MAC protocol. In this section, the UW-POLLING protocol is evaluated through a lake test conducted at lake Kreidesee in Hemmoor (Germany). The goal of the test is to simulate the complete pipeline of a smart port, from the collection of data from sensor nodes to its forwarding and processing to the shore. In this test an ASV, namely the SeaML developed by the Fraunhofer Center for Maritime Logistics and Service, patrols the lake and collects the data from the underwater acoustic sensors deployed in the lake area. To this purpose, all the nodes are equipped with the AHOI modem developed by the Technical University of Hamburg [55], the same simulated in the previous chapters in the UW-POLLING analysis. In addition, to simulate the whole pipeline, a shore operation center is deployed close to the lake shore. The operation center is connected

to the ASV through an RF link. Specifically, the two elements communicate through a Wi-Fi link.

2.6.1 System Design and Implementation

2.6.1.1 Underwater Network

An underwater network enables the communication between ASVs and battery-powered underwater sensors, used to collect environmental data to monitor a certain area. In this test, the underwater communication is set up with the AHOI low-cost and low-power acoustic modems [55], and a complete communication stack implemented using the DESERT Underwater Framework [149], which implements a complete ISO/OSI modular stack for underwater communications on top of the Network Simulator 2 (ns2). An important feature of this framework is the *RealTime scheduler*, which overwrites the existing ns2 scheduler and synchronizes the program with the machine internal processor, so that the simulator can operate in real time with other programs or external devices while using the same protocols implemented for simulations. The network protocol used to collect the data from the underwater sensor nodes is the UW-POLLING described in this chapter.

Concerning the transmission of packets through the medium, a physical module was implemented to act as driver with the AHOI modems: more details on the implementation can be found in [167].

2.6.1.2 Above Water Network

The above water section of the network has been designed and implemented using equipment manufactured by Mikrotik [168], which provides a vast variety of reliable and cost-effective network devices. All Mikrotik devices provide an easy-to-use, yet powerful configuration interface, able to exploit any ISO/OSI Layer 2 and Layer 3 technologies, such as Ethernet Bridging, 802.1Q VLANs, static and dynamic routing and DHCP. On WiFi-capable devices, WiFi parameters such as SSID, ACLs, WPA2 and some physical layer parameters can be easily configured [169]. Every device can be easily configured to act as an Access Point or a Client, which connects to an existing WiFi network. Moreover, devices with multiple antennas can be independently configured to act as an Access Point and a Client, thus realizing a range extender. Hereafter, we give a brief description of the topology of the above water network deployed. On the ASV, a Mikrotik Metal 52AC WiFi CPE [170] has been adopted. The Metal 52AC device is a full-fledged router, mounting a Gigabit Ethernet port, a WiFi 802.11b/g/n and 802.11ac compliant WiFi radio (configured to act as a client) and a 6 dB `i` omnidirectional antenna. The Metal device has been connected with a Gigabit Ethernet cable to the ASV's hardware used to forward the data acquired from the underwater network to the above water network. On the piers, we adopted mANTBox 2 12s [171] as the WiFi AP, which mounts a dB `i` 120° directional antenna and an 802.11b/g/n WiFi modem, able to cover long distances and challenging radio channels. We adopted a 5 Ethernet ports Mikrotik hEX [172] as the core router, which connects the mANTBox,

Table 2.2: DATA_SENS string format

Parameter	ID	TIMESTAMP	DATATYPE	VALUE
Format	`XX`	`HHmmss`	`<type initial>`	`<value>`
Size [bytes]	2	6	1	4

the RabbitMQ servers (placed at the shore operation center) and optionally laptops for troubleshooting and monitoring, thus allowing the end-to-end connectivity between the Tinkerboard, i.e, the Single-Board Computer (SBC) used in the buoys and ASV, and the servers.

2.6.1.3 Data Compression and Live Data Generation

In addition to DESERT, other two programs have been implemented for the underwater telecommunication pipeline, namely:

- DATA_SENS, an application that acquires data from either a real or a mocked sensor, formats it to a compressed string, and sends it through the DESERT application layer;
- NET_BRIDGE, the application used in the ASV to forward the collected sensor data from the underwater network to the shore operation center.

DATA_SENS interfaces with the DESERT Framework at the application level through a module that generates a socket connection, aiming to communicate with an external program that generates the data to be transmitted following the protocols in DESERT. The main reason DATA_SENS sends the data through the acoustic channel with a formatted and compressed string with a fixed size rather than standard JSON formatted strings, is due to the limitation of the acoustic channel, that allows only the transmission of small data packets. DATA_SENS, therefore, creates a string for each sensor measurement formatted as presented in Table 2.2. The first 2 Bytes of the string compose the ID of the sensor node that generated the data, the following 6 Bytes the timestamp at which the data was generated, formatted in hours minutes seconds, 1 Byte is used to represent the datatype (temperature, salinity, etc.) acquired by the sensor, and, finally, the remaining bytes are used to represent the data value. For instance, in the case of the temperature data type, 4 Bytes are used for the value: with this data compression, the underwater nodes transmit temperature data with a packet of 25 Bytes, including the header introduced by DESERT. For other data types, the length depends on the number of bytes used to represent the value.

NET_BRIDGE, instead, parses the sensor packets received by the ASV, and converts them into a JSON file, that is then sent to the cloud by using an Advanced Message Queuing Protocol (AMQP) client that transmits the JSON file by using the above water network. For example, the string sent by node 16 through the underwater network at time 14:23:45, with temperature data with value 18.5 $°C$, would be `16142345T18.5`,

which is translated on the other end in this JSON file, where the date is either the current date or the day before, as we assume that the data has been generated up to 24 hours before the current time:

```
{
   "buoy_id" :  "16",
   "data_type" :  "temperature",
   "recorded_at" :  "2021-03-02T14:23:45Z",
   "value" :  18.5
}
```

2.6.2 Lake Test Settings

Field tests were performed at lake Kreidesee in Hemmoor, Lower-Saxony, Germany. The onshore equipment included a mobile station acting as a server, supporting the main above water communication brokers and saving the information into the database. For the off-shore counterparts, which were previously either simulated or run in confined environments, besides the SeaML ASV, a series of buoys with ballast to avoid drifting were manually deployed in the different topologies. The SeaML ASV carried all the necessary equipment for performing the whole pipeline of the data-muling service, i.e., the Mikrotik Metal 52AC to communicate to the shore and AHOI modem to collect the data from the sensors. Each node of the network was equipped with a SBC running the DESERT Framework simulator to test the UW-POLLING protocol.

2.6.2.1 Shore Operation Centre

On the shore, the following hardware components were used to deploy the software solution:

- the server-side application was deployed on a Dell Latitude 5411 laptop running Ubuntu 18.04;
- the underwater network was controlled and monitored from a Panasonic Toughbook CF-53, running Linux Mint 18.1 and connected via SSH to the buoys;
- above water communication network was deployed on a Mikrotik hEX PoE lite, which had a 4G modem attached for internet connectivity and Mikrotik mANTBox 2 12S Wi-Fi antenna.

2.6.2.2 Above Water Network Setup

The underwater communication and all the off-shore devices were supported by a radio network operating in the 2.4 GHz WiFi band and implemented using Mikrotik devices.

Specifically, the shore station was equipped with a Mikrotik hEX POE lite [172] main LAN hub that holds the DHCP server and thus provides network addresses and configurations to all other radio devices. This device is a router with 4 output ports supporting 100 Mbit Ethernet. To distribute the WiFi signal from shore to the middle

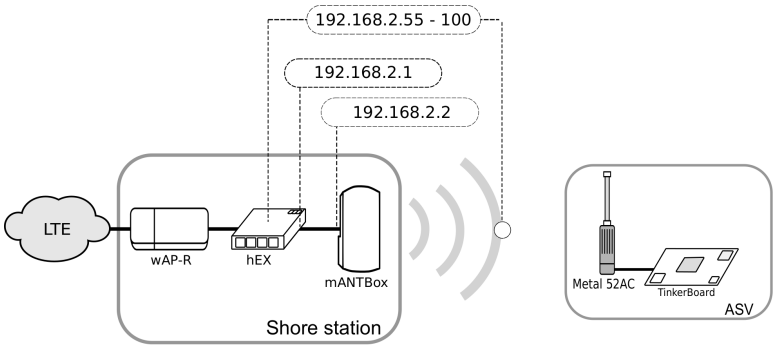


Figure 2.16: Above water network architecture. The shore station (left), exchanges data with the ASV (right) using a long range WiFi link, and acts as the gateway of the network providing all nodes Internet connectivity thought LTE.

of the lake we connected to the hEX router a Mikrotik mANTBox 2 12 which is equipped with a 12 dB i 120° aperture antenna. Also the mANTBox is a router device: to avoid collision with the hEX DHCP server, the mANTBox was set to be DHCP transparent, hence it forwarded the DHCP hEX signaling to all the devices in the lake through the long range WiFi link. Also a Mikrotik wAP-R [173] wireless access point with LTE connectivity was connected to the hEX router via Ethernet, in order to allow the entire network to access the internet via LTE: this simplified the maintenance and the management of the devices.

The antenna installed in the vehicle, instead, was the Mikrotik Metal 52AC long range omnidirectional antenna, with a gain of 6 dBi. Also this device is a router, but we mainly used it for the long range antenna that provided direct WiFi connection from the SBCs installed in the ASV to the shore station server.

The IP address pool used by the hEX DHCP server was ranging from 192.168.2.55 to 192.168.2.100, while the address of the hEX DHCP server was 192.168.2.1, and the address of the mANTBox wired interface was 192.168.2.2: the whole network connections and addressing are depicted in Figure 2.16.

2.6.2.3 Underwater Nodes

The underwater nodes for the data collection service are equipped with sensors to measure environmental data, a power supply, a processing unit, and the AHOI acoustic underwater modem. Usually the nodes are submerged in an area of interest and do not have any cabled connectivity or above water communication link. In a preliminary test performed to evaluate the underwater nodes functionality in combination with the AHOI modems we experienced a communication range between 150 m to 250 m in static scenarios and up to 150 m in mobile scenarios.

To evaluate the data collection service, we decided to use buoys instead of fully submerged nodes to simplify the deployment. A buoy consists of an ASUS TinkerBoard S used to execute the DESERT Underwater Framework, an AHOI modem, a Navilock

(a) Electronic components

(b) Buoys with hydrophones

(c) Deployed buoy and SeaML

Figure 2.17: For the evaluation and demonstration of the data collection service a prototype node based on a buoy was built.

NL-6002U GPS receiver, and a power supply. An external WiFi antenna connected to the TinkerBoard enables debugging and monitoring during the evaluation. However, WiFi connection and GPS receiver are just part of the prototype for the service demonstration with buoys, and would be removed from the actual nodes in the case of a long-term deployment. For power supply a battery (11.1 V, 2400 mA) and a power management board are used. The power management board provides a stabilized 5 V supply to the TinkerBoard, measures the battery voltage and current, and protects the components against under-voltage, reverse polarity and short circuits. Depending on the computational load of the TinkerBoard and the transmission intervals of the AHOI modem, the battery allows an operation time between 6 h and 10 h. The external hydrophone is placed about 1.1 m under the buoy to avoid the region directly under the water surface that is strongly affected by acoustic reflections. Furthermore, a cage protects the hydrophone against physical damage. Finally, a rope is connected to the protection cage to fix the buoys in the water with an anchor.

For the deployment, five buoys were constructed. Figure 2.17 depicts electronic components, the buoys before the deployment and a single buoy with the SeaML ASV. In addition, a sixth node was prepared. It uses a Raspberry Pi 4 board, connected to an AHOI modem. The electronic components were installed in a box to place the node on a jetty and submerge the hydrophone at 1 m depth.

To collect environmental data from the sensor nodes, the SeaML was equipped with an AHOI modem and a TinkerBoard as well. Similar electronic components from buoys were installed in a waterproofed case on top of the SeaML. Furthermore, a hydrophone in a protection case was mounted 0.9 m under the SeaML. In order to measure the position of the mobile and the fixed node on a jetty, a Navilock NL-8001U GPS receiver was used.

2.6.2.4 Underwater Network Settings

The underwater network is composed by one mobile node and a number of static nodes that ranges from four to six, depending on the considered network topology. In our scenario, every 60 s each static node generates a packet with a payload of 13 Byte (if we consider the packet headers introduced by the communication protocols the size increase up to 25 Byte). The transmission bitrate used by the AHOI modems is 200 bit/s.

To collect the data from the sensor nodes, the UW-POLLING MAC protocol has been used. As described at the beginning of this chapter, an important parameter of this protocol is the maximum backoff time $T_{b_{max}}$ used in the discovery phase by the sensor nodes to randomize the channel access trying to reduce the collisions. In our tests, we set $T_{b_{max}} = 15$ s. In addition, each node can transmit in each protocol cycle (i.e., every polling phase) a burst of up to five consecutive packets, in order to limit the number of packets sent by a node in each cycle and thus reduce the probability that the ASV moves out of range during a packet burst transmission.

2.6.2.5 Data-Muling Topologies

As previously mentioned, the underwater network relied on the DESERT Underwater Framework to accomplish the network protocol management. The communication is performed with acoustic signals, so that the sensors could be placed even on the seabed and still transmit wirelessly to the surface. Moreover, the data is transmitted using the DESERT Framework, whose packet header contains the required information for the routing and its packet payload is formatted to include all the needed information for the data acquisition, as explained in Section 2.6.1.3.

To better understand the behavior of the UW-POLLING MAC protocol, different network topologies have been tested to analyze how the protocol behaves under different network conditions, such as node density and distances. The performance has been assessed in terms of fairness and packet delay, and the overall underwater network has been evaluated taking into consideration the PDR of the data packets.

To perform this analysis we designed four different topologies, to reproduce four networks with different node densities. In particular, we designed the networks to obtain clusters with a different number of nodes, to test the behavior of the discovery phase of the UW-POLLING protocols. Based on the maximum range D_{MAX} of the AHOI modem (about 150 m to 250 m) and the area of operations in Figure 2.18, we depicted the following topologies:

1. *equally distanced nodes with* $D > D_{MAX}$: in this topology all the adjacent nodes have a distance greater than the transmission range, thus the ASV would be in range with no more than two sensor nodes at the same time. This topology is shown in Figure 2.18a;

2. *equally distanced nodes with* $D < D_{MAX}$: in this topology the nodes are placed close enough to obtain a single cluster with all the available nodes, such that the ASV can be in range with all the sensor nodes during the entire test. The goal is

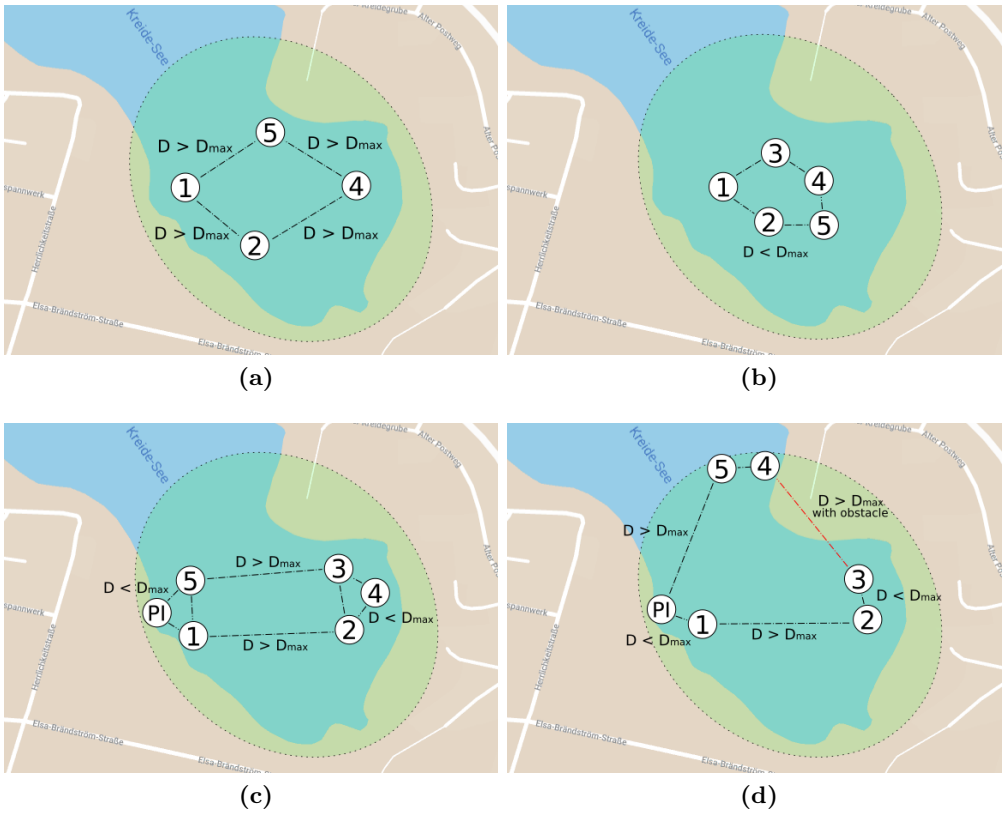


Figure 2.18: Theoretical topologies represented with the required minimum distances.

to let all the nodes participate to the same channel contention during a discovery phase. This topology is shown in Figure 2.18b;

3. *two clusters at* $D > D_{MAX}$: in this topology there are two main clusters, composed of three sensor nodes each and placed at a distance such that when the ASV is close to one cluster it is not in range with the other one. With this topology we obtain a network in which one of the clusters is not visited for a while and accumulates packets in the internal queue until the next visit, when the muling node returns in range. This topology is shown in Figure. 2.18c;

4. *three clusters at* $D > D_{MAX}$ *and physical obstacle* : in this topology there are three clusters, composed of two sensor nodes each and placed at a distance larger than the communication range; in addition to the previous case, two clusters are divided also by a physical obstacle (in the Hemmoor test it was a small headland), so that there is a certain communications shielding between the ASV and the sensor nodes in the other clusters, even while traveling to them. This topology is shown in Figure 2.18d.

Table 2.3: Metrics for Topology 1

Topology 1	Sent	Received	PDR [%]	PDD [s]
Node 1	48	38	0.791	501.46
Node 2	63	55	0.873	326.55
Node 4	65	60	0.923	438.97
Node 5	70	58	0.828	417.30

2.6.3 Lake Test Results

2.6.3.1 Underwater Network Performance

The underwater scenario is characterized by high propagation delay and Bit Error Rate (BER), combined with a low bitrate determined by the frequencies available for communications. Thus, on one hand, the overall throughput is orders of magnitude lower than the one observed in wireless terrestrial networks. On the other hand, the PDD, defined as the time elapsed from the packet generation to the correct packet reception at the destination, in underwater acoustic networks is in the order of several seconds or minutes, not of milliseconds like in terrestrial networks. These phenomena, typical of Delay Tolerant Networks (DTNs) such as satellite and underwater acoustic networks [174], are even more evident in a data-muling scenario, where a mobile node collects the data from submerged sensors when it enters their communication range, as the link between a sensor and the mobile node is systematically disrupted as soon as the mobile node moves out of the sensor's coverage area. Another important metric that shows how the network worked during the tests is Jain's *fairness* index (JFI) [165], which indicates whether or not all the nodes were treated fairly by the network protocol stack. Specifically, Jain's Fairness index is computed as

$$JFI = \frac{\left(\sum_{i=1}^{N_{nodes}} P_{rx,i}\right)^2}{N_{nodes} \sum_{i=1}^{N_{nodes}} P_{rx,i}^2}, \tag{2.11}$$

where N_{nodes} is the number of nodes in the network and $P_{rx,i}$ is the number of data packets received by the ASV from node i.

In Tables 2.3, 2.4, 2.5 and 2.6, we present the network performance for the four different topologies. As defined above, the PDD is the delay after which a packet is correctly received, and is computed starting from the packet generation time. Thus, for the topologies where the nodes are not always in the ASV range, the delay also includes the time the vehicle takes to visit all the other nodes and to come back. This can be observed for topologies 1, 3, and 4, where the average PDDs of the networks (i.e., the PDD averaged over the nodes) are 421.07 s, 241.33 s, and 336.95 s, respectively, against the average network PDD of topology 2 equal to 73.17 s. Indeed, in Topology 2 the mean delays per node are far smaller than in the other topologies, because all nodes

Table 2.4: Metrics for Topology 2

Topology 2	Sent	Received	PDR [%]	PDD [s]
Node 1	60	48	0.8	81.16
Node 2	59	39	0.661	74.49
Node 3	60	48	0.8	54.7
Node 4	60	37	0.616	81.08
Node 5	59	46	0.779	74.41

Table 2.5: Metrics for Topology 3

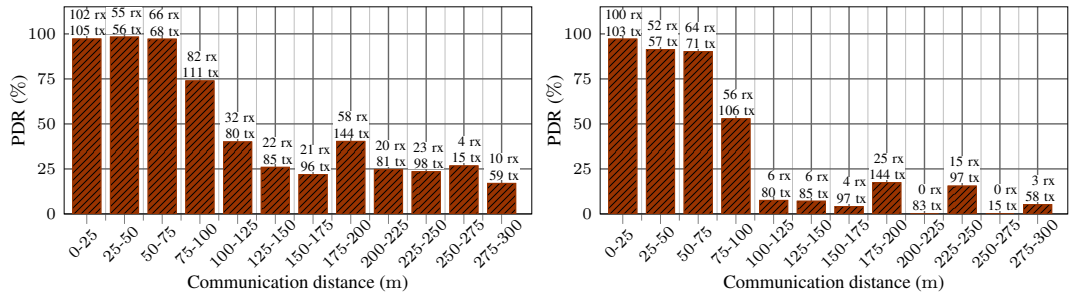
Topology 3	Sent	Received	PDR [%]	PDD [s]
Node 1	57	42	0.737	331.39
Node 2	59	46	0.779	187.57
Node 3	60	34	0.566	97.42
Node 4	60	44	0.733	237.95
Node 5	59	37	0.616	296.81
Node PI	60	32	0.533	296.81

are in range of the ASV approximately during the whole test, thus the reception time for the packets is determined mainly by the time the UW-POLLING protocol takes to poll every node that successfully takes part in the discovery phase.

From the analysis of the log files obtained during the test campaign, we observe that the PDRs change for the different topologies and depend on both preamble synchronization problems (as stated in [162]) and asymmetry of the acoustic link due to the ASV movement and the noise produced by its thrusters. The asymmetry of the acoustic link can be observed in Figure 2.19 where the PDR between the static nodes and the SeaML (Figure 2.19b) and the PDR between the SeaML and the static nodes (Figure 2.19a) are depicted. These results have been obtained in a preliminary tests done to analyze the physical transmission with the AHOI modems, before the underwater network trials with the DESERT Underwater Framework. The numbers on top of the bars indicate the number of received (rx) and transmitted (tx) packets. The communication from SeaML to the static nodes is better than the communication from the nodes to the SeaML. In both cases, the PDRs for distances from 0 m to 75 m are between 90% and 98 %. Conversely, the PDR for distances from 75 m to 100 m is 74% in the first case (SeaML to static nodes) and 53% in the second case (static nodes to SeaML). For longer distances from 100 m to 300 m the PDRs are between 17% and 40% (SeaML to static nodes) and between 0% and 17% (static nodes to SeaML). During the tests with the UW-POLLING protocol, in some cases a few packets are lost inside the burst of 5 packets sent by a node in its polling phase, while in other cases the burst of packets is completely (or almost completely) lost. In the last case, when the packets are lost for the asymmetry of the acoustic link, most of the control packets

Table 2.6: Metrics for Topology 4

Topology 4	Sent	Received	PDR [%]	PDD [s]
Node 1	50	28	0.560	233.89
Node 2	50	41	0.820	571.40
Node 3	60	37	0.616	267.01
Node 4	57	38	0.666	438.29
Node 5	56	41	0.732	255.55
Node PI	51	37	0.725	255.55



(a) SeaML (transmitter) ⇒ static nodes (receiver)

(b) Static nodes (transmitter) ⇒ SeaML (receiver)

Figure 2.19: PDRs between a mobile node (SeaML) and static nodes (five buoys and node on a jetty).

exchanged by the vehicles and the nodes (i.e., TRIGGER, PROBE, and POLL) are correctly received, but then the link quality drops faster due to the movement and the asymmetry mainly caused by the ASV propellers' noise, therefore the data packets are lost. While in the first case (preamble synchronization problems), packet loss happens independently in one or more packets inside the burst of 5 data packets, in the second case the whole burst of 5 packets (or at least the last part of a burst) is lost. Depending on the topologies and on the specific node position the prevalent cause of packet loss can be different. In Topology 1, the PDR at the MAC layer (i.e., without considering the packets that remained in the queue at the end of the simulation) is higher than 80% for all the nodes. In this case, except for node 5, the main cause of packet loss is a failure in the preamble synchronization. Even in Topology 2, the main cause of packet loss is the failure in the synchronization. Indeed, in this case most of the time the ASV is in range with all the nodes, therefore the ASV movement does not impact too much the reception rate. On the other hand, the channel asymmetry and the ASV movement become the prevalent cause of packet loss in Topology 3 and 4, where the ASV moves between clusters of nodes that are not within range of each other or are even separated by a physical obstacle (as in Topology 4).

Figures 2.20 digs deeper into the relation between the ASV positions and the re-

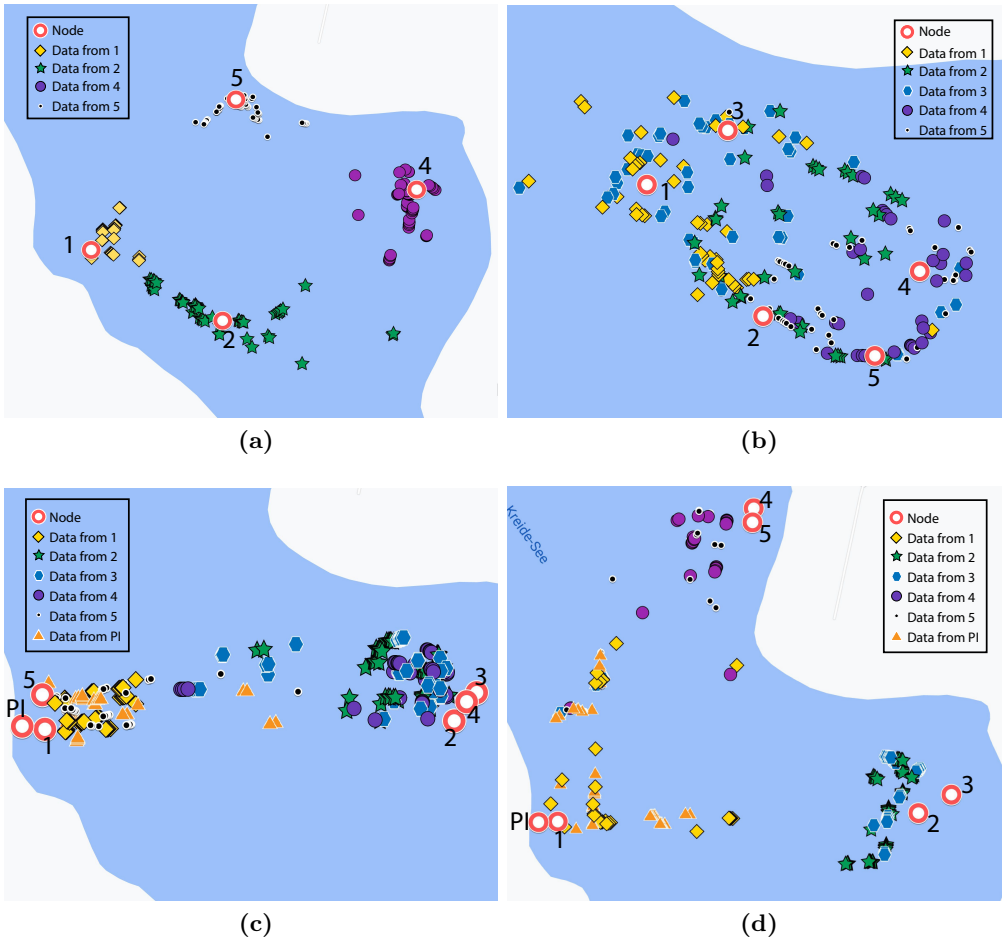


Figure 2.20: ASV's positions for each received packet for all topologies.

ceived packets, showing the actual ASV position for each received packet from each node in all the four analyzed topologies. In the figures, the packets received by the ASV have a different marker for each transmitting node, i.e., yellow diamonds for the packets received from Node 1, green stars for those received from Node 2, blue hexagons for the packets received from Node 3, purple circles for the packets received from Node 4, black dots for the packets received from Node 5 and orange triangles for the packets received from Node 6. In topologies where the distance between nodes or between clusters of nodes is greater than the maximum transmission range, the cluster formation is well defined in proximity of the interested nodes, like in Topologies 1, 3 and 4. Specifically, in Figure 2.20a we can observe the network behavior in Topology 1, where the ASV can receive the packets only when it is in the proximity of a single node, hence the derived figure shows a strong clustering of the packet receptions. Instead, in Topology 2 depicted in Figure 2.20b, the reception of packets by the sink node can happen even when the ASV was on the opposite side of the network, meaning that

Table 2.7: Fairness Index for every topology

	Topology 1	Topology 2	Topology 3	Topology 4
JFI	0.9755	0.9869	0.9874	0.9842

all the nodes are in range most of the experiment time period, depending on the instantaneous channel conditions. In Topology 3, presented in Figure 2.20c, the cluster formation is divided in two regions that correspond to the two main groups of nodes at the opposite sides of the lake. Some of the packets are received even in the middle of the lake since the ASV was still in range during their transmission. Finally, Figure 2.20d shows the received packets in Topology 4. Also in this case the three clusters are well defined. Still, there are some packets received even during the travel of the ASV from one cluster to another, as in Topology 3, but this occurs less often than in the previous case, especially between the group 4-5 and group 2-3, since the ASV was traveling in a clockwise path between the different groups. Thus, group 2-3 is shielded by the natural headland until the ASV reached at least the middle of that path segment.

Finally, the JFI for each topology is reported in Table 2.7: the closer the index to one, the more fairly the protocol treats the nodes of the network. The results show that all nodes are treated fairly by the polling protocol, since the JFI ranges from 0.976 to 0.987, depending on the network topology.

2.7 Conclusions

In this chapter we presented the performance evaluation of the UW-POLLING protocol in a high density acoustic network. We assessed the protocol in a data muling scenario where an AUV collects the data from sensor nodes, and forwards the data to the shore, either resurfacing and transmitting the data itself, or forwarding the data via acoustics to a sink node directly connected to the shore with a radio link. We analyzed how the protocol performance is impacted by the choice of the maximum backoff time, and how an adaptive strategy can be effective when the node density is either unknown, or not constant across the network area.

First of all, we analyzed the performance in a scenario with a uniformly distributed node deployment considering the nodes equipped with the EvoLogics S2C HS acoustic modem. We found, via simulation, the optimal backoff values for different fixed node densities. With the high rate modem, the backoff time does not have a big impact on the network performance: $T_{b_{Max}} \leq 2$ s is a value that basically maximizes the throughput for all the analyzed node densities. Moreover, we compared the fixed backoff strategy with two adaptive approaches based on the estimate of the number of neighbors: the full knowledge case (FC), used as a benchmark, and the realistic case (RC). We found that with node densities ≤ 200 nodes/km^2 the performance in the three cases (fixed backoff, FC and RC) is equivalent.

Since commercial off-the shelf modems are too expensive for an actual dense node

deployment in a civilian scenario, we also evaluated the UW-POLLING protocol performance with the low-cost, low-rate and vessel-noise resistant AHOI modem, developed by the Technical University of Hamburg. Also in this case, we identified the optimal maximum backoff time in a scenario with uniformly distributed nodes, considering different node densities. Differently from the high rate modem, in this scenario the maximum backoff time plays an important role in avoiding the collisions of PROBE packets and, therefore, in increasing the overall throughput of the network. Also in this case we compared the fixed backoff case with the adaptive cases. We considered the three different approaches (fixed backoff, FC and RC) in a scenario with both fixed and variable node density. In the variable density case, with the adaptive mechanisms we obtained an improvement in the overall throughput of about 8% compared to the fixed backoff case.

Then, we evaluate the protocol performance in the scenario of the port of Hamburg. In this scenario, a multimodal solution was employed to collect data from sensor nodes and forward it to the sink. To better simulate the channel conditions for the high speed modem, we employed the WOSS framework using the Bellhop ray tracer and the bathymetry of the port of Hamburg. We analyzed the overall throughput of the network and the fairness for the sensor nodes as a function of the offered traffic. In particular, we observed the fairness for three clusters of nodes, each one with a different node density. Our results showed that an adaptive backoff mechanism not only increases the overall throughput of the network, but also allows to achieve a better level of fairness in the areas with a high node density than in the fixed backoff case.

In this chapter we proved via field measurements-based simulations that the AHOI modem can be used to retrieve data from a dense underwater sensor network deployment. This modem is very low cost, but still requires the use of an off-the shelf transducer, whose price is twice the cost of the modem itself.

Lastly, in this chapter we also presented the lake test performed at lake Kreidesee in Hemmoor (Germany). The goal of the test was to evaluate the complete pipeline of the data collection service in a smart port scenario. Specifically, we evaluated the performance of the UW-POLLING protocol to understand the behavior of the protocol and to validate it in an actual deployment. We performed tests with different topologies to evaluate the polling-based protocol with different clusters of nodes in the network and therefore, with a different number of nodes participating to the same channel contention phase. The data from the sensor nodes was collected using the SeaML ASV equipped with both underwater modem and a WiFi antenna to forward the data to the shore station.

Chapter 3

LoRaWAN and Underwater Acoustic Networks in the Data Muling Scenario

3.1 Introduction

In this chapter we analyze the performance of E2E communications in the data muling scenario. Specifically, we will asses the performance of the whole communications pipeline, depicted in Figure 3.1, from the underwater sensor nodes generating the data to the collecting and monitoring station on the shore. As described in Chapter 2, in the data collection service an AUV retrieving data from underwater sensor nodes deployed in the harbor acts as a mule, forwarding all the collected data to surface buoys deployed along the network. These surface nodes, acting as sink for the underwater network, are equipped with two different communication interfaces, the first one being an underwater modem employed to gather packets from the AUV, and the second one used to send the received data to shore via radio.

The data collection in the underwater network is performed through acoustic communication employing the UW-POLLING MAC protocol described in Chapter 2.

The communication between surface nodes and the shore can be obtained using different RF technologies, from satellite to cellular networks, such as LTE [175, 176], each with different available bandwidth, coverage and cost [154]. Specifically, satellite communication is the most widely used technology in off-shore missions because it provides unlimited range, however its service cost is very high (e.g., the price of a V-SAT antenna is about €2000, with a service cost of €19 per Megabyte) and does not suit well an IoT application. LTE, on the other hand, can provide a broadband link up to a range of 30 km in an open-sea scenario, at a lower price (an antenna price of less than €60, and a service cost of about €0.01 per Megabyte). However, it requires the presence of an existing LTE cellular deployment: in case of no cellular coverage, an ad hoc deployment is not affordable for an IoT application, as the price of an LTE core starts from €300000, plus the cost of the bandwidth licence.

In our work we employ LoRaWAN [177], a Low Power Wide Area Network (LP-

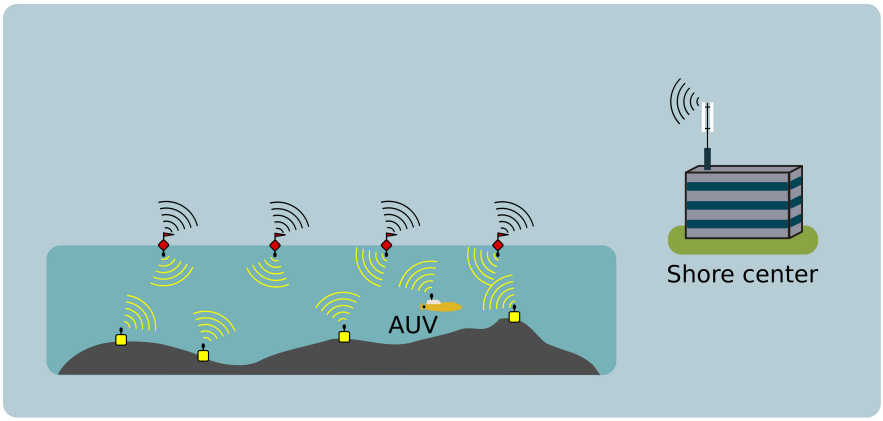


Figure 3.1: E2E environmental data collection scenario: an AUV collects data from static underwater sensor nodes, and forwards the data to surface buoys connected to shore via LoRaWAN.

WAN) technology [154], that seems particularly well suited for the application in this scenario: this kind of network operates in sub-GHz unlicensed bands, and has a transmission range in the order of kilometers, making it a very cost-effective solution, and enabling the communication between the shore gateway and the surface buoys deployed up to tens of kilometers away from the coast. The price of a LoRaWAN deployment is indeed very low, as one antenna costs less than €10, and the price of a LoRaWAN gateway is less than €100. It should be noted that these benefits come at the price of a low throughput and no delay guarantees, which however are usually not critical for sensor data collection applications.

To evaluate the performance of our system we use two different simulation tools for the underwater and above water parts. In particular, the DESERT Underwater Network Simulator [149] is used to simulate data collection from the underwater sensor nodes, while a `lorawan` module for ns-3 [178, 179] is employed to analyze the behavior of the above water network. The two simulators are connected by feeding the output of the DESERT simulator as the input for the ns-3 simulator: in such a way, we are able to track the arrival of the packets at the sink nodes and use this information to simulate the transmission of these packets to shore through the LoRaWAN network. In addition, we also implement in the ns-3 simulator the near-sea-surface propagation model presented in [154], which takes into account the characteristics of RF communications in a marine environment, such as the evaporation duct effect. This channel model allows us to simulate a scenario in which LoRaWAN nodes are deployed few kilometers in front of the harbor.

The goal of our work is to assess whether LoRaWAN is an enabling technology for the data collection service. In order to accomplish this, in this chapter we analyze the performance of the network when using acoustic modems with different capabilities, a variable number of sink nodes, and data generation frequencies, with the aim of locating possible bottlenecks in the mixed acoustic and LoRaWAN network.

The rest of this chapter is organized as follows: Section 3.2 describes the LoRaWAN technology used for the above water netowkr in the collection service, Section 3.3 describes the simulation scenario and parameters, Section 3.4 shows the results obtained via simulations, and Section 3.5 draws some conclusions.

3.2 LoRaWAN Data Forwarding

The E2E pipeline in the data collection service is composed by underwater and above water communications. While the former employs acoustic communications as described in the previous chapter, the latter, in our scenario, is performed thorough the LoRa technology. In this Section, we described the most important features of the above water technology.

Packets collected by the underwater interface of sink nodes are forwarded via radio to the shore through LoRaWAN, an LPWAN technology that operates on top of the LoRa modulation, a Chirp Spread Spectrum (CSS) technique that allows to trade bitrate for range. This trade-off is parameterized through the Spreading Factor (SF) setting, which can take values between 7 and 12, with higher values achieving longer ranges (up to several km in urban scenarios) and lower values sacrificing range for throughput. At the MAC layer, the LoRaWAN standard defines three main entities that operate in a network: End Devices (EDs) are defined as typically low power devices, that collect data and send it to one or more Gateways (GWs), equipped with chips that enable them to listen to multiple frequencies simultaneously, lock on multiple packets in parallel, and correctly demodulate packets overlapping in time, provided that some conditions on the employed Spreading Factor and reception power are respected [179]. GWs, in turn, are typically connected through a fast and reliable connection to a Network Server (NS), which is tasked with de-duplicating packets and controlling the network configuration.

3.3 Scenario Description and Simulation Setup

In our scenario the underwater network is composed of an AUV that patrols an area collecting data from underwater sensor nodes. The AUV forwards the collected packets to the surface nodes equipped with a LoRaWAN interface, used to relay the data to the LoRaWAN gateway placed on the shore. We assess the performance varying the number of sink nodes N_s used in the network. In addition, a set of LoRaWAN nodes is placed on the shore, in order to take into account the possible presence of a parallel LoRaWAN deployment operating as part of the port infrastructure, and creating interference at the gateway. These devices generate one 12-Byte packet once every 5 minutes, as was assumed in [154].

We assess the end-to-end performance with 3 different network configurations, analyzing whether the bottleneck is the underwater acoustic network or the LoRa network. In the first configuration, the commercial EvoLogics S2C HS [6] acoustic modem is used

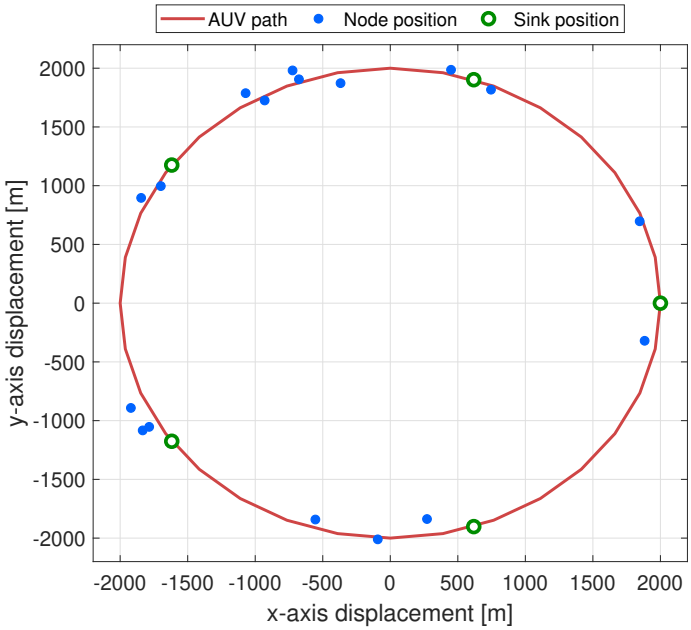


Figure 3.2: Scenario example with 5 sink nodes

for all the underwater nodes. This modem, already presented in Chapter 2, is a high-performance commercial off the shelf acoustic modem, which provides a nominal bitrate up to 62.5 kbit/s with a bandwidth of 60 kHz around a central frequency of 150 kHz and a transmission power equal to 156 dB re 1μPa. As the bitrate varies depending on the scenario, and differs from the actual datarate due to the error correction code used by the modem, in our simulations we set the S2C HS datarate to 7 kbit/s. Indeed, we consider shallow water transmissions in a river port with mobile nodes, that is quite a challenging scenario for acoustic communications. The packet error rate of this modem is modeled according to [5], as we have no field measurements for this specific modem.

In the second configuration the AHOI modem [55] is used for all the underwater nodes. The AHOI modem, which main features have been described in Section 2.4.1, is a prototype developed by the Technical University of Hamburg, with a transmission rate of 200 bit/s (potentially, a higher transmission rate is feasible in good conditions) and an extremely low cost. This modem transmits at a frequency of 62.5 kHz with a bandwidth of 25 kHz and a transmit power equal to 156 dB re 1μPa. The model used for the AHOI modem is based on field measurements in very shallow water, whose integration into the DESERT Underwater simulation framework has been presented in [11].

The last configuration consists of a multimodal setting and considers the use of both modems described above. Specifically, the AHOI modems are employed for the communication between the AUV and the sensor nodes, while the S2C HS modems are used for the communication between the AUV and the sink nodes. The AUV is

therefore equipped with both devices, while the sink nodes are equipped only with S2C HS, and the sensor nodes with the AHOI modem. In addition, the packet length in case AHOI modem is used is equal $L_{AHOI} = 24$ Byte, in order to limit the signal duration and, therefore, the Doppler effect [55]. With the S2C configuration, the packet length is limited by the maximum packet size allowed by the LoRaWAN standard, equal to $L_{S2C} = 220$ Byte.

The AUV moves at a constant speed of 2 m/s in a circular path of radius 2 km performing 10 laps. The nodes are randomly placed in an area 300 m wide around the AUV path according to a Poisson Point Process (PPP) with an average node density $\lambda = 5$ nodes/km^2. The sink nodes are equally spaced and placed along the AUV path. An example of the deployment with $N_s = 5$ sink nodes is depicted in Figure 3.2.

As soon as a packet is received by the underwater interface of a buoy node, it is directly forwarded to a LoRaWAN ED, which in turn transmits it to a GW placed on the shore using the LoRa modulation. Since we consider a European deployment of the LoRaWAN network, EDs have at their disposal three separate channels for uplink communication (at 868.1, 868.3 and 868.5 MHz), and randomly pick one for each transmission. Since the three frequencies are all placed inside the same regulatory sub-band, transmissions must respect a duty cycle of 1%: after each transmission of duration T seconds, a silent period of $99T$ seconds must be respected by the devices. Because of this limitation, data cannot typically be forwarded to the shore as soon as they are collected by a sink node but needs to be buffered until the next duty cycle, thus an additional delay will be experienced by the packets. On the shore, we assume the presence of a harbor deployment of a LoRaWAN network, creating additional interference and entailing the presence of a GW that will receive the packets forwarded by the surface buoys. In our application, we assume that all LoRaWAN nodes are using a Spreading Factor setting of 7 and a bandwidth of 125 kHz, thus employing the fastest available transmission rate that allows the usage of three separate channels for the uplink. Furthermore, we assume the presence of no confirmed traffic in the network, whose effect has been proven to be detrimental to the network performance if not carefully used [179].

3.3.1 Channel Model

To better characterize the communication performance in our scenario, the channel model for above water communications, presented in [154], has been implemented in the ns-3 simulator. Indeed, in a marine environment such as the one studied in this chapter, the Rayleigh fading model is no longer suitable, mainly because, as a result of the lack of obstacles between transmitter and receiver, the line-of-sight (LOS) component is usually the dominant part. In addition, reflections from the sea surface are probable and the evaporation duct, caused by water evaporation from the sea surface, makes communications possible even beyond the LOS. For these reasons, the path loss can be approximated with a two-rays model up to a given distance, where only the LOS component and the reflection with the sea surface take place. For longer distances, the model can be approximated with a three-ray channel model, which also includes

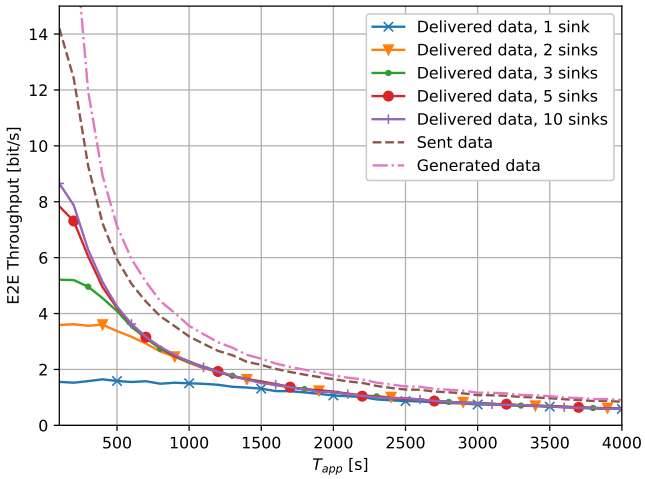


Figure 3.3: E2E throughput of the network with only AHOI modems. The throughput has been analyzed for different numbers of sink nodes.

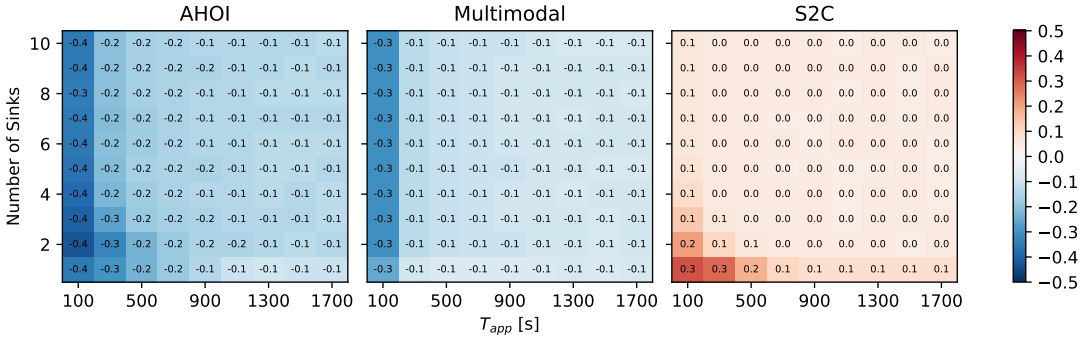


Figure 3.4: Identification of the bottleneck for different network configurations.

reflections with the evaporation duct layer in the atmosphere.

3.4 Results

In this section we present the results obtained through the simulation campaign. In order to simulate a realistic LoRaWAN deployment in the area akin to the port, we assume the presence of 300 interfering nodes, employing SF7 and sending packets once every 5 minutes, causing a loss of 3% of all packets sent by the surface buoys to the LoRaWAN GW.

Figure 3.3 shows the E2E throughput of a deployment in which AHOI modems are employed for all underwater communications, for varying values of the data generation period employed by the sensor nodes. Solid lines in the plot represent the throughput achieved by the network for different numbers of sink nodes, the dashed line represents

the rate at which packets are transmitted by the sensor nodes to the AUV, and the dash-dotted line is plotted as a reference for the rate at which data is produced by the sensors. For data generation periods higher than 2000 s, all generated data can be transmitted from the sensors to the AUV, and around 65% of the packets reach the sink first and the LoRaWAN GW next (the difference between generated data packets and received packets is mostly caused by packet losses due to bad channel conditions). For lower data generation periods, instead, the deployment density of sink nodes plays a more and more significant role in the degradation of the throughput of the system. For the lowest simulated generation period of 100 s, for instance, of the 28.8 bit/s that are generated by the applications only 14 bit/s can be transmitted to the AUV, and only 9 bit/s reach the LoRaWAN GW. In this heavy traffic scenario, increasing the number of deployed sinks has a direct effect on both the underwater and the LoRaWAN sections of the pipeline. Underwater, the AUV will have more chances to empty its queues: since it moves at a constant speed, when its queues are full of data its data delivery rate is determined by the amount of time it can spend in range of a sink node; the more sinks are available, the larger the amount of data it will be able to deliver. Similarly, above the water, the duty-cycled LoRaWAN network will benefit from a larger number of nodes that can deliver data in parallel.

Since these two effects both influence throughput significantly, we decided to create a metric to help assess which one is the bottleneck. Let t_o be the offered traffic, t_s the rate at which data is received at the sinks, and t_e the rate at which data is received at the LoRaWAN GW. The throughput bottleneck indication metric is then computed as follows:

$$b = \frac{1}{2}\left(\frac{t_s - t_e}{t_s} - \frac{t_o - t_s}{t_o}\right), \tag{3.1}$$

where the first term between parentheses represents the throughput loss caused by LoRaWAN, while the second term represents the throughput loss caused by the underwater section of the network. When $b > 0$, we can say that LoRaWAN is the bottleneck; for $b < 0$, instead, underwater communication is the more limiting section between the two.

Figure 3.4 shows the value of b for various generation periods and numbers of sinks, for all considered underwater modem solutions. For the AHOI system, the values are negative for every network configuration, indicating that the modem bit rate (which is heavily influenced by packet losses) is the main limitation to the E2E throughput. While this is also true for the multimodal solution, it's worth noting that in this case b takes values closer to zero, implying a better underwater performance. Finally, Figure 3.4 shows that when the S2C modem is used for all underwater communications, LoRaWAN becomes the bottleneck when a small number of sinks has to serve sensor nodes that transmit very frequently. This effect, however, can be easily mitigated by either increasing the number of sink nodes or by reducing the data generation frequency.

Figure 3.5 plots the Experimental Cumulative Density Functions (ECDFs) of the packet delay for the three modem solutions, with dashed lines representing the delay

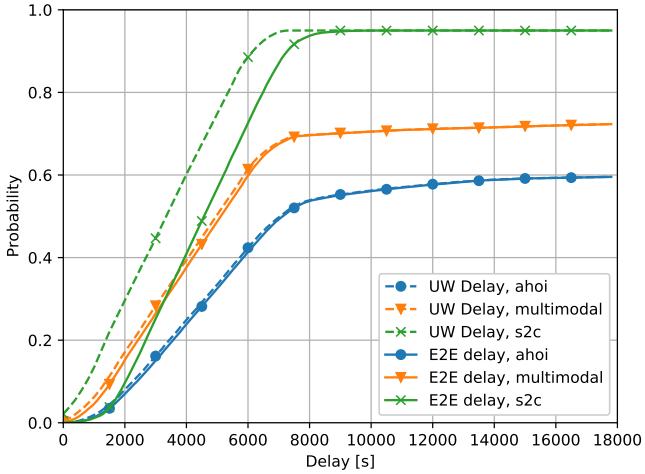


Figure 3.5: Comparison between E2E delay and underwater delay for the three different configurations. Results obtained with $T_{app} = 600$ s and 3 sink nodes.

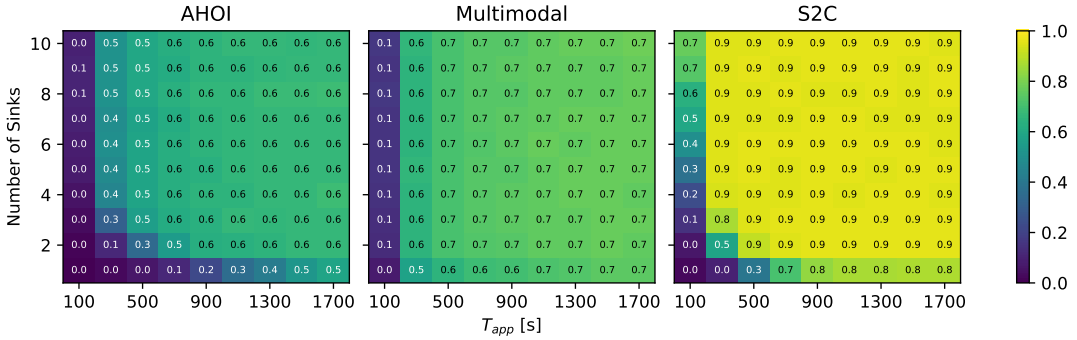


Figure 3.6: Fraction of packets delivered (E2E) in two rounds for the three configurations

introduced by the underwater section and solid lines representing the distribution of E2E delays. These results have been obtained considering 3 sink nodes. We note that ECDFs are computed by keeping into account all generated packets, which leads to the curves not reaching 1, since some generated packets are lost due to interference or noise and will never reach the GW. The contribution of LoRaWAN to the E2E delay can be estimated by comparing a dashed line with its solid counterpart: in the AHOI and multimodal cases, the additional delay introduced above the water is negligible, especially when compared to the underwater delay, which inherently suffers from the AUV having to physically move to collect the data, forcing some packets to wait in the node's buffer up to 6280 s before they can be retrieved by the AUV (i.e., the time it takes for the AUV to complete a lap of its path). In the S2C configuration, instead, LoRaWAN adds a considerable amount of delay, mainly because of its duty cycle constraints; this effect is significantly reduced when denser deployments of sink

nodes are used.

Figure 3.6 reports the fraction of delivered packets within two AUV rounds in the three configurations described in Section 3.3. In both the configurations with only AHOI acoustic modems and the multimodal setting, the network is not able to forward all the generated packets, even with low offered traffic and 10 sink nodes, mainly because of the packet losses suffered by the AHOI modems. Despite this, the multimodal configuration performs better than the setting with only AHOI modems. Considering the setting with only S2C modems, the network is able to handle the generated traffic in most cases. The network is not able to deliver any packets within two AUV rounds when $T_{app} < 300$ s and only one sink node is available; in this case, increasing the number of sink nodes is a solution. For all the other cases the fraction of delivered packets is greater than 90%. The reasons of these losses are various. First of all, packets are lost for bad channel conditions. Each packet is involved in three transmissions to reach the LoRaWAN GW to the shore, increasing the possibility of packet losses. We want to highlight that, according to the models used in our simulations, the AHOI modem performs worse than S2C in terms of packet delivery ratio. Indeed, the model used for the AHOI modem is based on field measurements retrieved in a very challenging scenario, while the S2C HS is modeled according to an empirical formula [5], that does not account for the multipath experienced in shallow water, as we had no field measurements for this specific modem. In addition, the different bitrate between the two underwater modem types allows S2C to deliver more packets. All these conditions favor configurations in which S2C is used. Other factors can decrease the fraction of delivered packets. In particular, when the AHOI modem is used to forward packets from AUV to sink nodes, the lower bitrate can cause some packets to be delayed in the AUV queue and thus to not be delivered to sink nodes within the two considered rounds. Similarly, if the number of packets received by a sink nodes is relatively high, the packets could be delayed due to the duty cycle restriction imposed by the LoRaWAN standard.

3.5 Conclusions

In this chapter, we evaluated the E2E performance of a data collection service from an underwater sensor network. We considered the data gathering with an AUV from sensor nodes based on different underwater network configurations, and we also included in the analysis the use of an above water network to forward data to the shore. Specifically, we analyzed a LoRaWAN network, that fits well in our scenario.

We assessed three different configurations for the underwater part of the network, using the low-cost prototype AHOI modem and the commercial S2C modem. We proved, via simulations, that LoRa is an enabling technology for data collection in most of the analyzed configurations. In particular, when considering the underwater network configuration with both only AHOI modems and the multimodal setting, LoRa does not introduce any further delay and does not act as the bottleneck of the network. Analyzing the configuration with only S2C modems, instead, we observed

that LoRaWAN may become the bottleneck of the network when the traffic generated by sensor nodes is high, but this problem can be mitigated at the cost of adding more sink nodes in the network.

Part II

Security in Underwater Acoustic Networks

Chapter 4

Game-Theoretical Analysis for Jamming Attacks in Underwater Acoustic Networks

4.1 Introduction

Underwater sensor networks have seen significant development over the last few years due to their extreme usefulness for both military and civilian applications [180]. Underwater sensor networks can be used in many application scenarios, such as oil and gas platform and pipeline maintenance, coastal and critical infrastructure surveillance, and environmental monitoring. As stated in Chapter 1, in the underwater environment, both radio and optical signals are greatly attenuated and hence, acoustic waves are the preferred way for wireless communications beyond about 50 m. However, underwater acoustic communications exhibit high latency due to the relatively slow speed of sound (1500 m/s, on average), high packet loss rate due to extended time-varying multipath, and low throughput due to distance-dependent bandwidth [10].

As the propagation environment is already hostile for underwater acoustic transmissions, DoS jamming attacks can be very effective at disrupting communications [181]. The simplest jamming attacks involve the transmission of a high-power signal that interferes with the legitimate signal in the same band and prevents its correct decoding. The significant differences between terrestrial networks, in which most jamming techniques and countermeasures have been studied, and UANs can lead to very different trade-offs [182] that might help or hinder the attacker. For example, the use of high-power jammers with no battery constraints is impossible in most underwater networks: such a jammer would require a large-scale operation using a boat or submarine, which would be detected long in advance and stopped from entering the area [183]. On the other hand, jamming is possible if the jammer is a small AUV, but this limits both the jamming power and the battery due to size and cost concerns. However, AUVs have

another advantage, as they can remain almost silent except for the jamming signal and flee at low speed, remaining almost undetectable after the jamming operation, which may last only a few minutes [183]. For this reason, detecting the presence of a jammer might not always be enough for the network to defend itself [184], particularly when there is a single receiver and triangulation is very difficult. This can be the case for many network deployments with a single "leader" floating node gathering data from multiple sensors and transmitting them to a boat or control station using above-surface electromagnetic communications [13].

Most of the radio-frequency literature on jamming countermeasures assumes that the jammer is *reactive*, sensing the packet transmission and jamming it with negligible delay [185]. As a consequence, it focuses on what we call *active* defense. In broad terms, active countermeasures aim at overcoming the jamming signal by strengthening the transmission, and involve an additional energy expenditure by the transmitter. There are three main types of active countermeasures that a transmitter can take to protect a message, potentially combining them to increase their effectiveness:

- *Power control*: the transmitter increases the transmission power [186], consequently increasing the Signal to Interference plus Noise Ratio (SINR) and so increasing the decoding probability, at the cost of spending more energy per packet;
- *MCS adaptation*: the transmitter can select a more robust modulation or a lower coding rate [187], thus reducing the data rate while increasing the probability of correct decoding. However, more energy is required to achieve the same SINR due to the longer duration of a packet transmission;
- *Packet-level coding*: the transmitter can encode the data packets and add some redundant packets [115], ensuring that the transmission is successful as long as a sufficiently large subset of packets is correctly decoded. This countermeasure does not change the power or duration of a packet transmission, but still increases the energy cost of each information packet, as the energy to transmit the redundant packets must be added to the tally.

However, there is also another type of countermeasure, which depends on the peculiar nature of underwater communications. Unlike in terrestrial networks, the long propagation delay of acoustic waves makes it possible for the transmitter to try avoiding the jamming signal entirely or partially. Indeed, depending on the link geometry, the malicious node might only be able to reactively jam part of each packet, or even not at all, reducing its effectiveness. In this case, the jammer can increase its chance to disrupt the transmission by acting *blind*, i.e., not reacting to sensed transmissions but proactively jamming the communication resources that it expects the transmitter to use. However, this makes another type of defense possible: if the legitimate transmitter can use multiple time slots or frequency channels for its transmission, it can try to avoid the jamming signal by randomizing its transmission pattern in time and frequency to increase the probability of transmitting its packets when the jammer is not active.

We call this type of countermeasure *evasive* defense: naturally, evasive defense has no effect against a reactive jammer, which can know exactly when a packet is being transmitted, albeit with some delay. In general, evasive defense can be applied over any kind of wireless communication, including radio-frequency electromagnetic communications. However, in these cases reactive jamming is almost always the best choice, as at the speed of light the propagation delay is much smaller, lower than 1 μs if the communication radius is below 300 m. In this case, the transmission time for each packet is often far longer than the propagation delay, making reactive jamming almost perfect and evasive defense a less attractive proposition. On the other hand, acoustic waves, which are often the only available medium for underwater communications over long distances due to the high electromagnetic attenuation of water, have a low bitrate and high propagation delays, making the trade-off between reactive and blind jammer less trivial than in terrestrial radio-frequency networks. In underwater acoustic networks, the propagation delay can also be higher than or comparable to the packet transmission time, resulting in lower effectiveness of a reactive jammer and making the blind solution more attractive. For these reasons, the geometry of the network affects the ability of a reactive jammer to promptly react to a transmission, making this problem completely different from its counterpart in terrestrial networks and well worth studying.

As stated above, evasive defense can be performed in both time and frequency, but there are two critical reasons to avoid frequency hopping. First, the bandwidth available for long-range acoustic transmissions is already very low in underwater scenarios, making it impractical to further reduce it by dividing it into subchannels. This is especially true for those modems that spread the signal over the whole bandwidth in order to mitigate the multipath distortion caused by the signal reflections with the sea surface and the sea bottom [6, 55]. Secondly, dividing the bandwidth to hop between different subchannels increases the transmission times for each packet, making it easier for a reactive jammer to listen for the transmission and jam it. On the other hand, transmitting over the whole available bandwidth ensures the minimum transmission time, giving little time to a reactive jammer to detect and disrupt the transmission, and as long as the total number of resources in time and frequency are the same, the effectiveness of the evasive defense against the blind jammer does not change. Continuous jamming could also be an option: the jammer could send a blanket jamming signal all the time, which would be impossible to evade. However, performing this kind of jamming with enough power to effectively disrupt the legitimate transmission would require a significant energy expense from the jammer, and would not be feasible for the battery-powered nodes that we consider.

In this chapter, we present a game theoretical analysis of the jamming attack in different scenarios and with different assumptions. In Section 4.2 we assess the effect of the jammer distance by only considering a blind jammer scenario, using packet level coding as reactive defense, and assuming perfect information, i.e., both transmitter and jammer know the opponent's position and battery level. In Section 4.3 we relax the complete information assumption, considering the jammer position unknown to the transmitter and modeling the jamming attack as a Bayesian game. In Section 4.4 we

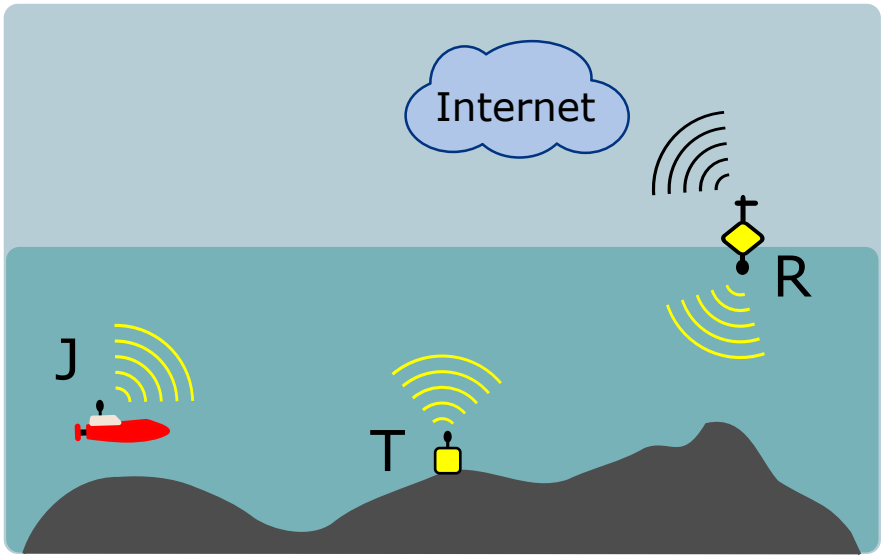


Figure 4.1: An underwater jamming attack: a jammer J tries disrupting the communication between a transmitter T and its intended receiver R.

introduce geometrical consideration comparing the effect of a blind and a reactive jammer for different network topologies, i.e., different relative positions between jammer, transmitter and receiver. Finally, in Section 4.5 we draw some conclusions.

4.2 Blind Jamming Effectiveness

In this Section, we analyze the effect of a blind jammer as a function of the distance between the jammer and the receiver.

4.2.1 Game Theoretic Model

We consider a transmitter T at a distance d_{TR} from a receiver R, under attack from a jammer J, which is placed at a distance d_{JR} from R. The scenario is shown in Figure 4.1: T needs to periodically send an update to a receiver R, and a malicious jammer J tries to block its transmission and deplete its battery.

In order to protect its transmission from the attack and from ambient noise, T uses packet-level coding as an active defense. Assuming an efficient packet-level code, the K information packets can be recovered if at least any K of the N coded packets are correctly received [188].

The jamming attack is modeled as a zero-sum game $\mathbb{G}$ between the two rational players T and J, i.e., a completely adversarial and symmetric game in which each gain for one player is balanced by a loss for the other [128]. The zero-sum model is justified by the fact that an attacker will naturally want to disrupt the operation of the legitimate node as much as possible, thus having completely adversarial goals;

this assumption is often used in jamming games. In this section, we study a complete information scenario. The complete information assumption is motivated by the fact that, at the end of each time frame, R sends a feedback packet containing information on how many packets from T it detected, how many slots were jammed by J, and how many packets it received successfully. We assume that such feedback packets are perfectly received by both J and T, as R is not power constrained.

The jamming game is composed of a series of packet transmission subgames G_m, with $m \in \mathbb{N}$. In each subgame, node T uploads its data to node R, in an attempt to report information on the surrounding environment. Such data is chunked into K payload packets, and T can exploit *(i)* Forward Error Correction (FEC) in order to increase the probability of successful communication over unreliable or noisy communication channels, and *(ii)* Cyclic Redundancy Check (CRC) to detect residual error-laden packets and discard them. In each subgame G_m, T can decide the amount of redundancy to use, i.e., the number $N_T^{(m)}$ of packets to send over the channel. A maximum of $2K$ transmission opportunities is configured in each subgame, thus $K \leq N_T^{(m)} \leq 2K$.

The outcome of each transmission attempt depends on the choices made by T and J, and on the conditions of the channel, which can be either modeled stochastically or obtained through experimental results. In particular, the transmission succeeds if T is able to counteract the channel impairments *and* the jamming attacks and to deliver at least K packets to the destination node within the duration of the subgame. We assume a packet erasure channel and an efficient code, so R can recover the K information packets if any K of the $N_T^{(m)}$ coded packets are correctly received [188].

Both players are battery-powered nodes, and the dynamics of the game are exhaustively characterized by their energy evolution, i.e., the evolution of their battery charge during the game. The battery levels take discrete values in the sets $\mathcal{B}_i \triangleq [0, 1, \ldots, B_i^{(0)}]$, $i \in \{T, J\}$, with $B_i^{(0)} \in \mathbb{N}$ being the initial charge of the battery. The battery levels in the sets $\mathcal{B}_i$ are normalized by the energy $E_{\text{tx},i}$, $i \in \{T, J\}$, used to transmit each legitimate packet or jam each slot; we consider the quantum $E_{\text{tx},i}$ to be constant, since our active defense strategy does not involve power control. Note that, as neither energy harvesting nor other forms of energy replenishment are considered, the battery levels can only decrease during the game. In each subgame, node T decides the number of packets $N_T^{(m)}$ to send to complete the data transmission, and this corresponds to an energy consumption of $N_T^{(m)}$ quanta, since battery levels are normalized. Note that, the larger $N_T^{(m)}$, the more robust the communication, but the faster the depletion of T's battery and the whole game duration. Similar energy considerations affect the choice of the jammer, which has to decide the number of transmission opportunities $N_J^{(m)}$ to jam in order to disrupt T's communication.

We now describe the structure of a single subgame and then illustrate the evolution of the multistage full game. Table 4.1 reports a summary of the notation used.

Table 4.1: Notation and meaning of system parameters for game players $i \in \{T, J\}$.

Parameter	Meaning
T	Transmitter
J	Jammer
R	Receiver
d_{in}	Distance between nodes i and n
K	Minimum number of packets to be delivered for success
τ	Duration of a packet transmission
Γ	Time horizon of multistage game $\mathbb{G}$
λ	Exponential discounting factor
α_i	Energy/PDR weighing factor
$u_i^{(m)}$	Payoff function in subgame m
$U_i^{(m)}$	Payoff function in multistage game $\mathbb{G}$ in subgame m
ε	Final state for the game
$\chi_i^{(m)}$	Indicator function of the success of subgame m
$f_i^{(m)}$	Energy penalty function in subgame m
$N_T^{(m)}$	Number of packets that T sends in subgame m
$N_{\mathrm{C}}^{(m)}$	Packets sent over clear channel in subgame m
$N_{\mathrm{B}}^{(m)}$	Packets sent over jammed channel in subgame m
$N_J^{(m)}$	Number of slots that J tries to jam in subgame m
$r^{(m)}$	Total packets delivered in subgame m
$r_{\mathrm{C}}^{(m)}$	Packets delivered over clear channel in subgame m
$r_{\mathrm{B}}^{(m)}$	Packets delivered over jammed channel in subgame m
$p_{e_{\mathrm{C}}}$	Packet error probability over clear channel
$p_{e_{\mathrm{B}}}$	Packet error probability over jammed channel
$B_i^{(m)}$	Battery level in subgame m
$\mathbf{B}^{(m)}$	State in subgame m, combining $B_T^{(m)}$ and $B_J^{(m)}$
$E_{\mathrm{tx},i}$	Energy required to transmit/jam a packet
$P_{\mathrm{tx},i}$	Transmission/jamming power
$\mathbf{f}^{(m)}$	Feedback in subgame m
$p(\hat{d}_{JR})$	Belief distribution of J's distance from R
Φ_i^*	Optimal mixed strategy for player i

4.2.1.1 The Packet Transmission Subgame

Each subgame G_m models the attempt made by T to transmit K information packets to R. The time after the beginning of the first packet transmission is slotted into a time frame of $2K$ time slots; each slot corresponds to the time τ necessary to transmit a packet. Note that the long propagation delays that characterize the underwater scenario give an advantage to T: the first packet can never be jammed, as the jammer does not have the time to sense the transmission and send the jamming signal. However, since J knows the duration of the time slot and the position of the transmitter and receiver, we assume that it can trigger its transmissions to perfectly jam the subsequent time slots.

Thus, T decides *(i)* how many packets $N_T^{(m)} \in \mathcal{N}_T^{(m)} \triangleq \{K, K+1, \ldots, \min(2K, B_T^{(m)})\}$ to send to R, and *(ii)* which time slots to employ for the transmission among the $2K$ available. Similarly, J chooses *(i)* the number of slots $N_J^{(m)} \in \mathcal{N}_J^{(m)} \triangleq \{0, 1, \ldots, \min(2K-$

$1, B_J^{(m)})\}$ to jam, and *(ii)* the $N_J^{(m)}$ jammed time slots out of $2K-1$ (as the first packet cannot be jammed). Note that the actions of both players are limited by the current battery level at stage m, i.e., $B_i^{(m)}$, $i \in \{T, J\}$. T and J make independent decisions on $N_T^{(m)}$ and $N_J^{(m)}$, respectively. Such decisions are made in advance for the whole time frame, right before the transmission of the first packet.

The payoffs of the players are convex combinations of monotonic functions of the energy required to transmit/jam the packets and of the PDR. By tuning the weight $\alpha \in [0, 1]$, the main objective of the players can be shifted between saving energy, thereby reducing $N_T^{(m)}$ and $N_J^{(m)}$, and delivering more packets. Based on these considerations, we express the players' payoffs for a single subgame m as:

$$u_T^{(m)} = \alpha\, f_T^{(m)} + (1-\alpha)\chi_T^{(m)} \tag{4.1}$$

$$u_J^{(m)} = -u_T^{(m)}\,. \tag{4.2}$$

The first term of Equation (4.1) is related to energy, while the second term concerns the outcome of the communication. In particular, the indicator term $\chi_T^{(m)}$ is equal to one if the subgame m ends with T successfully delivering at least K packets to R, and zero otherwise.

Function $f_T^{(m)}$ gives T a penalty for consuming energy when transmitting packets. In particular, we set:

$$f_T^{(m)} = -\frac{N_T^{(m)}}{(2K+1)}. \tag{4.3}$$

The additional term 1 in the denominator of (4.3) is arbitrary and ensures that the absolute value of $f_T^{(m)}$ is always smaller than 1, thus preventing any strategy to be dominated by not transmitting at all. Moreover, notice that the number of slots $N_J^{(m)}$ jammed by node J is not explicitly present in the payoffs for the single subgame, since we assumed a zero-sum game. Nevertheless, $N_J^{(m)}$ still plays a major role in the full game: the larger $N_J^{(m)}$, the higher the energy consumed by node J, and the faster its battery depletion.

Finally, the transmitter's choice of the time slots in which to transmit packets, and the jammer's choice of which time slots to jam, can be modeled as a simple anti-coordination game [189]: T's objective is to avoid the jammer and transmit as many of its packets as possible on a clear channel, while J's objective is to correctly guess the slots that T will use and jam them, so as to maximally disrupt the communication.

4.2.1.2 The Full Jamming Game

In a battery-limited scenario, the greedy strategy that maximizes the payoff for the next subgame is not always optimal. The solution of the full jamming game $\mathbb{G}$ maximizes a long-term payoff function within a given time horizon Γ, which represents the number of future subgames to consider in the payoff. The players' payoffs in the multistage

game $\mathbb{G}$ at stage m are given by:

$$U_i^{(m)}(\Gamma) = \sum_{\gamma=m}^{m+\Gamma-1} \lambda^{\gamma-m} u_i^{(\gamma)}, \quad i \in \{T, J\} \tag{4.4}$$

where $\lambda \in [0, 1]$ is a future exponential discounting factor [190], $u_i^{(m)}$, $i \in \{T, J\}$ is the subgame payoff defined in (4.1) and (4.2), and Γ is the length of the payoff horizon, i.e., the number of subgames that are considered. When Γ is finite, we can consider $\lambda = 1$ with no convergence issues, while, for $\Gamma = +\infty$, we must consider $\lambda < 1$. Note that the payoff $u_i^{(m)}$ for a single subgame coincides with $U_i^{(m)}(1)$. In general, J will behave in a foresighted manner if Γ is large enough: its energy expenditure is not explicitly penalized, but it can reduce the reward if it affects the number of subgames it can play in.

4.2.2 Analytical Solution of the Game

In this Section, we explain how to derive the optimal strategies for the two players in the case of perfect knowledge about the opponent's position and battery level at the beginning of each subgame. We define as strategy s_i the action chosen by player $i \in \{T, J\}$, i.e., the amount of energy required to transmit the legitimate packets or to jam the slots, respectively. According to the game defined in Section 4.2.1, the strategy space is thus $\mathcal{N}_i^{(m)}$ $i \in \{T, J\}$ in each subgame. Note that the strategies concern what to do in each subgame, but are chosen based on the expected evolution over multiple subgames, as dictated by Γ. We are interested in evaluating the Nash Equilibrium (NE), i.e., the pair of optimal strategies (s_T^*, s_J^*) that are mutual best responses [191]. In other words, a NE is reached when neither player can improve its expected payoff by changing its strategy unilaterally. Since the payoff functions of the two players (see (4.4)) can include multiple subgames, the NE of the jamming game can be calculated exactly with *dynamic programming*. The NE may be *pure*, i.e., correspond to deterministic strategies, or *mixed*, when strategy $s_i^{(m)}$ for player $i \in \{T, J\}$ is a probability distribution $\Phi_{s_i}(N_i)$ over $\mathcal{N}_i^{(m)}$. Under the assumption of complete information, strategies are determined by the state of the two players, assuming an optimal strategy for lower battery states.

In the following, we first present the expressions for the expected payoffs of nodes T and J that are needed to compute the NE, and then describe the procedure to solve the game analytically through dynamic programming.

4.2.2.1 Expected Payoff Calculation

To derive the NE, we need to characterize the expected payoff for a single subgame, denoted as $\mathbb{E}\left[U_i^{(m)}(1) \middle| N_T^{(m)}, N_J^{(m)}\right]$ for the m-th stage of game $\mathbb{G}$. Such expected payoff is equal to the expectation of the payoffs $u_i^{(m)}$, $i \in \{T, J\}$ given in Equations (4.1) and

(4.2). In the remainder of this section, we will omit superscript (m) for the sake of a lighter notation.

The expected payoffs $\mathbb{E}\Big[u_i\Big|N_T,\ N_J\Big]$, $i \in \{T, J\}$ can be calculated from the quantity $\mathbb{E}\Big[\chi_i\Big|N_T,\ N_J\Big]$, $i \in \{T, J\}$, which represents the expected outcome of the subgame (as introduced in Section 4.2.1.1, χ_T and χ_J are indicator terms for the transmission and jamming success, respectively). We introduce quantities $N_{\mathrm{C}} \leq N_T$ and $N_{\mathrm{B}} \leq N_T$ to indicate the number of packets that node T sends over a clear and blocked (i.e., jammed) channel, respectively. Obviously, $N_T = N_{\mathrm{C}} + N_{\mathrm{B}}$, so we can easily obtain the value of N_{C} once we know N_{B}. We also know that $N_C \geq 1$, as the first packet can never be jammed. Using the law of total probability, for node T we have:

$$\mathbb{E}\Big[\chi_T \ \Big|\ N_{\mathrm{T}},\ N_{\mathrm{J}}\Big] = \sum_{N_{\mathrm{B}}=0}^{N_{\mathrm{T}}-1} \mathbb{E}\Big[\chi_T \ \Big|\ N_{\mathrm{B}}, N_{\mathrm{T}}\Big] \mathrm{P}\Big(N_{\mathrm{B}} \ \Big|\ N_{\mathrm{T}},\ N_{\mathrm{J}}\Big) \tag{4.5}$$

The first term inside the summation is the expectation of a subgame success, given the number of packets successfully delivered and jammed during that subgame, and can be expressed as:

$$\begin{aligned}\mathbb{E}\Big[\chi_T \Big| N_{\mathrm{B}}, N_{\mathrm{T}}\Big] = \sum_{r=K}^{N_T} \sum_{r_{\mathrm{B}}=0}^{r} \binom{N_{\mathrm{T}} - N_{\mathrm{B}}}{r - r_{\mathrm{B}}} p_{e_{\mathrm{C}}}^{(N_{\mathrm{T}}-N_{\mathrm{B}})-(r-r_{\mathrm{B}})} \\ \times (1 - p_{e_{\mathrm{C}}})^{r-r_{\mathrm{B}}} \binom{N_{\mathrm{B}}}{r_{\mathrm{B}}} p_{e_{\mathrm{B}}}^{N_{\mathrm{B}}-r_{\mathrm{B}}} (1 - p_{e_{\mathrm{B}}})^{r_{\mathrm{B}}}\,.\end{aligned} \tag{4.6}$$

The external summation iterates on all possible values of the number of delivered packets $r \leq N_T$ resulting in a success. Equation (4.6) then splits r between packets that are delivered over a jammed channel, i.e., $r_{\mathrm{B}} \leq r$, and those which are delivered over a clear channel, i.e., $r - r_{\mathrm{B}}$. For the two cases, the packet error probability is equal to $p_{e_{\mathrm{C}}}$ and $p_{e_{\mathrm{B}}}$, respectively, and is a function of the SNR or SINR.

We consider a realistic modulation, such as CSS, which is used in several real underwater acoustic modems. If a different modulation is used, the only required change is in (4.7), while the rest of the model remains the same. The packet error probability in the case of a jammed signal, p_{eB}, can be computed as presented in [192], which computes the BER considering Differential Quadrature Phase Shift Keying (DQPSK) modulation in a radio frequency channel. We now adapt the definition of the BER to the acoustic underwater scenario. The BER for a CSS signal, $p_{\mathrm{bit}}^{\mathrm{CSS}}$, is computed as:

$$p_{\mathrm{bit}}^{\mathrm{CSS}} = Q(a, b) - \frac{1}{2} e^{-(a^2+b^2)/2} I_0(ab) \tag{4.7}$$

where Q is the Marcum Q function, I_0 is the modified Bessel function of order 0, and

a and b are defined as:

$$\begin{aligned} a &= \sqrt{\frac{2E_b/N_0}{1+J_0/N_0}(1-\sqrt{0.5})} \\ b &= \sqrt{\frac{2E_b/N_0}{1+J_0/N_0}(1+\sqrt{0.5})} \end{aligned}, \tag{4.8}$$

where E_b is the received energy per bit of the transmitter, N_0 is the noise power spectral density, and J_0 is the power spectral density of the jammer, given by:

$$E_b = \frac{\tau}{L} P_{\mathrm{tx},T}\, g_{\mathrm{T}}, \qquad J_0 = \frac{P_{\mathrm{tx},J}\, g_J}{B}\,. \tag{4.9}$$

The SINR is then given by $\mathrm{SINR} = \frac{E_b L/\tau}{N_0 B + J_0 B}$, where L is the packet length in bits, $P_{\mathrm{tx},i}$ represents the transmit power of node $i \in \{T, J\}$, B is the transmission bandwidth, and g_T and g_J model the gain of the underwater acoustic channel between T and R and between J and R, respectively. Their values depend on the distances d_{TR} and d_{JR} to the receiver, respectively, as well as on the carrier frequency of the signal. Both noise and channel gain for an underwater acoustic channel can be computed as described in [5].

Finally, the packet error probability $p_{e_{\mathrm{B}}}$ is given by:

$$p_{e_{\mathrm{B}}} = 1 - \left(1 - p_{\mathrm{bit}}^{\mathrm{CSS}}\right)^L, \tag{4.10}$$

We also consider the packet error probability if a Reed-Solomon (RS) channel code is employed [193]. We analyzed the performance with an RS(127,78) with $q = 7$ bits per symbol and an error correction capability of $t = 24$ symbols. In this scenario, the packet is lost if more than t symbols are not received correctly:

$$p_{e_{\mathrm{B}}} = \sum_{i=t+1}^{N} \binom{N}{i} p_s^i (1-p_s)^{(N-i)} \tag{4.11}$$

where $p_s = 1 - \left(1 - p_{\mathrm{bit}}^{\mathrm{CSS}}\right)^q$ is the symbol error probability.

Finally, the second term in Equation (4.5) can be expressed as:

$$\mathrm{P}\Big(N_{\mathrm{B}} \;\Big|\; N_T,\, N_J\Big) = \frac{\binom{N_T-1}{N_{\mathrm{B}}}\binom{(2K-1)-(N_T-1)}{N_J-N_{\mathrm{B}}}}{\binom{2K-1}{N_J}}\,, \tag{4.12}$$

where we have imposed the condition that the first transmitted packet cannot be jammed due to the signal propagation characteristics of the underwater scenario, as described in Section 4.2.1. Naturally, (4.12) is only valid for $N_B < N_T$. In Equation (4.12) , we assume that both the transmitter and the jammer choose the slots to transmit (or jam) according to a uniform distribution among all possible N_T-tuples (or N_J-tuples) of slots. This is the choice that maximizes (for the transmitter) or minimizes

(for the jammer) the probability that at least K slots in the transmission are free from collision. This strategy pair is the NE for the anti-coordination slot selection game we mentioned in Section 4.2.1.1: since all slots after the first have the same success probability, the optimal strategy for both players is to randomly choose $N_T - 1$ and N_J among them. Any other strategy would be strictly dominated, since it would provide the opponent with a pattern to exploit: if T chooses a slot with high probability, J will try to mirror it and jam the communication more effectively. The only exception to this is the first slot, which the jammer cannot jam; it is trivial to show that a strategy that includes it with probability 1 and selects the others with uniform probability strictly dominates any others for the transmitter.

Substituting (4.6) and (4.12) into (4.5), we can finally obtain the expected value of the indicator function $\chi_i^{(m)}$ and then the expected value of the payoffs $u_i^{(m)}$.

4.2.2.2 Dynamic Programming Solution

In the case of complete information, an optimal solution of the multistage game can be determined through a dynamic programming procedure. We define the system state as $S^{(m)} \triangleq \left(B_T^{(m)}, B_J^{(m)}\right)$, where $B_i^{(m)}$ is limited by the initial battery level $B_i^{(0)}$ of player $i \in \{T, J\}$. The state space is then defined as $\mathcal{S} = \{0, \ldots, B_T^{(0)}\} \times \{0, \ldots, B_J^{(0)}\}$. If $\Gamma > 1$, the payoff in state $S^{(m)}$ takes the payoff of the future $\Gamma - 1$ subgames into account. The game ends when the transmitter's battery level is too low to transmit at least K packets, i.e., when $B_T^{(m)} < K$. We can aggregate all states that satisfy the ending condition into a final state ε and define its payoff as:

$$U_i^{(m)}\left(\Gamma \;\middle|\; S^{(m)} = \varepsilon\right) = 0 \quad \forall\, i, \Gamma\,. \tag{4.13}$$

We can now compute $\mathbb{E}\left[U_i^{(m)}(\Gamma) \;\middle|\; S^{(m)}\right]$ recursively for all other states, considering that the battery charge can never increase, hence $B_i^{(m+1)} \leq B_i^{(m)}\, \forall i, m$. It is:

$$\begin{aligned} \mathbb{E}\left[U_i^{(m)}(\Gamma) \;\middle|\; S^{(m)}\right] = \mathbb{E}\left[u_i^{(m)} \;\middle|\; S^{(m)}\right] + \\ \lambda \sum_{S \in \mathcal{S}} \mathbb{E}\left[U_i^{(m+1)}(\Gamma - 1) \;\middle|\; S\right] \mathrm{P}\left(S^{(m+1)} = S \;\middle|\; S^{(m)}\right). \end{aligned} \tag{4.14}$$

The payoff in a state is thus computed as the expected payoff $\mathbb{E}[u_i^{(m)}]$ obtained in the subgame corresponding to that state plus the payoff that is expected to be obtained in the next $\Gamma - 1$ subgames, with an exponential discount factor λ (see Equation (4.4)). This latter term is calculated by averaging over each possible next state $S^{(m+1)}$ weighed by the probability of transitioning to that state. For a given pair of strategies (s_T, s_J), such state transition probability is given by:

$$\begin{aligned} \mathrm{P}\left(S^{(m+1)} = (B_T, B_J) \;\middle|\; S^{(m)}\right) = \\ \Phi_{s_T}\left(B_T^{(m)} - B_T^{(m+1)}\right) \Phi_{s_J}\left(B_J^{(m)} - B_J^{(m+1)}\right). \end{aligned} \tag{4.15}$$

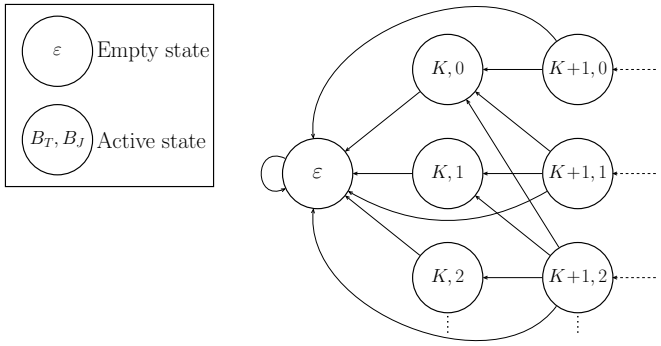


Figure 4.2: State transitions for the multistage game $\mathbb{G}$.

By substituting (4.15) into (4.14), we have a full recursive formulation for the expected long-term payoff $\mathbb{E}[U_i^{(m)}(\Gamma)]$ for any strategy pair. Once the payoff bimatrix is thus constructed, the Lemke-Howson algorithm can be used to find the mixed NE [194]. By starting from the lowest states and calculating the expected payoffs $\mathbb{E}\left[U_i^{(m)}(\gamma)\right]$, $\gamma \in \{1,\ldots,\Gamma\}$, the game can be solved completely. Figure 4.2 shows the state transition graph for the multistage game $\mathbb{G}$. Transitions are allowed from bottom to top and from right to left, as a consequence of nodes T or J consuming energy to send packets or jam slots, respectively. The game ends at stage $h \in \mathbb{N}$ when state ε is reached, i.e., $B_T^{(h)} < K \leq B_T^{(h-1)}$. Notice that, if the battery of J empties before T's, the game evolves in the limit condition of T playing against the channel.

4.2.2.3 Analytical Performance Evaluation

After computing the strategies, we can evaluate the expected lifetime $\mathbb{E}\left[L|S^{(m)}\right]$ of the transmitter node, defined as the number of blocks that it can transmit (either successfully or not), i.e., the number of subgames that will be played before its battery is depleted. Using (4.15), we define the expected lifetime for state $S^{(m)}$ recursively:

$$\mathbb{E}\left[L \,\middle|\, S^{(m)}\right] = \sum_{B_J=0}^{B_J^{(m)}} \sum_{B_T=0}^{B_T^{(m)}-K} \left(1+\mathbb{E}\left[L \,\middle|\, S^{(m+1)}=(B_T,B_J)\right]\right) \times \mathrm{P}\left(S^{(m+1)}=(B_T,B_J) \,\middle|\, S^{(m)}\right). \quad (4.16)$$

The lifetime takes into account the subgame (m), which is summed to the expected lifetime of each possible next state $S^{(m+1)}$, weighed by its probability. Since the game ends in state ε, we can now define the base step of the recursive formulation:

$$\mathbb{E}\left[L \,\middle|\, S^{(m)}=\varepsilon\right] = 0. \quad (4.17)$$

We can also derive the expected success probability $P_S\left(S^{(m)}\right)$ using the same reasoning. The success probability for the current subgame is averaged with the success probability in future states, weighed by the expected lifetime and the probability of reaching those states using (4.16):

$$\begin{aligned} P_S\left(S^{(m)}\right) = & \sum_{N_T=K}^{2K} \sum_{N_J=0}^{2K-1} \mathrm{P}\left(S^{(m+1)} = S_{N_T,N_J}^{(m+1)} \;\middle|\; S^{(m)}\right) \\ & \times \frac{\mathbb{E}\left[\chi_T | N_T, N_J\right] + \mathbb{E}\left[L \;\middle|\; \left(S_{N_T,N_J}^{(m+1)}\right)\right] P_S\left(S^{(m+1)}\right)}{1 + \mathbb{E}\left[L \;\middle|\; S_{N_T,N_J}^{(m+1)}\right]}, \end{aligned} \tag{4.18}$$

where $S_{N_T,N_J}^{(m+1)} = \left(B_T^{(m)} - N_T, B_J^{(m)} - N_J\right)$. The base step is the same as for the lifetime:

$$P_S\left(\varepsilon\right) = 0. \tag{4.19}$$

4.2.2.4 Computational Complexity

The computation of the optimal strategies requires the dynamic programming approach described in Section 4.2.2.2, starting from the states with the lowest battery level and exploiting the results to calculate the strategy for the subsequent ones. The solution of the game in a given state $S = (B_T, B_J)$ then requires the knowledge of the expected payoff for the current subgame and for future ones, which are then given as input to the Lemke-Howson algorithm to find the NE. If we denote the complexity of finding the expected payoff of a subgame as M_{sub}, and the complexity of the Lemke-Howson algorithm as M_{LH}, the overall complexity of the solution in state S is $O(B_T B_J M_{\text{sub}} M_{\text{LH}})$.

While the Lemke-Howson algorithm is efficient in practice, its worst-case complexity has been shown to be $M_{\text{LH}} \sim O(2^{3K})$ [195], as it depends directly on the length of the longest pivoting path in the strategy space. [1]

We now compute M_{sub}, the last term of the overall complexity formula. In order to find the expected payoff in a subgame, the nodes need to solve (4.5) for each pair of possible moves (N_T, N_J). The solution of (4.5) requires to compute $\mathbb{E}\left[\chi_T \,|\, N_{\text{C}}\right]$ using (4.6) N_T times, and (4.6) requires $O(K^2)$ operations in the worst case. Therefore, the complexity of (4.5) is $O(K^3)$, and computing the expected payoff for all the $(K+1)(2K-1)$ possible pairs of moves is $O(K^5)$.

The overall time to find the solution of the game in state S is then $O(B_T B_J K^5 2^{3K})$, which is clearly intractable for a computationally limited underwater node. However, the optimal strategies can be computed offline and loaded in the agent as a simple lookup table, reducing the complexity to a simple memory read.

4.2.3 Scenario settings

We evaluate the performance of the optimal strategies by studying the energy consumption and the PDR of T in two scenarios, a model-based one and an experimental

[1] We remind the reader that K is the number of information packets in a burst.

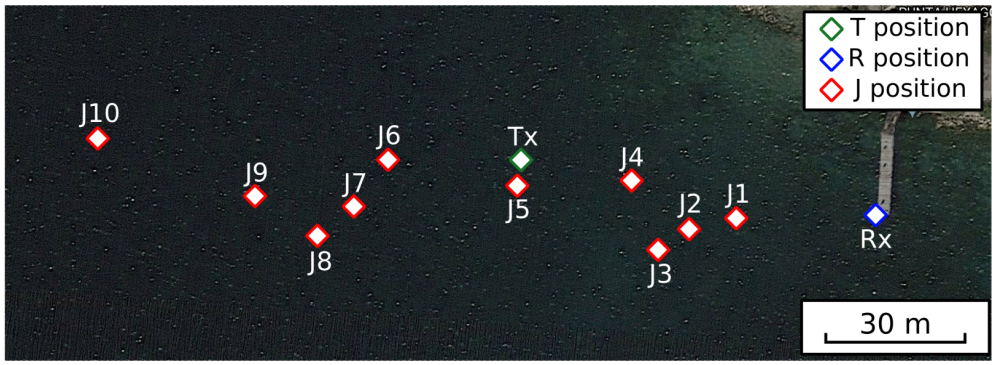


Figure 4.3: Node deployment in the Garda lake. The figure reports all the positions (red diamond) in which the jammer node was placed during the experiment. Transmitter position (green diamond) and receiver position (blue diamond) are reported as well.

one. We set up the two scenarios using the same transmitter and scenario parameters, trying to make them as comparable as possible: for this reason, the relative positions of the three nodes (transmitter, jammer, and receiver) are the same in both scenarios. We considered a carrier frequency equal to 26 kHz and a bandwidth of 16 kHz. The transmit power was the same for both transmitter and jammer, namely $P_{\text{tx},i} = 180$ dB re 1μPa, $i \in \{T, J\}$. However, the two scenarios used different packet error probabilities, derived from a theoretical model and a lake experiment, respectively.

4.2.3.1 Model-based Scenario

In this scenario, jammer and transmitter are trained and evaluated using the uncoded CSS modulation with DQPSK; the packet error probability is given in (4.10). As mentioned above, the considered propagation model is described in [5], where only the Line of Sight (LoS) component is considered. The wind speed, shipping factor and geometrical spreading factor are set to 3 m/s, 1, and 1.75, respectively. The channel settings (with few reflections and a strong line of sight component) are optimistic, as real scenarios in shallow water often have strong reflections and environmental noise. The parameters of the model are summarized in Table 4.2.

4.2.3.2 Experimental Settings

The lake experiment took place in the Garda lake on Thursday 17th October 2019, just off the Bardolino town coastline. The weather was sunny, and the maximum wind speed we experienced during the experiment was 8 m/s. Most of the waves were caused by the motion of the surrounding ships: shipping activity was very heavy, as our network was deployed at only 500 m from the Bardolino ferry station, and the receiver node was placed close to a boat rental service. All the measurements were performed from 10 AM to 4 PM. The experimental setup was composed of 3 nodes equipped with EvoLogics S2C R 18/34 WiSE modems [6]: the receiver was deployed from a floating

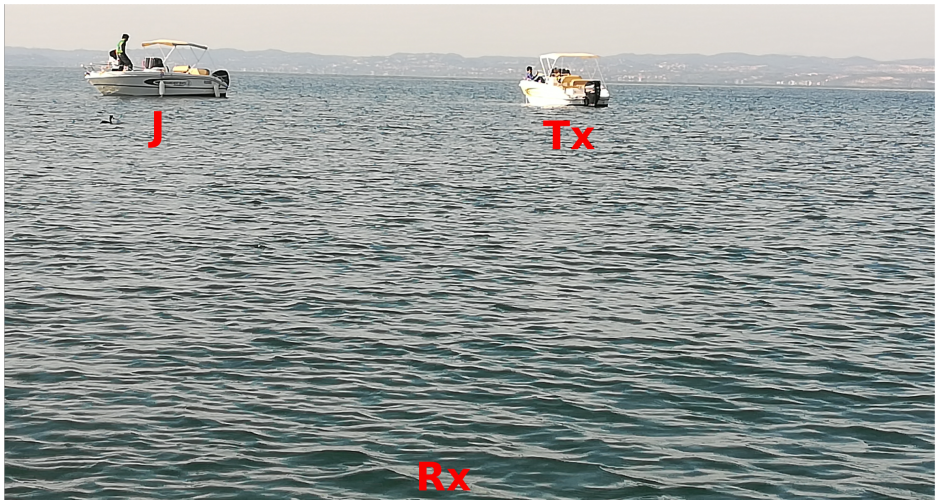


Figure 4.4: Picture of the experiment taken from the receiver node station when the jammer was in position J5 (Figure 4.3).

Figure 4.5: Picture of the apparatus used in the experiment. Each node was equipped with batteries, a laptop and an acoustic modem.

pier (N 45.549108, E 10.715181), the transmitter from a working boat anchored 80 meters west of the receiver (N 45.549165, E 10.714172), and the jammer from a working boat placed at different locations, between 20 and 180 meters west of the receiver. The map of the node positions is shown in Figure 4.3, Figure 4.4 is a photo of the scenario from the receiver's perspective, and Figure 4.5 shows the equipment used for each node in the experiment. The water depth was 4 m at the receiver, 10 m at the transmitter, and varied from 4 to 15 meters at the jammer, depending on its location. All nodes were deployed at a depth of 2 m, and both J and T were sending signals with an acoustic power of 180 dB re 1μPa. Both modems deployed from the Tx and the Rx stations used the standard EvoLogics firmware, while node J was transmitting continuous signals at 1 kbit/s by using the low-level EvoLogics firmware, described in [196]. Every 2 seconds, T sent one instant message packet with a payload length of 64 Bytes at the same bitrate

Table 4.2: Parameters setting.

Parameter	Value
Modem carrier freq	26 kHz
Modem bandwidth	16 kHz
Modem bitrate	1 kbit/s
Payload length	64 Bytes
T and J P_{tx}	180 dB re 1μPa
$p_{e_{\text{C}}}$ (lake exp)	0.04
$p_{e_{\text{C}}}$ (models)	0
Spreading factor k	1.75
Shipping factor s	1
Wind speed w	3 m/s

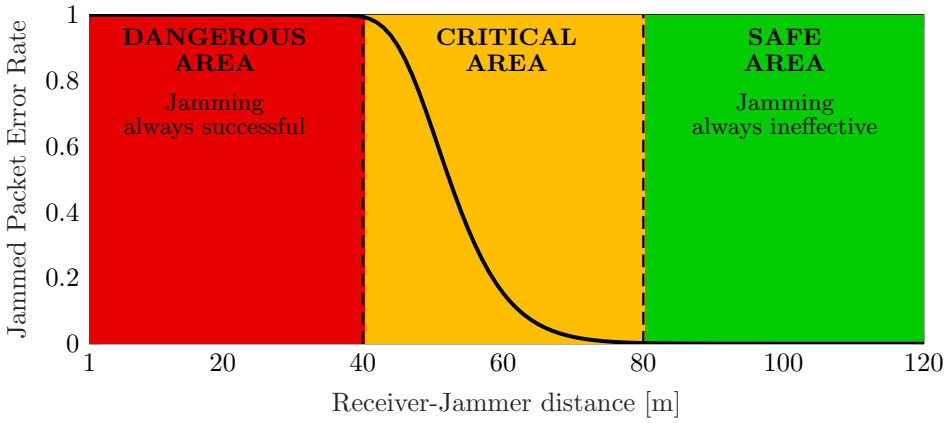


Figure 4.6: Blocked channel packet error rate $p_{e_{\text{B}}}$ for a jammed slot as a function of the distance d_{JR} between J and R when the distance between T and R is $d_{\text{TR}} = 78$ m, using the uncoded model.

of J. Together with the EvoLogics header and coding used by the standard EvoLogics, the packet duration was approximately 0.86 s (value provided by the modem at the moment of the reception). In order to prevent T's transmissions from being blocked by the reception of J's signals (as the acoustic modems are, for their nature, half-duplex devices), T was set in deaf mode, i.e., its receiver unit was disabled. Both T and J used a Quadrature Phase-Shift Keying (QPSK) modulation, with each symbol spread to the whole bandwidth (using the so-called sweep-spread carrier (S2C) technology).

4.2.4 Numerical Evaluation

In this section we report and assess the results for both the model-based and the lake experiment scenarios described in Section 4.2.3.

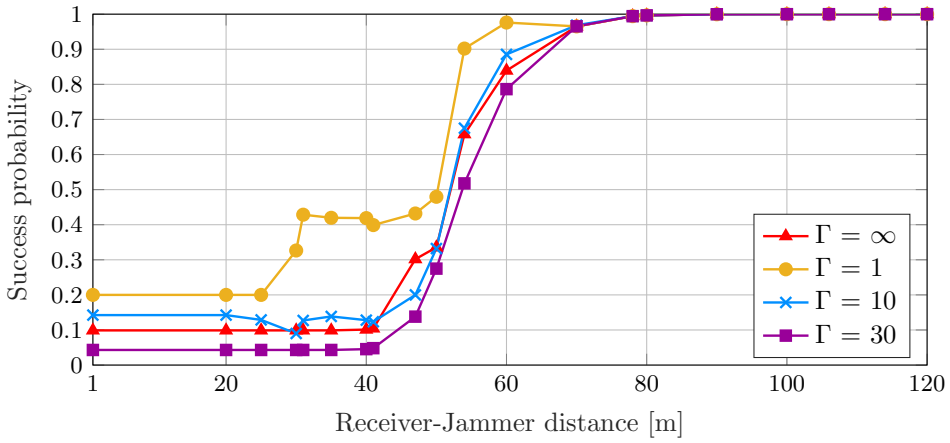


Figure 4.7: Success probability in a single subgame as a function of d_{JR} using the uncoded model, for different values of Γ when $\alpha = 0.4$.

4.2.4.1 Model-based Scenario Results

Based on the position of the jammer, we can distinguish three regions in the underwater area, as shown in Figure 4.6. When the jammer is close to the receiver, any jammed packet is almost surely lost, as the received jamming signal is powerful enough to cause errors in the transmission. In our system scenario, this situation happens when the receiver-jammer distance is less than 40 m. Conversely, when the jammer is far from the receiver, its attack is completely ineffective, as the legitimate signal is much stronger; in our case, this happens when J is farther than 80 m from R. Between these two extremes, an appropriate strategy might significantly improve the performance: it is interesting to investigate how the game evolves in the critical region (where $d_{\mathrm{JR}} \in [40, 80]$ m in our scenario), and which distances yield a successful game for T.

Although this performance figure refers to a specific combination of transmission power and modulation, a different configuration would still lead to the definition of the three regions, but at different distances between the transmitter and the jammer [197].

This partition is also clear from Figure 4.7, which shows the transmission success probability of a subgame as a function of the distance between J and R. The success probability is close to 1 when the jammer is far away, and quickly drops when it gets closer than 80 m. It is interesting to note that the success probability when the jamming node is close decreases for longer time horizons; in this case, T tries to save energy while still transmitting, and a shorter window leads to a more aggressive policy. However, agents with a longer time horizon can avoid suboptimal choices. The jammer is particularly affected by this short-sightedness, as its reward function does not explicitly have a penalty for energy expenditure, and it will waste energy if its horizon is too short, quickly exhausting its own battery. This causes a temporary drop in the success probability, which is quickly reversed when the jammer depletes its battery and tries to fight a lost battle against the transmitter. In fact, a short time

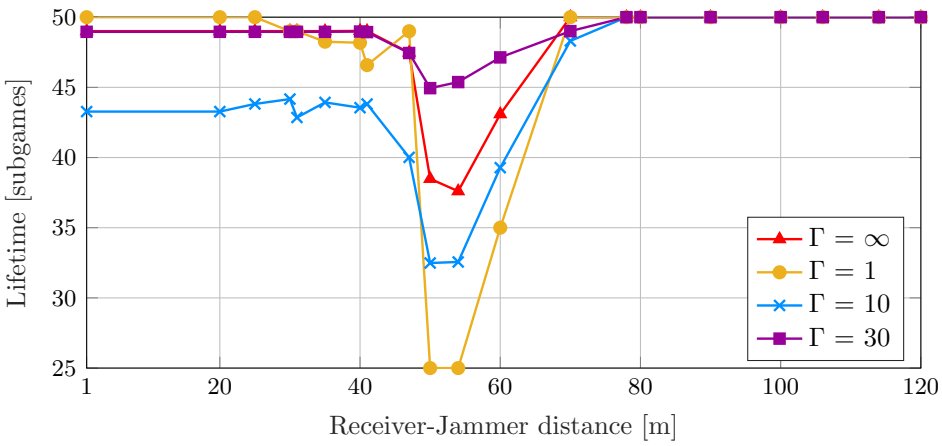


Figure 4.8: Transmitter's lifetime as a function of d_{JR} using the uncoded model, for different values of the time horizon Γ when $\alpha = 0.4$.

horizon corresponds to both a higher success probability and a higher lifetime for the transmitter, as shown in Figure 4.8. Since the initial jammer battery $B_J^{(1)}$ is set to 200 packets, $\Gamma = 30$ is the only value that ensures that the jammer will not act in a myopic way. We remark that in this case, simply switching to a short-term strategy will not benefit the legitimate transmitter: since the long-term result is the NE, choosing any other strategy will decrease its expected payoff even further. It is interesting to note that the infinite horizon jammer also suffers from this issue, since its temporal discount $\lambda = 0.9$ is small enough to make it weigh present rewards more than heavy future losses. For the rest of this analysis, we will consider the scenario in which $\Gamma = 30$.

The aggressiveness of a long-sighted jammer seems to have little effect on the results: as Figure 4.9 shows, lower values of the parameter α correspond to a slightly higher success probability, but the curves are very close. A jammer close to the receiver can reduce the transmission success probability to less than 10%, but the aggressiveness parameter only has a significant impact on the success probability in the critical region. Since $K = 4$ and $B_{T,0} = 200$, the maximum lifetime of T is 50 subgames, and is reached when T does not add any FEC. The minimum lifetime is 25 subgames, in the case in which T always sends $2K$ packets, providing the maximum possible protection to its payload. Figure 4.10 confirms that there is a downside to aggressiveness: more conservative nodes with a higher α have a slightly longer lifetime in the critical region. Naturally, the lifetime is maximized when $d_{\mathrm{JR}} > 80$ m, i.e., when the jammer no longer affects the packet reception. This result holds for each value of α; in this situation, since almost all packets are received correctly, the best strategy for the transmitter is to send exactly K packets, in order to minimize the energy consumption. Naturally, the critical area definition depends on the transmission power and modulation, and its boundaries can be different in other scenarios.

We also note that the lifetime decreases when the jammer is in the critical region, where strategies have a significant impact on the outcome of the game, and transmitters

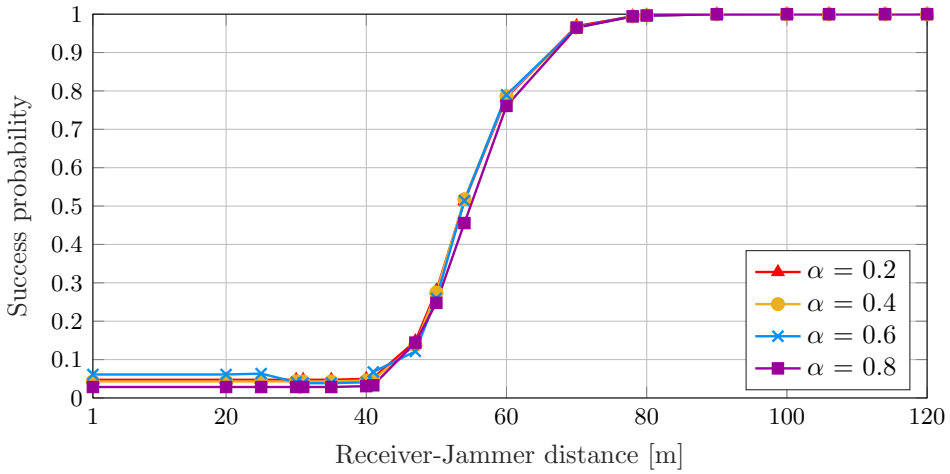


Figure 4.9: Success probability in a single subgame as a function of d_{JR} using the uncoded model, for different values of α when $\Gamma = 30$.

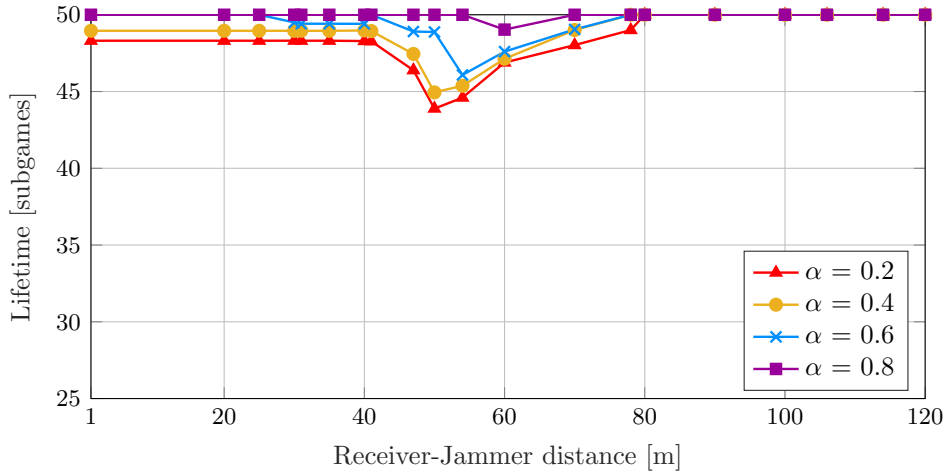


Figure 4.10: Transmitter's lifetime as a function of d_{JR} using the uncoded model, for different values of α when $\Gamma = 30$.

have to behave more aggressively to maximize their payoff. Accordingly, the decrease is far less pronounced for higher values of α and longer time horizons.

We also perform a sensitivity analysis by running a Monte Carlo simulation of this scenario, changing the error probabilities $p_{e_{\mathrm{C}}}$ and $p_{e_{\mathrm{B}}}$ randomly at each run. We set a threshold for the blocked channel error probability, so that it is never lower than the clear channel error probability, and add two independent Gaussian components with zero mean and standard deviation σ to each component. In this case, the choices of the two players become suboptimal, since they are operating with an incorrect model of the environment. Node lifetime is not affected, since the nodes make the same choices,

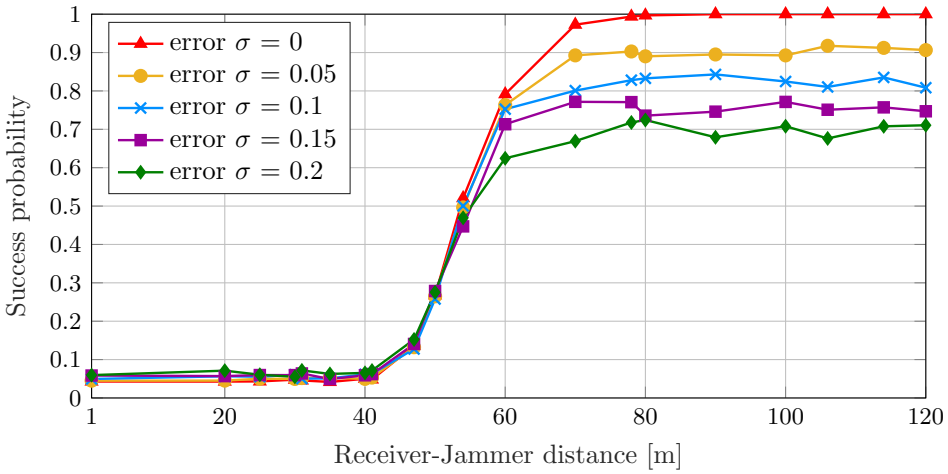


Figure 4.11: Success probabilities for different values of the error standard deviation σ as a function of d_{JR} using the uncoded model, for $\alpha = 0.4$ and $\Gamma = 30$.

but the success probability is, as Figure 4.11 shows. The effect is interesting, and most noticeable outside the critical region: when the jammer is very close to the receiver, the success probability slightly improves as σ grows, while the opposite happens (with much larger effects) when the jammer is far. This might be due to the threshold effect, as the packet error probability cannot be lower than 0 or higher than 1: in this case, the errors are biased. In the critical region, the model error has a slightly negative effect on the success probability, favoring the jammer.

4.2.4.2 Experimental Scenario Results

Figure 4.12 shows the packet error rate measured at different distances in the lake experiment, and compares it to those obtained with the the coded and uncoded LoS channel models. The three curves have a sigmoid-like shape, but the real results have a relatively high packet error rate even when the jammer is far from the receiver. The uncoded packet error rate curve is similar to the measured curve for $d_{\mathrm{JR}} \leq 60$ m, while the coded packet error curve is completely different, as the jammer is already supposed to be completely ineffective at a distance $d_{\mathrm{JR}} \simeq 40$ m. This shows that the channel model we used in the theoretical analysis was extremely optimistic, with a strong LoS component and a very low ambient noise. The real propagation environment is instead much more hostile, and as a result the packet error probability is generally higher (even though the communication system used channel coding), especially in a shallow water scenario akin to the one experienced during the lake experiment.

In this scenario, we consider the lake experiment curve as the real packet error rate, and test players that devise their strategies according to different internal models: since it would be impractical to perform sea trial scenarios before deployment, nodes may have to be trained based on a theoretical model, but the difference between the

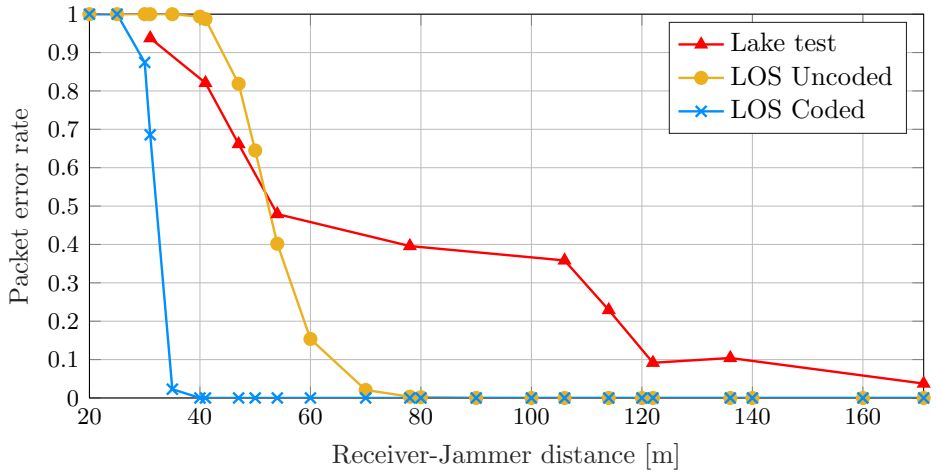


Figure 4.12: Blocked channel packet error rate p_{e_B} for different channel models as a function of d_{JR}.

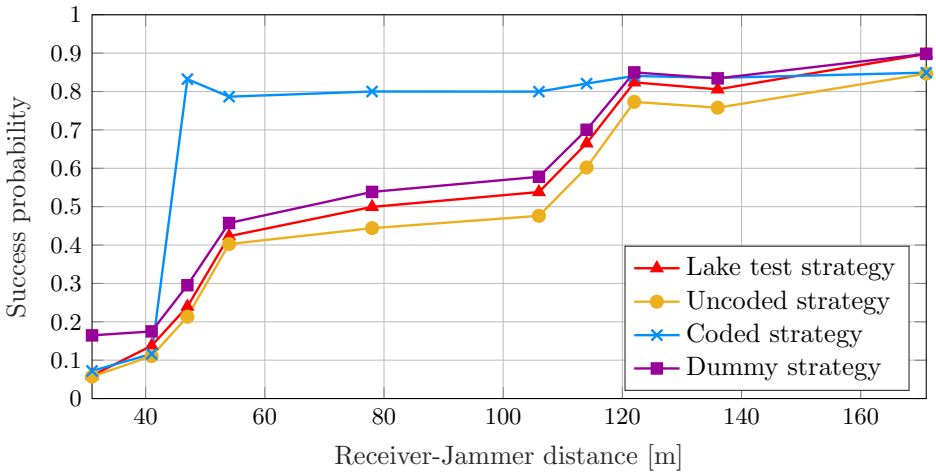


Figure 4.13: Success probability for different strategies as a function of d_{JR} in the lake scenario, for $\alpha = 0.4$ and $\Gamma = 30$.

models and reality may have effects on the performance, which we will analyze in the following. We also consider a dummy jammer which always jams $K + 1$ slots, allowing the transmitter to find the best response: since the dummy strategy is not necessarily a best response, it is worse for the jammer than the NE solution. This case is used as a baseline, since it is the most favorable for the transmitter.

Figure 4.13 shows how using a very optimistic model of the packet error probability leads to an unbalanced scenario: the jammer, convinced that its actions will have little or no effect, saves energy by limiting its transmissions. Most of the time the transmitter has a free channel and just has to contend with the ambient noise. The players using

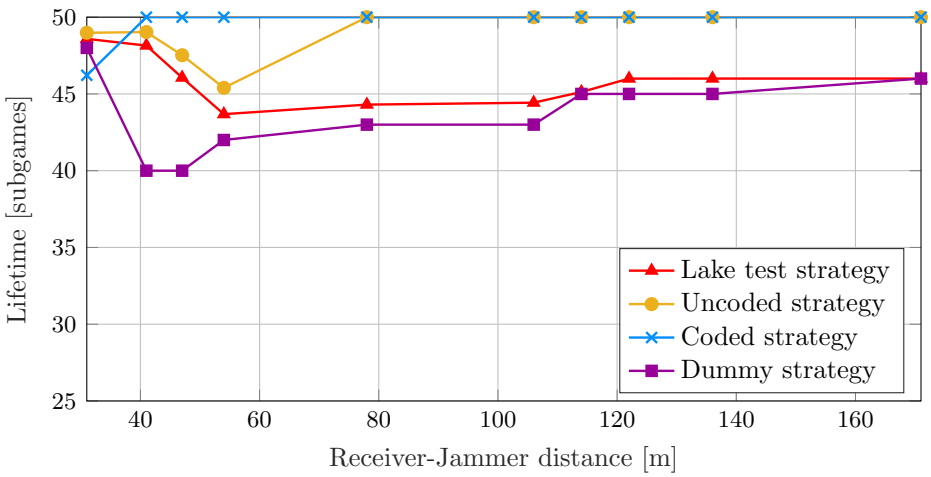


Figure 4.14: Transmitter's lifetime for different strategies as a function of d_{JR} in the lake scenario, for $\alpha = 0.4$ and $\Gamma = 30$.

the uncoded model, which is much closer to the real packet error probability curve, reach a similar equilibrium. Finally, the dummy jammer strategy is actually close to optimal at long distances, while it allows much more data to get through when the jammer is close.

In general, the transmitter almost always chooses a conservative strategy, as Figure 4.14 shows. Since the correct packet error probabilities are higher than those predicted by the models, and particularly the coded one, the lifetime of the node with the correct model decreases as it transmits slightly more redundancy. This is also true for the dummy jammer case, as its strategy of jamming $K + 1$ slots in each subgame is quite aggressive.

4.3 Bayesian Analysis for Acoustic Blind Jamming

In this section we analyze the same scenario described in Section 4.2 but relaxing the complete information assumption. Indeed, we consider the distance between the jammer and the receiver to be unknown to the transmitter. The transmitter will try to infer this information based on the outcome of the transmissions, updating its belief on the distance at each subgame.

The theoretical model of the subgame and the full game is the same of the complete information scenario and it is described in Section 4.2.1. In the following we described the bayesian part of the game. The notation and its meaning are described in Table 4.1

4.3.1 The Bayesian Jamming Game

We now relax the complete information assumption and consider an incomplete information game, in which the position of the jammer is unknown to the transmitter.

In each subgame, we derive the Bayesian Nash Equilibrium (BNE) [198] mixed strategies $(\Phi_T^\star, \Phi_J^\star)$ of the two rational players using the Lemke-Howson algorithm [194]. The outcome of the subgame depends on such strategies and on the stochastic channel conditions. When playing the subgame, T obtains some information, denoted as $\mathbf{f}$, about the outcome of the game. This feedback is used by T to update its estimate of J's position. This is repeated until the battery of the transmitter is depleted.

4.3.1.1 Computing the Expected Payoff

We now consider the expected payoff of the players in subgame m for a given set of strategies. As in Section 4.2.2, we need to compute the expected payoff $\mathbb{E}[u_i|N_T, N_J]$ from the quantity $\mathbb{E}[\chi|N_T, N_J]$, $i \in \{T, J\}$, where χ is an indicator term for transmission success as defined in Section 4.2.1.1. In this Section we also need to consider the dependency to the quantity d_{JR} that is unknown to the transmitter. Considering the number of transmitted packets N_T, the number of jammed slots N_J, the number of packets transmitted in a clear and a blocked channel, N_C and N_B, using the law of total probability, for node T we have:

$$\mathbb{E}[\chi|N_T, N_J] = \sum_{N_B=0}^{N_T-1} \mathbb{E}[\chi|N_B]\, \mathrm{P}(N_B|N_T, N_J)\,. \tag{4.20}$$

Where we considered the fact that the first packet cannot be jammed, as in Section 4.2.2.1. The first term inside the summation is the expectation of a subgame success, given the number of packets successfully delivered and jammed during that subgame and the distance between the jammer and the receiver. We can then distinguish between two cases: for the N_C packet transmissions over a clear channel, the packet error probability is equal to p_{e_C}, which is a function of the SNR. For the N_B transmissions that are disturbed by the jammer, the packet error probability is $p_{e_B}(d_{JR})$, which is a function of the SINR and depends on the jammer's position. The specific values of the two probabilities can be derived for each modulation, and Section 4.2 gives the complete results for DQPSK. Using these probabilities, $\mathbb{E}[\chi|N_B, d_{JR}]$ can be expressed as:

$$\begin{aligned}\mathbb{E}[\chi|N_B, d_{JR}] = \sum_{r=K}^{N_T} \sum_{r_C=\max(0,r-N_B)}^{\min(r,N_C)} \binom{N_C}{r_C} (1-p_{e_C})^{r_C} \\ p_{e_C}^{N_C-r_C} \binom{N_B}{r-r_C} p_{e_B}(d_{JR})^{N_B-(r-r_C)} \left(1-p_{e_B}(d_{JR})\right)^{r-r_C}.\end{aligned} \tag{4.21}$$

Similarly than Equation (4.6), we split r between packets that are delivered over a clear channel, i.e., $r_C \leq r$, and those which are delivered over a jammed channel, i.e., $r_B = r - r_C \leq N_B$. However, in this case we are making explicit the dependency on the distance d_{JR}.

Since T does not have complete knowledge of d_{JR}, but can only infer it from the feedback it receives, we can define its belief distribution $p(\hat{d}_{JR})$, $\hat{d}_{JR} \in \mathcal{D}$ where $\mathcal{D}$

is the discrete set of possible distances. The distribution represents the estimate of the distance: before the first subgame, the prior belief distribution is uniform, but is updated with the feedback obtained from the receiver after each subgame, reducing the uncertainty. Using the law of total probability, we can remove the condition on d_{JR} to obtain the first part of the sum in (4.20):

$$\mathbb{E}\left[\chi|N_B\right] = \sum_{\hat{d}_{JR}\in\mathcal{D}} \mathbb{E}\left[\chi|N_B, d_{JR}\right] p(\hat{d}_{JR}). \tag{4.22}$$

Finally, the second part of the sum in (4.20) can be computed with Equation (4.12) since the same condition that the first transmitted packet cannot be jammed holds. As explained in Section 4.2.2.1, we assume that both T and J choose the slots according to a uniform distribution, since this is the optimal choice for an anti-coordination game.

4.3.1.2 Finding the BNE

We now derive the BNE solution of the game, i.e., the pair of strategies used by rational players. We consider mixed strategies, as each strategy $\mathbf{\Phi}_i^{(m)}$ is a probability distribution over $\mathcal{N}_i^{(m)}$. The BNE strategies are *mutual best responses*, i.e., each strategy maximizes the player's payoff if the other player uses the other, given the knowledge about its state:

$$\begin{aligned}\mathbf{\Phi}_T^*(\mathbf{\Phi}_J, \mathbf{B}, p(\hat{d}_{JR})) = \arg\max_{\mathbf{\Phi}_T} \sum_{\hat{d}_{JR}\in\mathcal{D}} \sum_{N_T\in\mathcal{N}_T} \sum_{N_J\in\mathcal{N}_J} \\ \Phi_T(N_T)\Phi_J(N_J|\hat{d}_{JR})p(\hat{d}_{JR})\mathbb{E}\left[U_T|N_T, N_J, \mathbf{B}, \hat{d}_{JR}\right]\end{aligned} \tag{4.23}$$

$$\begin{aligned}\mathbf{\Phi}_J^*(\mathbf{\Phi}_T, \mathbf{B}) = \arg\max_{\mathbf{\Phi}_J} \sum_{N_T\in\mathcal{N}_T} \sum_{N_J\in\mathcal{N}_J} \Phi_T(N_T)\Phi_J(N_J) \\ \mathbb{E}\left[U_J|N_T, N_J, \mathbf{B}, d_{JR}\right].\end{aligned} \tag{4.24}$$

where $\mathbf{B} = [B_T, B_J]$. The BNE is then given by the pair of strategies that satisfies the following condition:

$$\begin{cases} \mathbf{\Phi}_T^* = \mathbf{\Phi}_T^*(\mathbf{\Phi}_J^*, \mathbf{B}, p(\hat{d}_{JR})) \\ \mathbf{\Phi}_J^* = \mathbf{\Phi}_J^*(\mathbf{\Phi}_T^*, \mathbf{B}) \end{cases} \tag{4.25}$$

The expected long-term payoffs can be computed by using the dynamic programming procedure described in Section 4.2.2.2. By using (4.22), we can obtain the expected value of the payoff U_T for a given belief $p(\hat{d}_{JR})$:

$$\mathbb{E}\left[U_T(\Gamma)|N_T, N_J\right] = \sum_{\hat{d}_{JR}\in\mathcal{D}} \mathbb{E}\left[U_T|\hat{d}_{JR}\right] p(\hat{d}_{JR}). \tag{4.26}$$

Naturally, since J has complete knowledge of its position, its expected payoff can be computed using the real value of d_{JR}:

$$\mathbb{E}\left[U_J(\Gamma)|N_T, N_J\right] = \mathbb{E}\left[U_J|d_{JR}\right]. \tag{4.27}$$

Using the two matrices derived from (4.26) and (4.27), we have the bimatrix form of the game and can use the well-known Lemke-Howson algorithm [194] to find the BNE [198].

In (4.26) and (4.27), we do not consider the additional knowledge that T will gain in future steps, and therefore make players more myopic and limit the analysis to a one-step approximation. In general, this puts the jammer at a slight disadvantage, since a more foresighted player might try to make intentionally suboptimal moves to confuse the transmitter. However, this simplification was necessary to obtain a closed-form solution.

4.3.1.3 Updating Beliefs

We consider that the feedback $\mathbf{f} = (r_B, r_C, N_J)$ is available to the transmitter node, as it can be sent as part of the acknowledgment packet by the receiver. The receiver specifies the number of packets delivered correctly over a clear and jammed channel, denoted as r_C and r_B, respectively. The transmitter also knows its own move N_T. If we assume that the reception of each packet is independent of the others, we obtain:

$$p(r_B|d_{JR}, N_B) = p_{e_B}(d_{JR})^{N_B - r_B}\,(1 - p_{e_B}(d_{JR}))^{r_B} \tag{4.28}$$

$$p(r_C|N_B) = p_{e_C}^{N_T - N_B - r_C}\,(1 - p_{e_C})^{r_C} \tag{4.29}$$

$$p(r_C, r_B|d_{JR}, N_B) = p(r_C|N_B)p(r_B|d_{JR}, N_B). \tag{4.30}$$

If we combine (4.30) with (4.12), we obtain:

$$p(r_C, r_B|d_{JR}, N_T, N_J) = \sum_{N_B = r_B}^{N_T - \min(r_C, 1)} p(r_C, r_B|d_{JR}, N_B)p(N_B|N_T, N_J). \tag{4.31}$$

We also know the optimal strategy $\Phi_J^*(N_J|d_{JR})$ for all values of d_{JR}, since we assume that both nodes are rational players in the game-theoretic sense. In this case, we can also calculate:

$$p(N_J|d_{JR}) = \Phi_J^*(N_J|d_{JR}). \tag{4.32}$$

We can now define the probability of making observation $\mathbf{f}$ given that the jammer is at distance d_{JR}:

$$p(\mathbf{f}|d_{JR}) = p(r_C, r_B|d_{JR}, N_T, N_J)p(N_J|d_{JR}). \tag{4.33}$$

We can now apply Bayes' theorem to obtain:

$$p(d_{JR}|\mathbf{f}) = \frac{p(\mathbf{f}|d_{JR})p(d_{JR})}{p(\mathbf{f})} = \frac{\Phi_J^*(N_J|d_{JR})p(r_C, r_B|d_{JR}, N_T, N_J)p(d_{JR})}{\sum_{d\in\mathcal{D}} p(d)\Phi_J^*(N_J|d)p(r_C, r_B|d, N_T, N_J)}. \tag{4.34}$$

By computing the *a posteriori* probability for all values of d_{JR} in $\mathcal{D}$, T can update its belief for the next round.

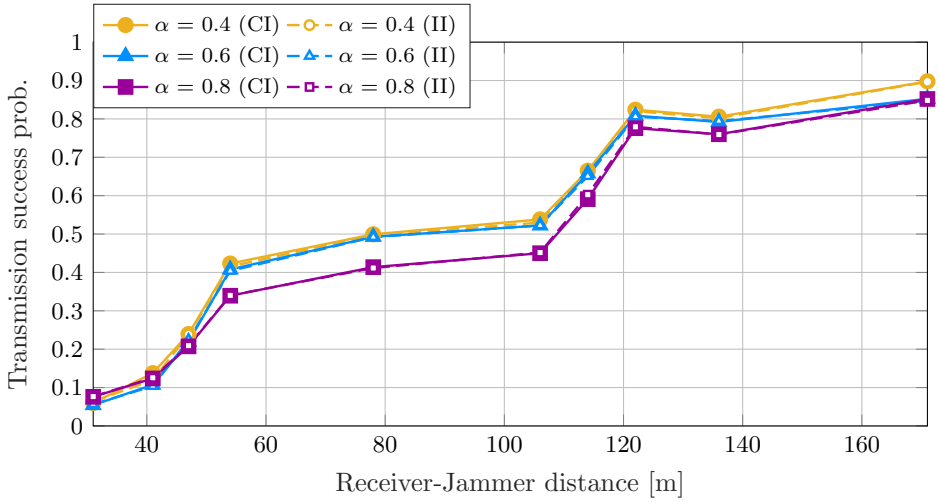


Figure 4.15: Subgame success probability as a function of d_{JR}, for different values of α when $\Gamma = 30$.

4.3.2 Numerical Evaluation

We evaluate the performance of the optimal strategies by studying the energy consumption and the PDR of T, comparing the results obtained in the complete information (CI) scenario with those of the incomplete information (II) scenario.

To compute the optimal strategies for both the CI and II scenarios, we considered a packet error probability derived from a measurement campaign we performed in the Garda lake on October 17th, 2019 and described in Section 4.2.3.2. The obtained Packet Error Rate (PER) of a jammed packet as a function of the distance is depicted in Figure 4.12 (solid red line) and the employed parameters summarized in Table 4.2.

The packet error probability is used to find the optimal strategies for both T and J. Since the feedback affects future beliefs in the II scenario, we ran a Monte Carlo simulation with 1000 trials to compute the performance of T and J. As in the previous section, in the simulated scenario we set the number of information packets in each subgame to $K = 4$ and the initial battery charge to $B_i^{(0)} = 200$, $i \in \{T, J\}$.

4.3.2.1 Simulation Results

As we discussed in Section 4.2, the packet error probability when the jammer is active has a strong effect on the outcome of the game. The aggressiveness of the players, tuned by the parameter α, also comes into play, as higher values of α (which correspond to more conservative players, i.e., players that try to save as much energy as possible, despite the low packet reception probability) translate into a longer lifetime and a lower success probability. Figure 4.15 shows the average subgame success probability as a function of the distance d_{JR}, for both the CI and II cases, and confirms this result in the lake experiment scenario. It is easy to notice that the curves for the II scenario closely follow those for CI: since the receiver's feedback allows to quickly identify the

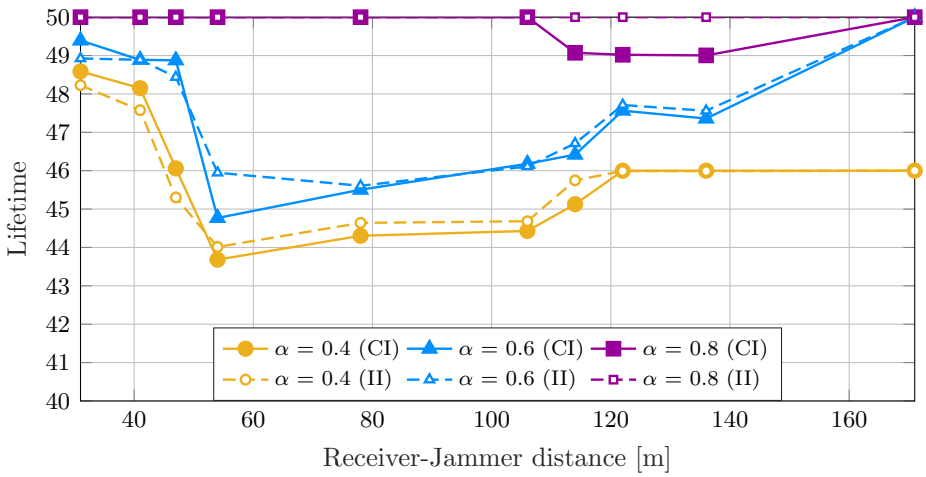


Figure 4.16: Transmitter's lifetime as a function of d_{JR}, for different values of α when $\Gamma = 30$.

real position of the jammer, the transmitter can start with no information about the jammer and still obtain almost the same results.

Figure 4.16 shows that the same holds for the battery life of the transmitter: the transmitter is always very conservative, with a lifetime between 43 and 50 subgames, as the maximum possible lifetime with the initial battery settings is 50 subgames, while a full duplication of each data burst would deplete the transmitter's battery in 25 subgames. There is a slight difference between the curves for the CI and II scenarios, where for some distances the CI scenario is slightly better. However, the difference in the average lifetime (measured in subgames) is smaller than 1 at all possible distances. We also note that the lifetime is lower when the packet error probability for jammed packets is close to 0.5: in this case, some redundancy can highly improve the transmitter's chances to correctly send its data burst, as adding one packet maximally increases $\mathbb{E}[\chi]$ as given by (4.22). Active defense is then more effective.

Figure 4.17 shows a summary of the trade-off between lifetime and success probability at different distances: in all cases, a slight gain in the success probability comes at the cost of a comparable lifetime reduction. The II transmitter is very similar to the CI one, and even outperforms it in some cases: the trend confirms that the performance is not significantly affected by the asymmetric knowledge.

Similarly as in previous section, we also performed a sensitivity analysis on the packet error probabilities: we ran Monte Carlo simulations with a Gaussian noise on both p_{e_B} and p_{e_C} (with the limiting conditions that no error probability can be below 0 or above 1, and $p'_{e_C} \leq p'_{e_B}$ in any case):

$$p'_{e_C} = \max\left(0, \min\left(1, p_{e_C} + v\right)\right) \tag{4.35}$$

$$p'_{e_B} = \max\left(p'_{e_C}, \min\left(1, p_{e_B} + w\right)\right), \tag{4.36}$$

where v and w are normally distributed independent random variables with zero mean

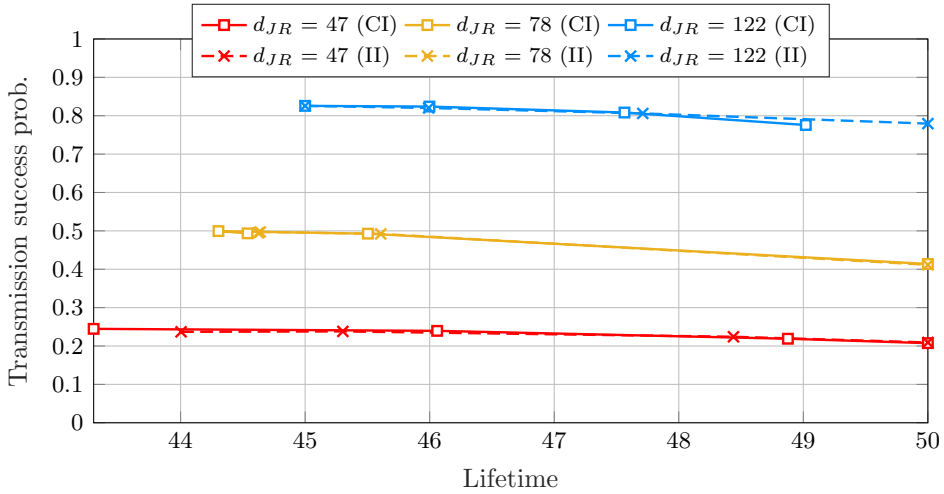


Figure 4.17: Success probability in a single subgame vs. lifetime, when $\Gamma = 30$, for different d_{JR} values, and varying α: each point of the same curve corresponds to a different value of α, ranging from 0.2 to 0.8. Lower values of α result in a lower lifetime.

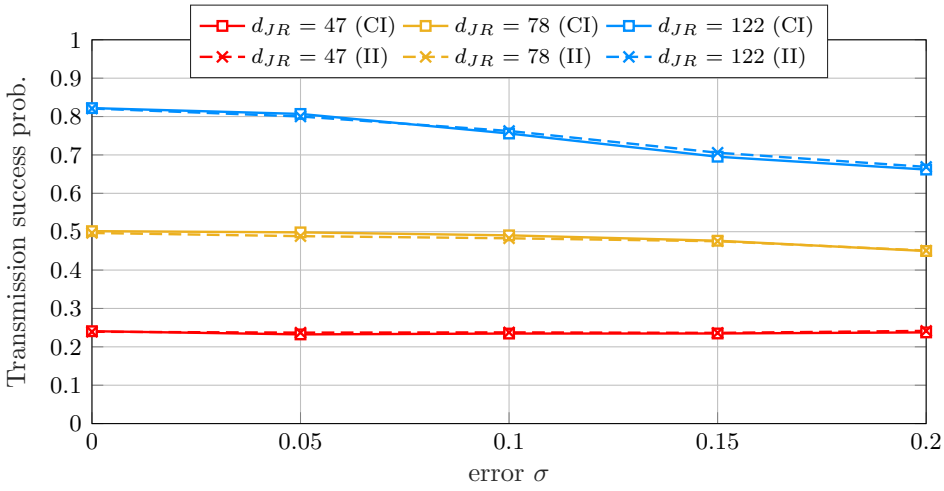


Figure 4.18: Success probability in a single subgame as a function of the error standard deviation σ, for different d_{JR} values, when $\alpha = 0.4$ and $\Gamma = 30$.

and standard deviation σ. As Figure 4.18 shows, a larger noise on the packet error probability reduces the overall success probability, with a larger effect if the distance between the jammer and the receiver is increased, but the performance of the II system is still indistinguishable from that with CI. The lifetime is not pictured, as it is almost unaffected by the noise on the packet error probability, and the differences with respect to Figure 4.16 are negligible.

4.4 Blind vs. Reactive Jamming: a Geometrical Analysis

In this section, we enhance the analysis on jamming by considering both reactive and blind jamming and by investigating the effect of active and evasive defense (differently from Sections 4.2 and 4.3 we also analyze the MCS adaptation as possible active countermeasure). We define a game theoretical model to address the selection of the best defense, which is typically a trade-off among the network geometry, the conducted attack and the available resources. Indeed, due to the peculiarities of underwater acoustic propagation, finding the optimal strategies against different jammer models becomes non-trivial and, to the best of our knowledge, has never been investigated before. The proposed game theoretical analysis, when carried out before an actual deployment, helps identify the network vulnerabilities, by discovering the critical areas where a reactive jammer can cause severe network disruption. This type of analysis is unique for underwater acoustic networks and does not apply in the usual deployment of a terrestrial radio-frequency network, where the trade-off becomes trivial in favor of a reactive jammer. In addition, the game theoretical framework helps predict how many packets an attacked node can send to the destination before depleting its battery. The proposed solution has been validated via simulations employing real acoustic data, recorded during the CMRE Littoral Acoustic Communications Experiment 2017 (LACE17) [2] sea trial in the Gulf of La Spezia, Italy. More specifically, BER measurements recorded during LACE17 have been used in our simulation to model the quality of the communication link. Although LACE17 was not conducted to model a jamming attack, the data can be adapted to our scenario with limited assumptions: the different transmission power and SINR settings explored in the dataset were taken to represent a communication channel with and without the presence of a jamming signal. In the LACE17 dataset, the jamming signal is modeled as an Additive White Gaussian Noise (AWGN) noise: however, it is possible for a jammer to use a different modulation, which might have a better chance of disrupting the communication [119]. However, this is irrelevant to the design of our model, as it would only affect the bit error probability of a jammed packet, changing the resulting strategies but not the procedure to solve the game. Our simulation results show that the geometry of the underwater scenario is a critical factor in determining the optimal jamming and defense strategies in our scenario, including evasive as well as active countermeasures.

4.4.1 Game Theoretical Model

Also in this section, the analyzed scenario is the same described in Section 4.2.3 where T sends un update of K packets of L_0 bits, while J tries to block the communication. However, differently from the previous sections, in this case we consider also the geometry of the network in the jamming analysis. The main notation used in this section is summarized in Tables 4.1 and 4.3. We define θ, the angle between the segment between R and T and the segment between R and J: if θ is 0 and $d_{JR} \leq d_{TR}$, the jammer is between the transmitter and the receiver, while if $\theta = \pi$, J and T are on opposite sides of the receiver. Fig. 4.19 displays different scenarios when θ varies from 0 to π and

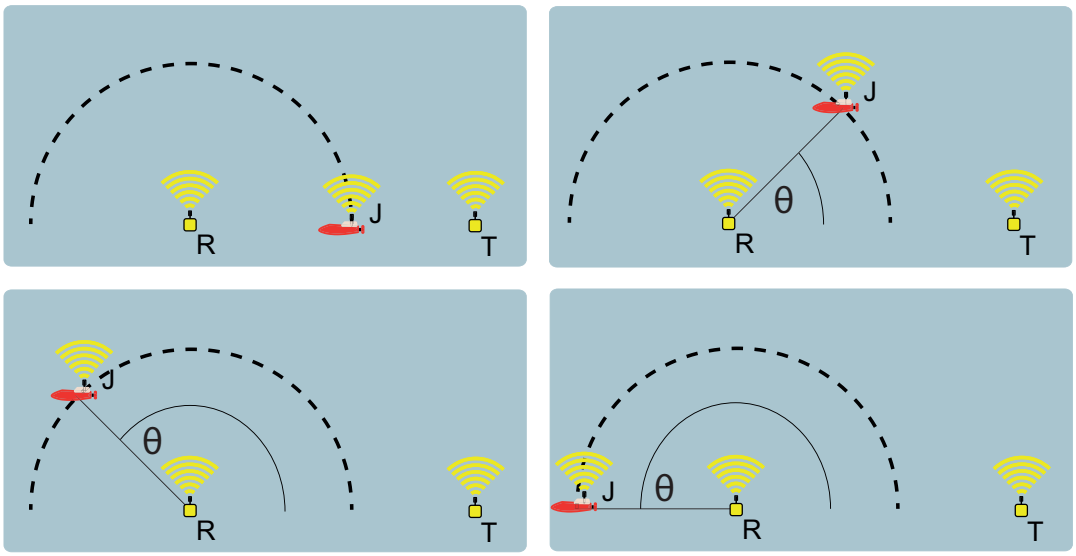


Figure 4.19: Example of analyzed topologies for the reactive and blind jammer scenario with different angle θ ($\theta = 0$ top-left; $\theta = \pi/4$ top-right; $\theta = 3\pi/4$ bottom-left; $\theta = \pi$ bottom-right). In this example $d_{JR} = d_{TR}/2$.

with $d_{JR} = d_{TR}/2$.

In the following, we also consider in the game model a wider set of strategies with respect to that employed in Sections 4.2 and 4.3. We assume that the transmitter can select the MCS $M_T \in \mathcal{M}$, the transmission power $P_T \in \mathcal{P}$, and the number of encoded messages $N_T \in \mathcal{N}$ (with $\mathcal{N} = \{K, K+1, \ldots, N_{\max}\}$) for each update. The set of possible configurations is then $\mathcal{M} \times \mathcal{P} \times \mathcal{N}$, and the specific selection depends on the target scenario. We assume that J always transmits at the same power. A reactive jammer can choose the number of sensed packets that it will jam ($N_J \leq N_T$), as it is able to sense incoming packets and choose whether to jam all or just a fraction of them. On the other hand, the blind jammer does not know how many packets will be transmitted, or in which time slots, and as such, the only decision it can make is the number of communication resources, i.e., time slots, over which it will send the jamming signal ($N_J \leq N_{max}$). As before, we assume a slotted time model, which is favorable to a blind jammer, as it will be able to block packets completely. Indeed, since the node positions are known to both T and J, we assume J to be able to synchronize its slots in order to be able to jam the whole packet. This is not entirely realistic, as the jammer might not be able to know the position of the other nodes precisely, or might suffer from drift due to currents and imprecise positioning; we do not handle this directly in our model, but we perform a sensitivity analysis showing that the strategies are robust to positioning errors.

As described in Section 4.2.1, we consider a zero-sum multistage game between T and J, in which each stage represents the transmission of an update. We consider the two nodes' batteries to be quantized in terms of the maximal common divisor of the

Table 4.3: Notation for geometrical analysis.

Symbol	Meaning	Symbol	Meaning
L_0	Length of a packet in bits	θ	Angle between segments RT and RJ
$\mathcal{M}$	Set of MCSs	$\mathcal{P}$	Set of transmission power levels
$\mathcal{N}$	Set of packet-level coding choices	$p_{bit,C}(M_T, P_T)$	Clear channel BER
$p_{bit,B}(M_T, P_T)$	Jammed channel BER	$B_i^{(0)}$	Initial battery level for node i
$\mathcal{S}_{\text{re}}$	Set of reactive jammer moves	Δ_d	Delay for the jamming signal
$F(\theta, M_T)$	Fraction of a packet affected by reactive jamming	$R_b(M_T)$	Bitrate for MCS M
$\psi(M_T)$	Maximum number of bit errors for MCS M	$p_e(P_T, M_T, F)$	Packet error probability
$\mathcal{S}_{\text{bl}}$	Set of blind jammer moves	$\mathcal{B}$	Set of battery states
T_0	Transmission time for a packet with Binary Phase-Shift Keying (BPSK)	τ_h	Switching time for half-duplex modems
E_q	Energy quantum	$E_{\text{tx}}(M_T)$	Energy consumption for a packet

possible packet energy costs, so that the state can be discretized. The two nodes start from battery levels $B_T^{(0)}$ and $B_J^{(0)}$, and keep playing until the transmitter's battery does not allow it to send any more updates.

Given the wider set of strategies considered in this section, in each subgame m, the transmitter will choose $M_T^{(m)}$, $P_T^{(m)}$, and $N_T^{(m)}$, and the jammer will select its jamming strategy. The value of $M_T^{(m)}$ is known in advance to the jammer, as we assume that T sends a control message to R before the beginning of the update, which the jammer can listen to. This assumption allows us to also relax the hypothesis that the a blind jammer cannot jam the first transmitted packets. Of course, the transmission of a control message may not be true for all underwater networks, but we consider this assumption to be a worst-case scenario for transmission. In some networks, encrypted information may be exchanged between legitimate transmitter and receiver. If the jammer has less information, the scenario becomes easier for the defender, tilting the game its way and improving the transmission performance. However, there are cases where this information is sent in clear before the data transmission. This is true for both proprietary protocols implemented by underwater acoustic modem manufacturers [199, 200] or solutions [201, 202] where the JANUS standard [203] is used for coordination. In both these cases, control packets can be easily overheard and exploited by the attacker to make the jamming more successful. The reward $u_T^{(m)}$ for the transmitter is 1 if the transmission is successful, i.e., if at least K of the $N_T^{(m)}$ packets are received and decoded correctly, and 0 otherwise. This is equivalent to set $\alpha = 0$ in Equation (4.1) [1]. As the game is purely adversarial, the reward for the jammer is $u_J^{(m)} = -u_T^{(m)}$. After the subgame, the battery levels of the two nodes are decreased for the next subgame, to take the energy consumption into account.

As in the previous sections, in order to model the long-term consequences of energy consumption, and to encourage the nodes to maximize their lifetime, we model the payoff of the users as the expected reward for the next Γ subgames. The players' payoffs in the multistage game can be computed as in Equation 4.4. In the following, we compute the expected payoff for the reactive and blind jammer separately, considering

[1] differently from Sections 4.2 and 4.3 we do not consider the energy consumption of the transmitter in the payoff computation. However, it is still intrinsically taken into account in the evolution of the game.

the effect of the scenario geometry on their performance.

4.4.1.1 Expected Payoff with Reactive Jamming

If the jammer is reactive, there could be a partial overlap between the transmitted packet and the jamming signal. As we described above, T, J, and R form a triangle. The distance between T and R is d_{TR}, the distance between J and R is d_{JR}, and the angle at the receiver is θ. In this case, the jammer's action is the choice of the number of packets to jam, which we denote as $N_J \in \mathcal{S}_{\text{re}}$, with $\mathcal{S}_{\text{re}} = \{0, 1, \ldots, N_{\max}\}$.

The overlap between the packet and the interference depends on the relative position between transmitter, jammer and receiver. Indeed, the jammer needs to overhear the packets sent by the transmitter before starting to jam the signals and then the jamming signal needs to cover the distance between the jammer and the receiver. The delay Δ_d after which the interference from the jammer arrives at the receiver, computed with respect to the transmitted packet, is equal to

$$\Delta_d(\theta) = \frac{d_{TJ} + d_{JR}}{c} - \frac{d_{TR}}{c} + \tau_h, \tag{4.37}$$

where c is the speed of sound, and τ_h is the time required for the jammer's half-duplex model to switch from receiving to transmitting. In our scenario d_{JR} and d_{TR} are fixed and known while d_{TJ} depends on θ. Carnot's theorem can be used to find the distance d_{TJ}

$$d_{TJ}^2 = d_{TR}^2 + d_{JR}^2 - 2 d_{TR} d_{JR} \cos\theta. \tag{4.38}$$

We define $F(\theta, M_T)$ as the portion of a packet affected by the jamming signal, with $F \in [0, 1]$ depending on the modulation and on the position of the three nodes. Its value is therefore

$$F(\theta, M_T) = \max\left(0, 1 - \frac{\Delta_d(\theta) R_b(M_T)}{L_0}\right), \tag{4.39}$$

where $R_b(M_T)$ is the bitrate of the selected MCS. A more robust modulation will reduce the rate, decreasing the effectiveness of the jamming signal, but increasing the overlap at the same time. We define the BERs for a clear channel and for a jammed channel with MCS M_T and transmission power P_T as $p_{bit,C}(P_T, M_T)$ and $p_{bit,B}(P_T, M_T)$, respectively. As the MCS includes a channel code, packets are protected from errors as long as the number of flipped bits does not go over the correction capability of the code, which we denote as $\psi(M_T)$ and corresponds to half of the code's minimum Hamming distance [204]. Based on the overlap between the transmission and the jamming signal, we get the packet error probability $p_e(P_T, M_T, F)$:

$$\begin{aligned} p_e(P_T, M_T, F) = 1 - & \sum_{e_B=0}^{\min(\psi(M_T), FL_0)} \text{Bin}(e_B; FL_0, p_{bit,B}(P_T, M_T)) \\ & \sum_{e_C=0}^{\min((1-F)L_0, \psi(M_T)-e_B)} \text{Bin}(e_C; (1-F)L_0, p_{bit,C}(P_T, M_T)), \end{aligned} \tag{4.40}$$

where $\mathrm{Bin}(k; N, p)$ is the binomial probability mass function, defined as:

$$\mathrm{Bin}(k; N, p) = \binom{N}{k} p^k (1-p)^{N-k}, \quad 0 \leq k \leq N. \tag{4.41}$$

We can then compute the probability that at least K of the N_T packets are correctly received, given that the jammer jams N_J of them (with $N_J \leq N_T$ in the reactive jamming scenario):

$$\begin{aligned}\mathbb{E}_{\mathrm{re}}[u_T | M_T, N_T, N_J] = \sum_{r_B=0}^{N_J} \mathrm{Bin}(r_B; N_J, p_e(P_T, M_T, F(M_T, \theta))) \\ \sum_{r_C=K-r_B}^{N_T-N_J} \mathrm{Bin}(r_C; N_T - N_J, p_e(P_T, M_T, 0)).\end{aligned} \tag{4.42}$$

The $N_T - N_J$ packets that are not jammed can be considered as having zero overlap with the jamming signal.

4.4.1.2 Expected Payoff with Blind Jamming

In this case, we assume that there are $N_{\max}$ time slots, and that the blind jammer can perfectly synchronize with the packets. In this case, the optimal action for T is to use random slots to send its packets, and the possible actions for J are, again, the number of jammed slots, which we also denote as $N_J \in \mathcal{S}_{\mathrm{bl}}$, with the same set $\mathcal{S}_{\mathrm{bl}} = \{0, 1, \ldots, N_{\max}\}$. As we assume synchronization, the error probability for jammed packets is $p_e(P_T, M_T, 1)$. We can then compute the probability mass function (pmf) of the number of packets N_C that are transmitted without interference from J, which follows a hypergeometric distribution as proven by Vandermonde's Identity [205]:

$$p(N_C | N_T, N_J) = \frac{\binom{N_T}{N_C}\binom{N_{\max}-N_T}{N_J-(N_T-N_C)}}{\binom{N_{\max}}{N_J}}. \tag{4.43}$$

This equation differs from Equation (4.12) since we now assume the blind jammer can also jam the first slot by exploiting the control messages sent before the beginning of the update (see Section 4.4.1). Once we know the number of packets without interference, we can compute the expected payoff for a given move N_J by the blind jammer:

$$\begin{aligned}\mathbb{E}_{\mathrm{bl}}[u_T | M_T, N_T, N_J] = \sum_{N_C=0}^{N_T} p(N_C | N_T, N_J) \sum_{r_B=0}^{N_T-N_C} \mathrm{Bin}(r_B; N_T - N_C, p_e(P_T, M_T, 1)) \\ \sum_{r_C=K-r_B}^{N_C} \mathrm{Bin}(r_C; N_C, p_e(P_T, M_T, 0)).\end{aligned} \tag{4.44}$$

4.4.2 Analytical Solution of the Game

As in Section 4.2.2.2, in the case of complete information, dynamic programming can be used to determine the NE. The system state can be completely represented by the tuple (B_T, B_J), as the battery evolution is the only change in this scenario. The state space is limited by the initial battery levels, so the initial state is $(B_T^{(0)}, B_J^{(0)})$. The full state space is $\mathcal{B} = \{0, \dots, B_T^{(0)}\} \times \{0, \dots, B_J^{(0)}\}$. The payoff is then computed by considering the next Γ subgames, during which the system state will move to progressively lower values as the two nodes deplete their batteries. If the transmitter does not have enough energy to transmit the update with any MCS, the game is over. We can aggregate all states that satisfy the ending condition into a final state ε and define its payoff as in (4.13). We can then recursively compute the payoff for each state, starting from the base case in the final state and moving gradually upwards. The recursive formula for the payoff is similar to Equation 4.14, but it needs to consider also the wider strategy choice of the transmitter, therefore the equation is:

$$\begin{aligned} U_i(\Gamma|B_T, B_J) =& \mathbb{E}[u_i|B_T, B_J] + \\ & \lambda \sum_{B'_T=0}^{B_T} \sum_{B'_J=0}^{B_J} U_i(\Gamma - 1|B'_T, B'_J) p(B'_T, B'_J|B_T, B_J), \quad i \in \{T, J\}, \end{aligned} \tag{4.45}$$

where the transition probability $p(B'_T, B'_J|B_T, B_J)$ depends on the players' choices. We define a strategy Φ_T as a probability distribution over T's action space, and do the same for Φ_J. For a given set of strategies Φ_J, the transition probability is:

$$\begin{aligned} p\left(B'_T, B'_J|B_T, B_J\right) = & \sum_{(M_T, P_T, N_T) \in \mathcal{M} \times \mathcal{P} \times \mathcal{N}} \Phi_T\left(N_T, M_T, P_T\right) \delta\left(\frac{N_T P_T L_0}{R_b(M_T)} = B_T - B'_T\right) \\ & \times \sum_{N_J \in \mathcal{S}} \Phi_J\left(N_J\right) \delta\left(\frac{N_J P_J L_0 F(M_T, \theta)}{R_b(M_T)} = B_J - B'_J\right), \end{aligned} \tag{4.46}$$

where $\delta(\cdot)$ is a function equal to 1 when the statement is true and 0 otherwise, and where $F(M_T, \theta)$ is always 1 for the blind jamming case. Fig. 4.2 shows the state transition graph for the multistage game $\mathbb{G}$. As for the previous scenarios, transitions are allowed from bottom to top and from right to left, as a consequence of nodes T or J consuming energy to send packets or jam slots, respectively.

By substituting (4.46) into (4.45), we have a full recursive formulation for the expected long-term payoff $\mathbb{E}[U_i^{(m)}(\Gamma)]$ for any strategy pair. Once the payoff bimatrix is thus constructed, the Lemke-Howson algorithm can be used to find the mixed NE [194].

4.4.3 Simulation Setup

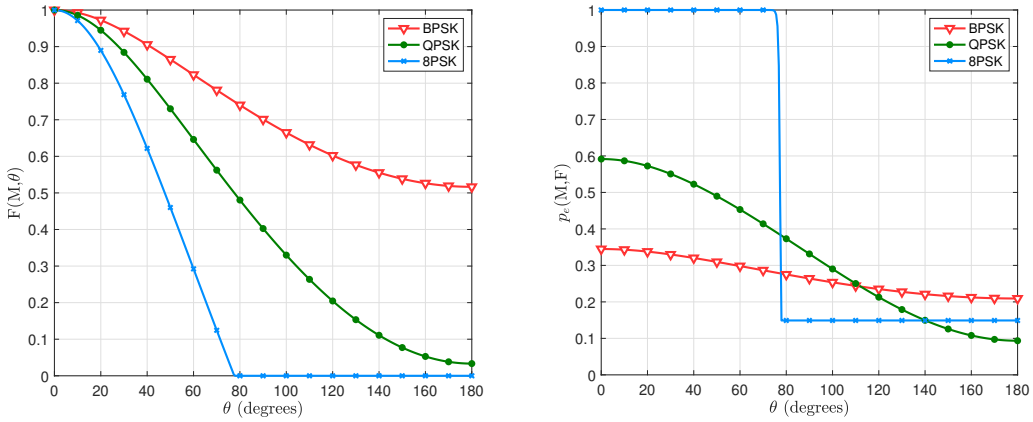
We analyze the performance of the transmitter and jammer using all the three possible modulations available in the LACE17 dataset: BPSK, QPSK, Eight Phase-Shift Keying (8PSK). Each modulation M_T corresponds to a different bitrate $R_b(M_T)$ and,

Table 4.4: Modulation and coding schemes used in the LACE17 dataset [2].

Modulation	Bitrate	Code type	Code rate
BPSK	116 bit/s	Convolutional	1/2
QPSK	232 bit/s	Trellis Coded Modulation (TCM)	1/2
8PSK	464 bit/s	TCM	2/3

consequently, to a different packet duration time $T(M_T)$. The length L_0 of the packet, expressed in bits, is constant for all the modulations. The modulations use channel coding to protect their content, with different rates: if we define the bitrate for a BPSK modulation as $R_b(BPSK) = R_0$, we have $R_b(QPSK) = 2R_0$ and $R_b(8PSK) = 4R_0$ (uncoded 8PSK would only have a bitrate 3 times larger than BPSK, but the MCS also uses a code with less redundancy). Consequently, the packet duration time is $T(BPSK) = L_0/R_0 = T_0$, $T(QPSK) = L_0/R_b(QPSK) = T_0/2$, $T(8PSK) = L_0/R_b(8PSK) = T_0/4$. In the next section we will consider whether it is better for the transmitter to use a high bitrate which reduces the packet transmission time (making the system less prone to reactive jamming) and also the energy consumption at the price of a higher BER, or to use a more robust modulation which increases the packet duration, and the energy consumption, and makes the system more prone to reactive jamming.

We consider a scenario with a packet size $L_0 = 192$ bit and with a bitrate $R_0 = 116$ bit/s for a BPSK modulation, as in the LACE17 dataset. The modulation and coding schemes are summarized in Table 4.4. The distances of T and J from R in the scenario are equal to $d_{TR} = 1200$ m and $d_{JR} = 600$ m, respectively, while d_{TJ} depends on the angle θ and can be computed according to Equation (4.38). Using these values, we are able to simulate a scenario where 8PSK packets cannot be reactively jammed for high values of θ, the QPSK packets can always be reactively jammed (although only for a small portion of the packet for θ around 180 degrees), and the BPSK packets can always be reactively jammed with an overlap higher than 50%. This choice of the parameters enables us to analyze the case in which the packet transmission duration is of the same order of magnitude as the propagation delay, which is unique to the underwater scenario and has a non-trivial strategy that depends on the geometry of the problem and on the communication parameters. In addition, we consider $\tau_h = 0$, which is the setting that is most advantageous to a reactive jammer: furthermore, in most practical scenarios, in which the duration of a packet and the propagation delay will be measured in seconds, τ_h has a negligible effect. Indeed, the two extreme scenarios in which the packet transmission time is far longer (or shorter) than the propagation delay has been already well-studied. In the former, reactive jamming is always the best choice, as the jammer can avoid wasting energy on jamming an empty channel and still be sure to block any packet transmission. In the latter, reactive jamming is impossible, as the jammer will only sense the packet when it is too late to jam any significant portion of it, unless it is directly between the transmitter and the receiver. The purely blind jammer case corresponds to the one we analyzed in Section 4.2, while the purely



(a) Overlap $F(M, \theta)$ between the signals.

(b) Jammed packet error rate.

Figure 4.20: Transmission parameters as a function of θ, for the three considered modulations.

reactive jammer is well analyzed in the relevant literature [185, 206, 207].

Figure 4.20a shows the overlap between the jamming signal and the packet transmitted by T for the three considered modulations, while varying the value of θ. Figure 4.20b displays instead the resulting jammed packet error rate.

Although our framework can support a wide set of strategies, i.e., of parameters that each player can choose, the evaluation was performed considering the information available in the LACE17 dataset. For this reason, we could not consider power control, i.e., the transmitter can use only one possible power level P and the same power level is also used by J. We consider each update to be composed of $K = 4$ information packets, and the maximum number of packets that can be transmitted at each subgame to be equal to $N_{\max} = 2K$, considering the additional redundant packets generated by the packet-level code. In our simulation, the action space for the transmitter is then $\mathcal{M} \times \mathcal{P} \times \mathcal{N}$, where $\mathcal{M} = \{\text{BPSK}, \text{QPSK}, \text{8PSK}\}$, $\mathcal{P} = \{P\}$, and $\mathcal{N} = \{K, K+1, \ldots, 2K\}$. This corresponds to a scenario in which the transmitter can use MCS control and packet-level coding as defense mechanisms, but not power control. The transmitter can also choose the slots in which to transmit over the $N_{\max}$ available slots, using a random strategy to maximize its chances of avoiding a blind jammer. Naturally, evasive defense is not effective against a reactive jammer.

We assume the same initial battery level for both T and J, i.e., $B_T^{(0)} = B_J^{(0)}$. The battery levels are quantized according to an energy quantum E_q such that each packet can be transmitted using an integer number of energy quanta $E_{\text{tx}}(M_T) = Q(M_T)E_q$. $Q(M_T)$ depends on the employed modulation and is equal to $Q(M_T) = 2, 4, 8$ for 8PSK, QPSK, BPSK, respectively. Consequently, the transmission of a packet with BPSK takes 8 energy quanta, while the transmission with 8PSK only takes 2. We consider $B_T^{(0)} = B_J^{(0)} = 400E_q$. This choice allows us to study a sufficiently long game

where both nodes can play any strategy for most of the game, while maintaining a low computational complexity. This poses a limit to the maximum number of packets that can be transmitted in the whole game, and therefore on the maximum number of subgames that can be played. Considering the 8PSK modulation, and the lowest possible number of packets that has to be transmitted in each subgame, i.e., $N_T = K = 4$ packets (i.e., no packet-level coding), the maximum number of subgames is limited to 50. For this reason, we set the time horizon $\Gamma = 50$ to let T and J play with full foresight of the rest of the game.

4.4.4 Results

In this section, we compare blind and reactive jammer performance while changing the geometry of the scenario, i.e., varying the angle θ from 0 to π. The goal is to understand for which scenario reactive jamming is more effective than blind jamming.

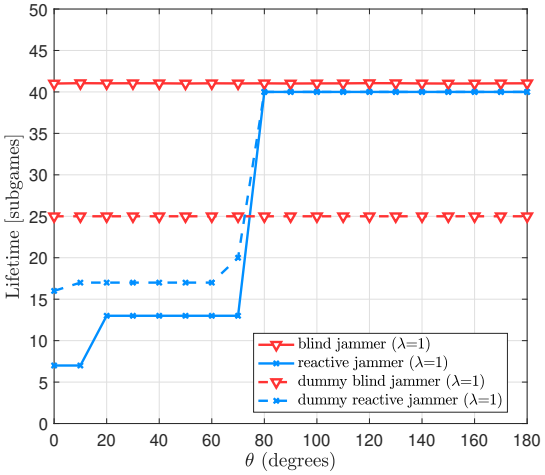
The framework presented in Section 4.4.1 allows us to compare the two jammer types for different distances and angle θ obtaining the strategies for each player and then analyzing the results through Monte Carlo simulation. First, we analyze in detail the performance and the strategies for both jammer types considering the distance $d_{JR} = d_{TJ}/2 = 600$ m as described in Section 4.4.3. Clearly, the obtained trade-offs are specific for the considered scenario, but similar analysis and considerations also apply for a different choice of the distances between the nodes. Then, to make our study more general, we analyze how the trade-off changes for different distances between jammer and receiver showing the threshold on the angle θ that makes one jammer type more effective than the other. Finally, we perform a sensitivity analysis considering imperfect information on the jammer position.

4.4.4.1 Performance Analysis

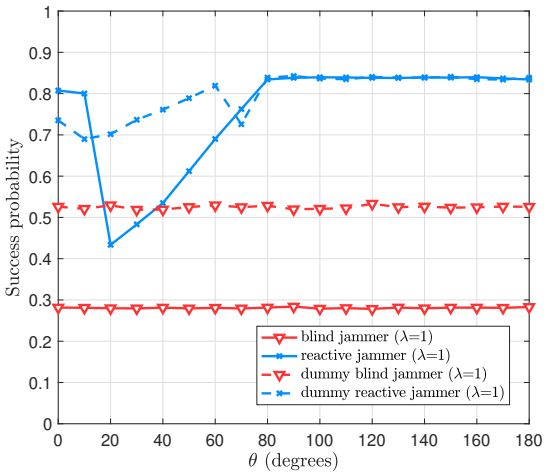
In the blind solution, the geometry is not affecting the performance of the jammer, while the reactive scenario depends on the geometry due to the different portion of the packet that the jammer can damage. In this section, we present the lifetime, i.e., the number of played subgames, as a function of θ (Figure 4.21a and Figure 4.22a), the corresponding subgame success probability (Figure 4.21b and Figure 4.22b), i.e., the number of subgames won with respect to the number of subgames played, and finally the overall number of subgames won by the transmitter (Figure 4.21c and Figure 4.22c).

Figure 4.21a and Figure 4.22a shows the lifetime computed as the number of subgames played before the transmitter runs out of battery. Specifically, Figure 4.21a represents the results without discount factor, i.e., with $\lambda = 1$, for the blind and reactive jammer, while Figure 4.22a shows the result with a discount factor $\lambda = 0.95$. As mentioned in Section 4.4.1, the discount factor makes the player more short-sighted, by letting future rewards for successive subgames count less than the present one.

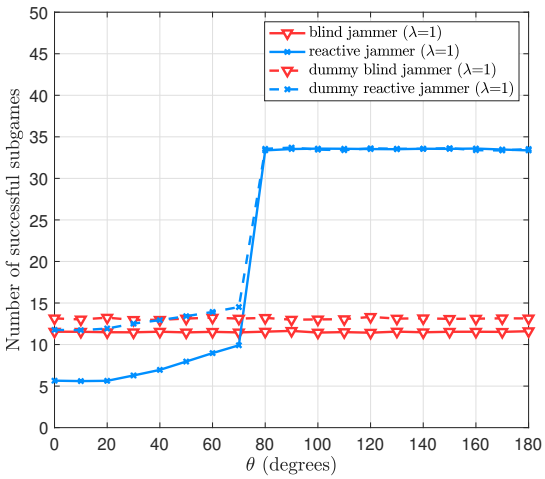
The consequence of using a discount factor less than 1 is to let T play more aggressively, sending more packets and with a more robust modulation to protect itself



(a) Lifetime of the transmitter.



(b) Transmission success probability.



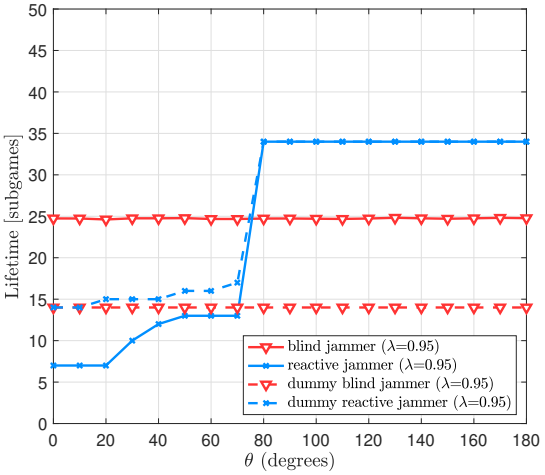
(c) Overall number of successful subgames.

Figure 4.21: Performance against a reactive and blind jammer, considering optimal and dummy strategies with λ=1, varying the geometry and considering optimal and dummy strategies.

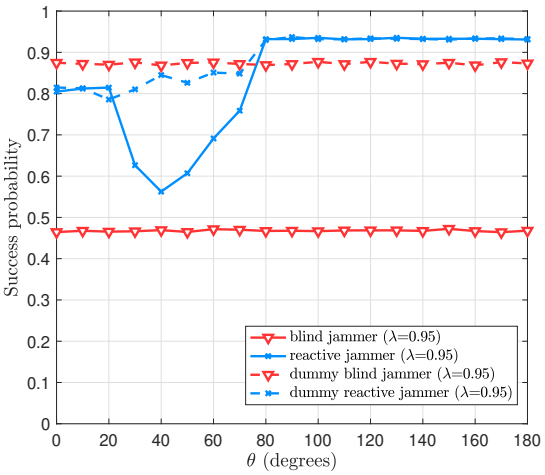
from J, with the results of reducing T's lifetime while increasing its subgame success probability (Figure 4.21b).

As mentioned above, the blind jammer scenario does not depend on the angle θ, while the reactive jammer's performance changes as θ changes.

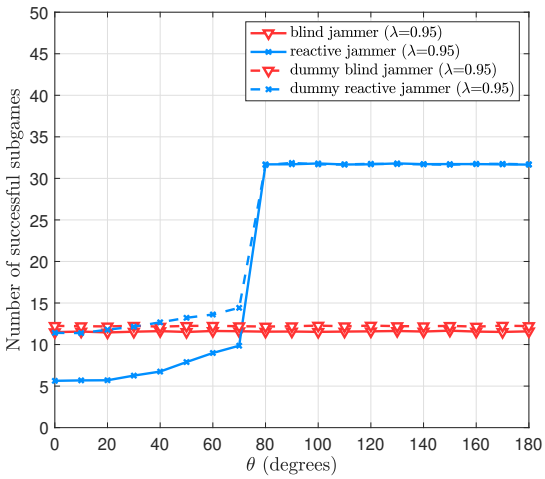
In the considered scenario with $d_{JR} = d_{JT}/2 = 600$ m, Figure 4.21c shows that when $\theta < 80$ degrees, a reactive jammer is more effective than a blind one. In particular, when $\theta < 20$ the reactive jammer can immediately counteract each transmission by almost completely overlapping each packet. In this case, the presence of a reactive jammer forces T to transmit with a robust modulation, such as BPSK, for which



(a) Lifetime of the transmitter.



(b) Transmission success probability.



(c) Overall number of successful subgames.

Figure 4.22: Performance against a reactive and blind jammer, considering optimal and dummy strategies with $\lambda = 0.95$, varying the geometry and considering optimal and dummy strategies.

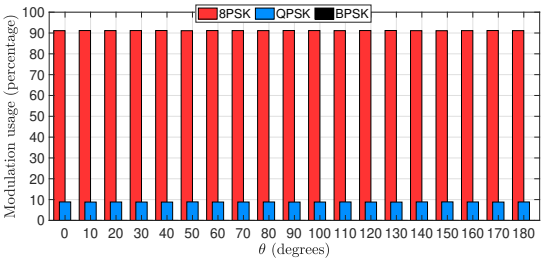
the jamming effectiveness is much lower than for the other modulations, adding also redundancy to protect the data, at the price of a fast battery depletion, thus obtaining a lifetime lower than 10 subgames. This gives T a probability to win the subgames higher than 0.8, but, due to the small number of subgames played, the overall number of T's updates successfully received by R is smaller than 10. Increasing the angle θ, the QPSK modulation also becomes a valid option for the transmitter, since the portion of the packet that J is able to jam decreases, thereby decreasing the reactive jammer's effectiveness. Indeed, Figure 4.21b shows a drop in the subgame success probability caused by a change in the modulation strategies from a more robust modulation (BPSK)

to a less robust one (QPSK). However, using QPSK results in spending less energy for each transmission, and thus gives T the possibility to increase the lifetime and the overall number of subgames won in the whole game. As soon as the angle increases, the shorter transmission time and shorter collision window for the reactive jammer enable T to achieve a higher success probability.

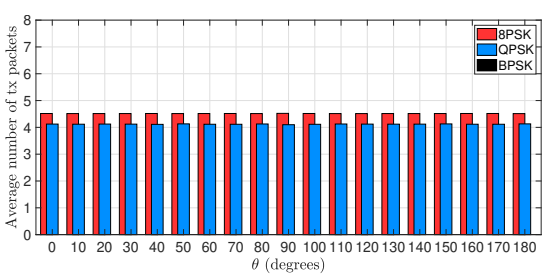
On the other hand, when $\theta \geq 80$ degrees, it is more convenient for the jammer to play blind, since the number of successful subgames for the transmitter increases up to 34 subgames against a reactive jammer (vs 15 against a blind one). In the reactive case, with $\theta \geq 80$ degrees, the jammer is no longer able to jam the 8PSK packets, as shown in Figure 4.20a. Therefore, the transmitter always chooses a strategy involving 8PSK modulation and only fights against the channel. For example, in the scenario with $\lambda = 1$, T always protects the update with one additional redundant packet, i.e., transmitting $N = 5$ overall packets with 8PSK modulation. This is confirmed by Figure 4.21a where the lifetime for the reactive jammer scenario with $\theta \geq 80$ is equal to 40 subgames. Considering a discount factor $\lambda = 0.95$, shown in Figure 4.22, the transmitter has a lower foresight and becomes more aggressive in terms of energy spent in each subgame, reducing its lifetime while increasing the subgame success probability, with the overall effect of decreasing the total number of successful subgames.

We also compare the results with two dummy policies, in which the blind jammer always tries to jam K packets for each update, while the reactive jammer jams enough packets to let the transmitter send only $K-1$ in a clear channel, regardless of the MCS used. These strategies are often used in practice and can be effective, as the performance for the blind jammer shows, but they are always suboptimal, as the jammer cannot react to the strategy of the transmitter. This is visible in Figure 4.21c, as the number of successful subgames is the metric that the players are optimizing for. Interestingly, the relatively small difference between the NE of the blind jammer and of the dummy is not due to a similarity in the strategies: the dummy jammer spends much more energy and makes the transmitter protect its transmissions in a far more expensive way, reducing the lifetime significantly, but it lets through about 85% of packets, while the game played by the NE nodes is much slower, with lower success probabilities and less energy expended per round, but a far longer lifetime for both nodes.

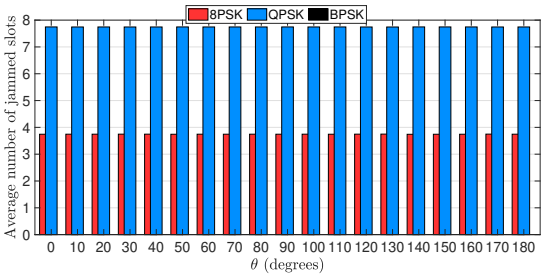
This result clearly depends on the specific distances and MCSs used in this scenario, and choosing a different geometry or different settings for the transmission might change the value of θ at which it is convenient for the jammer to switch to blind jamming, but a general rule holds: the lower the angle θ, the better reactive jamming works, while blind jamming is unaffected by θ as long as the slots can be synchronized (i.e., if the jammer knows the position of all nodes). Reactive jamming is more energy-efficient, as the jammer never wastes energy, transmitting the jamming signal only when it senses a legitimate transmission. Naturally, this means that it needs to deal with the delay, and that it can become ineffective, particularly over long distances.



(a) Usage, in percentage, of the three modulation types.

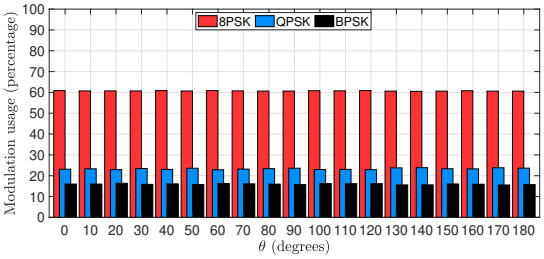


(b) Average number of packets sent by T.

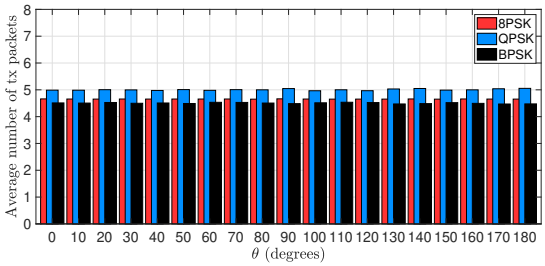


(c) Average number of slots jammed by J.

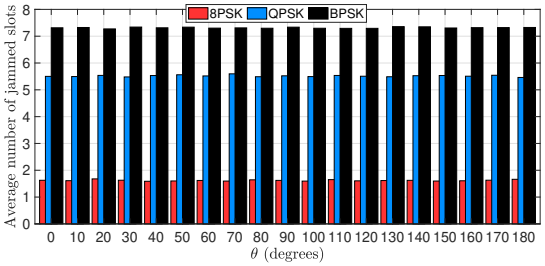
Figure 4.23: Strategies with a blind jammer with $\lambda = 1$.



(a) Usage, in percentage, of the three modulation types.



(b) Average number of packets sent by T.



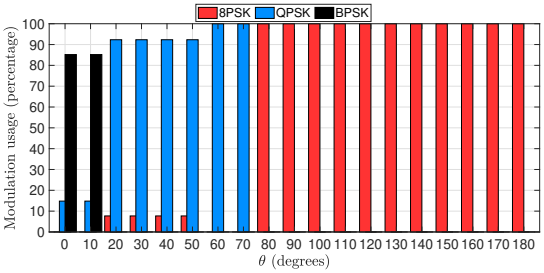
(c) Average number of slots jammed by J.

Figure 4.24: Strategies with a blind jammer with $\lambda = 0.95$.

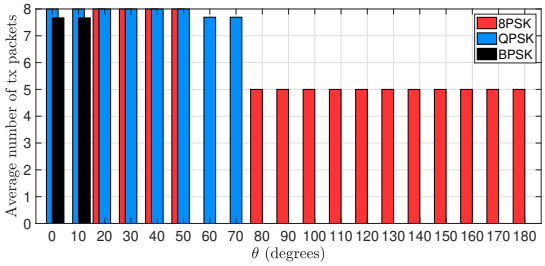
4.4.4.2 Strategies

In this Section we analyzed in more detail the strategies employed in the scenario with $d_{JR} = d_{TJ}/2 = 600$ m. Figures 4.23 and 4.24 delve deeper into the choices that the two agents make when the jammer is blind, with $\lambda = 1$ and $\lambda = 0.95$, respectively. As expected, the strategies in this case are not affected by the angle θ, as the jammer *proactively* jams part of the slots instead of reacting to the transmitter: by compensating for the different propagation delays of the legitimate and jammed signal, the jammer effectively synchronizes them, but has to give up any knowledge of whether there is a transmission in a given slot or it is just jamming an empty channel. We remind the reader that there are $N_{\max} = 2K$ slots in which T can transmit, and that the blind jammer needs to decide how many to jam. The choice of the slots is random for both nodes, as this is the optimal strategy in an anti-coordination game. Fig. 4.23 shows that the transmitter uses 8PSK most of the time, with some redundancy to protect itself from the jammer. About 10% of the time, the transmitter uses QPSK with no redundancy. When using QPSK, the jammer is more aggressive, as seen in Fig. 4.23c: if the transmitter uses QPSK, it is almost always active, i.e., it jams almost all the $N_{\max}$ slots employed for the update, while it is only active approximately half the time when the transmitter uses 8PSK, as being able to jam only few packets is enough to cause the loss of the update. We want to remind the reader that a blind jammer is able to jam the whole packet since J infers the modulation employed from the control messages sent by T before the beginning of a subgame. If we set $\lambda = 0.95$, the transmitter considers the current packet more than future ones, using more energy: as Fig. 4.24 shows, this causes it to use 8PSK less often, using QPSK 25% of the time and BPSK 15% of the time. For all the three employed modulations, a low level of redundant packets (or zero redundant packets) is used. Correspondingly, the jammer is more active: when the transmitter uses BPSK, it is almost always actively jamming all the slots, while it jams approximately 60% of the available slots in a subgame when the transmitter uses QPSK. The jammer is correspondingly less aggressive against the lightly protected 8PSK transmissions, which happens mostly as the transmitter's battery gets low, and it has to reduce its energy consumption to avoid depleting its battery prematurely. In all cases, the position of the jammer does not matter: as the blind jammer can synchronize with the transmission opportunities, the only parameter related to the geometry of the scenario that affects its performance is its distance from the receiver, while the angle θ does not have any effect.

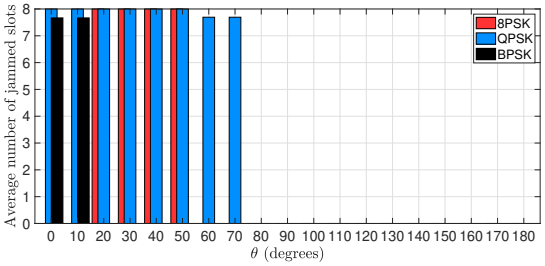
This is not true if the jammer is reactive: in this case, as Fig. 4.25 shows, the effectiveness of the jammer is strictly dependent on how quickly it can sense packets, i.e., on the value of the angle θ. If the angle between the jammer and the transmitter is small, the transmitter is forced to spend more energy to defend itself, using BPSK and sending 3 or 4 redundant packets for protection. On the other hand, the jammer tries to overcome the defenses by jamming all the available slots, increasing the chances of packet loss. As the angle grows, the jammer becomes less effective: from 30 to 70 degrees, the transmitter can use QPSK most of the time, still adding several redundant packets. Once $\theta \geq 80$, the jammer is completely harmless, as the jamming signal



(a) Usage, in percentage, of the three modulation types.

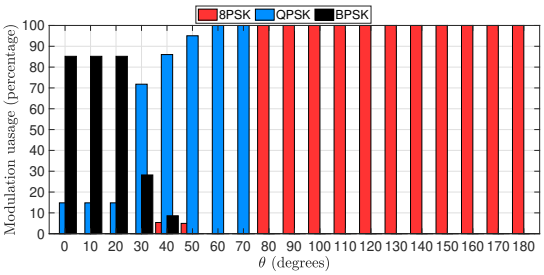


(b) Average number of packets sent by T.

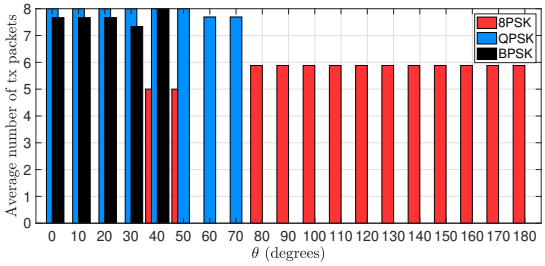


(c) Average number of packets jammed by J.

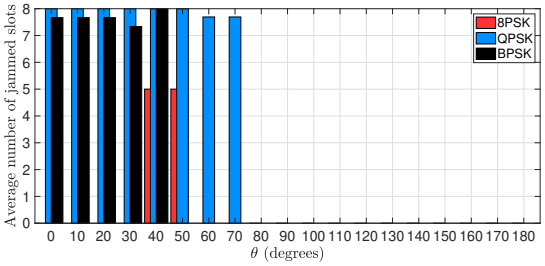
Figure 4.25: Strategies with a reactive jammer with $\lambda = 1$.



(a) Usage, in percentage, of the three modulation types.



(b) Average number of packets sent by T.



(c) Average number of packets jammed by J.

Figure 4.26: Strategies with a reactive jammer with $\lambda = 0.95$.

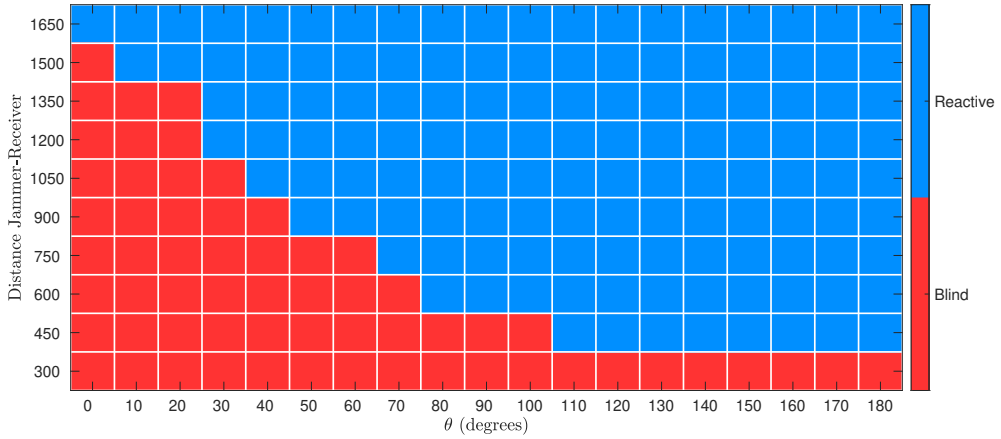


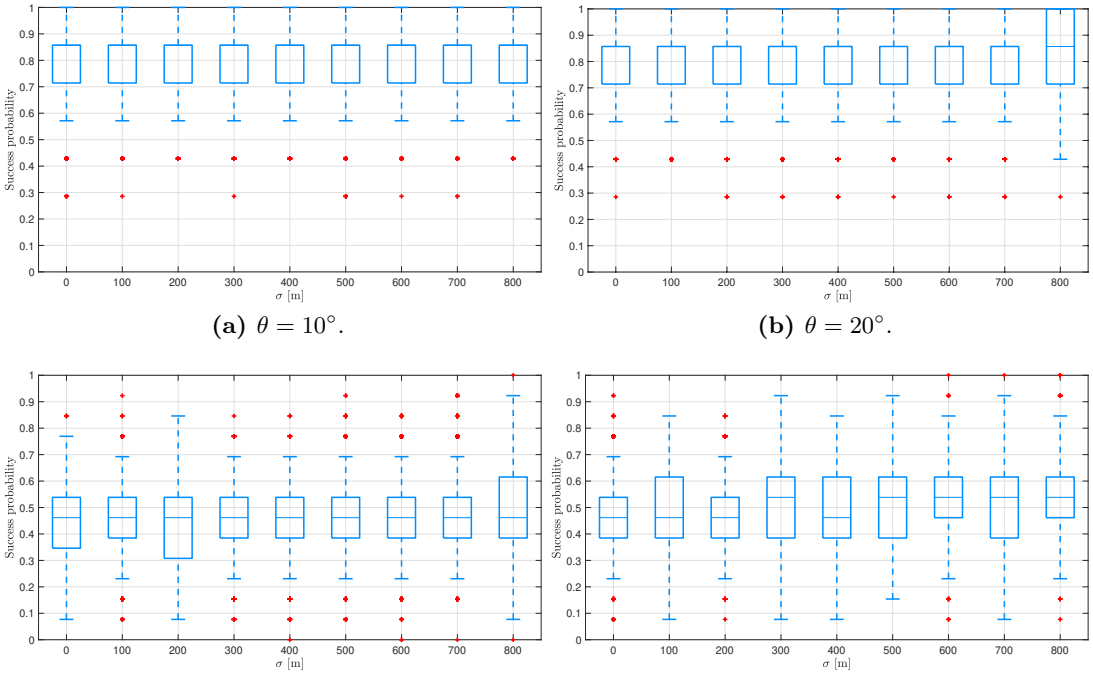
Figure 4.27: Analysis as a function of the angle θ and of the distance between jammer and receiver, while keeping constant $d_{TR} = 1200$ m. Blue squares mean that it is more convenient, in terms of overall number of subgames won, for T to play against a reactive jammer. Conversely, red squares mean that it is more convenient for T to play against a blind jammer.

reaches the receiver only when the packet transmission is already complete. In this case, the transmitter is free to act as if the jammer were not there, using the efficient 8PSK with only one extra packet to protect the transmission from channel errors.

Fig. 4.26 depicts the strategies for the reactive scenario with $\lambda = 0.95$. As for the blind jammer case, short-sighted players tend to concentrate on the current transmission, so the transmitter adds more redundancy and uses more conservative modulations more often. This is not true if the angle is 20 degrees or lower, as in that case the transmitter with $\lambda = 1$ already used the most conservative settings. This is also true for $\theta \geq 80$, as the transmitter adds more redundancy with packet-level coding: even if it just has to contend with the environment noise, it still privileges short-term success over a longer battery lifetime.

4.4.4.3 Analysis Varying the Jammer Distance

As stated above, all the results presented before depend on the considered distances between the nodes. However, the same analysis can be repeated changing the distances and obtaining similar trade-offs between reactive and blind jammer. As last step, we provide a general analysis of the trade-off between reactive and blind jammer, showing how the threshold on the angle θ depends on the jammer's distance. To this purpose, Figure 4.27 shows when it is more convenient for the transmitter to play against a reactive (blue square) or a blind (red square) jammer as a function of the distance between the jammer and the receiver. Specifically, we analyzed whether the overall number of subgames won by T is larger against a reactive or a blind jammer, for different angles θ and distances between jammer and receiver d_{JR}. When $d_{JR} = 300$ m, it is always better for the transmitter to play against a blind jammer, even for higher



(a) $\theta = 10°$. **(b)** $\theta = 20°$.

(c) $\theta = 40°$. **(d)** $\theta = 50°$.

Figure 4.28: Boxplot of the success probabilities in the reactive jammer scenario for different σ and at different angles θ.

values of θ. Indeed, in this case the reactive jammer is always able to reactively jam the packet with all the considered modulations, even if partially. When $d_{JR} \geq 1650$ m, the reactive jammer is not able to jam any of the employed modulations at any angle. Therefore, T plays against an empty channel most of the time, even with $\theta = 0$: in this case, with $d_{JR} \geq 1650$ m, T is placed between R and J, and therefore the jamming signal needs to travel a longer distance than the legitimate packet, becoming unable to jam it.

4.4.4.4 Imperfect Position Information: Sensitivity Analysis

The assumption of perfect information can be unrealistic in UANs, as underwater localization is often less than perfect, particularly if the jammer is trying to remain unobserved. In the following, we then perform a short sensitivity analysis for the examined strategies, in which we put the jammer at a disadvantage by reducing the precision of its localization. On the other hand, the transmitter has perfect information about the jammer, and the jammer knows the exact location of the legitimate nodes.

Naturally, this scenario is also not fully realistic, as the transmitter will likely have some uncertainty over the position of its adversary, but we consider this extreme case as the most advantageous for the transmitter. We then include a bivariate Gaussian noise $w \sim \mathcal{N}(0, I\sigma^2)$ on the jammer's estimate of its own position, where σ is the

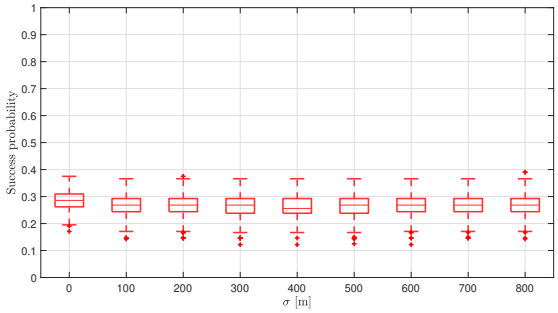


Figure 4.29: Boxplot of the success probabilities in the blind jammer scenario for different σ. The blind strategy does not depends on θ.

standard deviation and I is the identity matrix. This error affects both the estimate of the distance between the jammer and the receiver and the estimate of the delay between the legitimate and jamming signal, affecting the synchronization of the signals. The jammer signal will then be poorly synchronized with the legitimate packets, often overlapping with the previous or next slot. We consider this effect in the Monte Carlo simulations, whose results for a reactive jammer are shown in Fig. 4.28.

The reactive jammer is only slightly affected by the positioning error, particularly at low angles: if the two nodes are aligned, the precise distance matters less than getting the correct strategy, and the jamming is still effective even with some imprecision in the slot synchronization. On the other hand, Fig. 4.28d shows that, as θ gets closer to the cutoff value, the fraction of the packet that is jammed is increasingly small, and making intelligent decisions based on correct information becomes extremely important. In this case, the positioning error can affect the strategy, as even relatively small errors can significantly increase the success probability for the transmitter, and consequently reduce the jammer's payoff.

On the other hand, the blind jammer is almost unaffected by positioning errors, as its strategy is always the same at any angle, as Fig. 4.29 shows. In general, our results should hold if the information available to the nodes is imperfect, although we leave a more extensive analysis for future work.

4.5 Conclusions

In this chapter we studied underwater jamming attacks using game theory, exploring the effectiveness of using blind and reactive jamming that targets both the disruption of the victim's communication and the depletion of its battery. First, we studied blind jamming as a function of the distance between the jammer and the receiver. We compared the complete information scenario, where the positions of each node is known to all the others, and the incomplete information scenario, where the transmitter does not know the position of the jammer. The results show that the distance of the jammer from the receiver is the main variable affecting the performance of the transmitter, but knowing it in advance is not necessary, as the jammer's actions will rapidly unmask

it. Then we compared blind and reactive jamming as a function of the geometry of the network deployment, to understand where a blind jammer is more effective than a reactive one. Indeed, reactive jamming may not always be feasible due to the high latency caused by the low propagation speed of sound. We thoroughly investigated the strategies selected by the transmitter and the jammer, to understand the trade-off between using more robust modulation but more prone to reactive jamming, or using less robust modulation but with a shorter packet duration, therefore more difficult to be reactively jammed. In addition, we observed that there is a threshold on the angle θ after which it is more convenient for the jammer to play blind, choosing randomly which slots to jam.

Chapter 5

Replay Attack and Countermeasures in Underwater Acoustic Networks

5.1 Introduction

Although the wireless nature of the acoustic medium makes UANs vulnerable to various malicious attacks, limited consideration has been given to security challenges in this environment so far [23, 24]. Various types of DoS attacks can be conducted in UANs. As discussed in Section 1.2.1, some of these attacks assume the ability of the malicious node to produce or manipulate legitimate messages, e.g., *sinkhole* and *wormhole* attacks. For other attacks, instead, the malicious node does not need to be able to generate any legitimate message to disrupt the network operations, e.g., jamming and basic replay attacks[1]. Additionally, no sophisticated hardware or processing capability is required.

Although countermeasures to DoS attacks have been widely studied in the radio frequency domain [112, 128], only few solutions have been proposed for UANs, mainly focusing on jamming attacks [132, 208], wormhole attacks [209, 210] and the usage of security tools to protect the integrity and confidentiality of the received messages [211, 212, 213, 214]. In this chapter, we investigate countermeasures against a replay attack (Figure 5.1). In this type of attack, the malicious node records messages transmitted by legitimate nodes in the network and replays these messages. In this chapter, we assume the attacker knows the waveform used for the transmission, i.e., it is able to decode the packet, but without knowing the protocol stack used in the network, therefore with no capabilities to understand the content of the packets. The objective of the attacker is to waste the scarce network resources.

Different packet replay strategies can be used by the attacker, depending on its capabilities and on the selection of the packet(s) that will be replayed.

In this chapter, we analyze the effect of different replay attack strategies in a multi-

[1] In the replay attack considered in this chapter a node can only receive and retransmit a packet as it is, without the ability to modify its content.

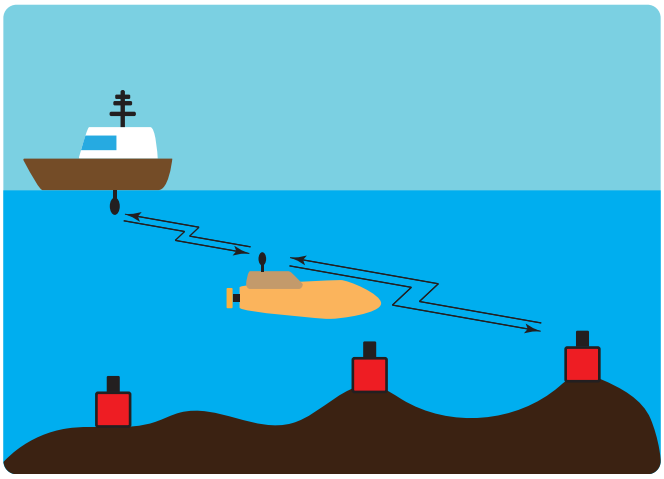


Figure 5.1: Replay attack: an AUV, acting as a malicious node, records packets transmitted by the surrounding nodes and re-injects them into the network.

hop UAN. Additionally, various countermeasures are proposed through the development of a network security layer with cross-layer capabilities, thus minimizing the overhead required for a secure communication. To validate and evaluate the proposed solution, a simulation study is conducted using the DESERT Underwater Framework [149] where all the various attacks and possible countermeasures are implemented.

In Section 5.2 we present all the configurations of the replay attack investigated in this chapter, as well as the design of the security layer used as countermeasure for this type of attack. In Section 5.3 we describe the scenario and the network topologies used to test the attacks and the countermeasures, while in Section 5.4 we present the simulation results and therefore the evaluation of both the attacks and the security system. Finally, in Section 5.5 we draw our concluding remarks.

5.2 Replay Attacks and Countermeasures

As stated in Section 1.2.1.2 some replay attack countermeasures for terrestrial networks consider the use of a timestamp [137] or a HASH index [140]. The same countermeasures cannot be directly applied to UANs because of the large packet delivery delay in acoustic networks and the high HASH collision probability that would characterize the short underwater packets. However, these countermeasures can be adapted to the peculiarities of underwater networks. Indeed, in this chapter we propose a timestamp-based countermeasure by considering a large validity period and a reduced complexity. Furthermore, we investigate a HASH index mechanism reducing the HASH collision probability by using a timestamp as input to the HASH function.

Despite the fact that we are not considering any packet encryption, in this chapter we assume that the malicious node only knows the waveform used for the transmission and is not aware of the protocols employed in the network, therefore it cannot modify

a recorded message, but can only send it later in time. In this chapter, four types of replay attack strategies are analyzed, in order to inspect which defense mechanism best reacts against different attacks.

1. FIRST-PACKET: only the first packet detected by the malicious node is replayed with a given repetition time during the entire simulation. This is the simplest attack, as the attacker needs to record only one packet and then retransmit it repeatedly.

2. LAST-PACKET: only the last packet sensed by the malicious node is replayed with a given repetition time during the entire simulation. The identification of the presence of this attack is quite hard, as the malicious node always transmits fresh data.

3. MULTI-PACKET: all packets sensed by the malicious node are recorded and replayed only once. Based on the considered strategies, the amount of traffic injected by the MULTI-PACKET approach depends on the number of messages transmitted by the legitimate nodes, while the other attacks inject a fixed number of packets, independent of the network traffic.

4. HOLD-PACKET: all packets sensed by the malicious node are recorded. After a certain amount of time (e.g., one hour of recording), the attacker chooses at random and replays one of the recorded packets at a time [134].

All attacks aim to inject packets into the network, with the goal to fill the MAC queues of the nodes, and, therefore, saturate the network.

In this chapter, we propose a security layer placed between the routing and MAC layers, able to verify the freshness of a packet with different approaches, either based on time or on a unique packet identifier computed by combining time and the generating node address information with the HASH function (more details are presented later in this section). Regardless of the freshness mechanism employed, the security layer performs the two operations listed in the following and summarized in Figure 5.2.

- When generating a new packet, a 4 Bytes freshness index information is added to the packet (Figure 5.2(a)).

- When receiving a packet, the freshness of the received message is evaluated at the lower layer. If the packet passes the security check, it is forwarded to the upper layer, otherwise it is dropped (Figure 5.2(b)).

Indeed, with the network protocols considered in this chapter (flooding and static routing), when no security mechanisms are applied, whenever a node receives a packet from a neighbor, it forwards it to the next hop, without checking the content of the message. For this reason a security layer is required to check the freshness of a packet. The two replay attack countermeasures analyzed in this chapter are named TIME and HASH, respectively.

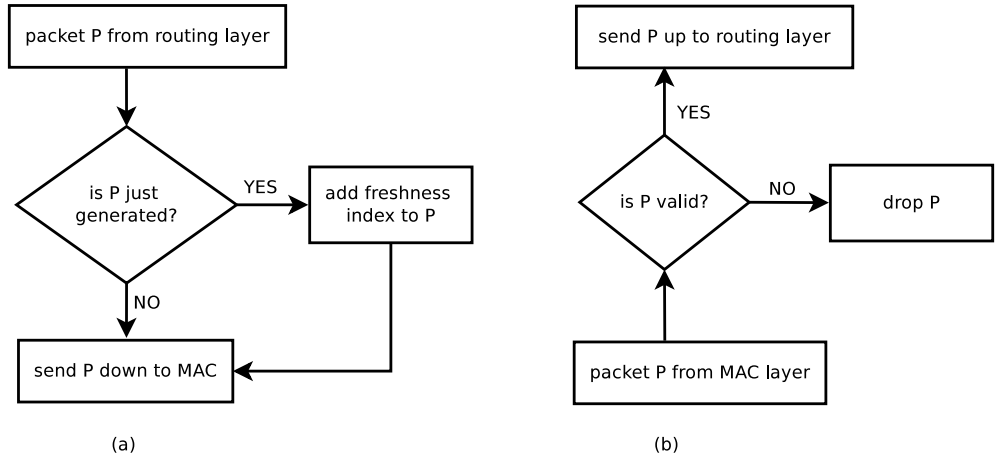


Figure 5.2: Diagram describing the operations performed by the security layer: (a) packets arrived from the routing layer; (b) packet received from the MAC layer.

1. TIME uses the packet generation time to verify the freshness of a packet: if the difference between the current time and the generation time of a received packet is above a pre-defined time threshold, the packet is discarded. This method requires the transmitter to store the packet generation time in an additional header with size 4 Bytes.

2. HASH computes the XOR operation between the HASH of the packet generation time and the HASH of the node address. This method requires the transmitter to store the HASH value in an additional header with size 4 Bytes: each time a packet is received by a node, the node checks in a HASH list whether a packet with the same HASH has already been received or not. If so, it discards the packet, otherwise the packet is forwarded to the upper layer and the value of the HASH is stored in the HASH list.

The HASH list has fixed size: once the list is full the HASH corresponding to the oldest packet is discarded: during the protocol evaluation the HASH list size required to ensure security to all replay attacks will be analyzed.

For the freshness information size, if we assume a maximum deployment time of one year, keeping the time precision in tenths of a second, we need at least 29 bits for the time representation, hence with 4 Bytes we can ensure the attacker needs to wait 10 years before the time index overflows. We also select 4 Bytes for the HASH index, not only because most of HASH operations return a 4 Bytes number, but also because it ensures a very low collision probability. For example, if the HASH list size is 2000 packets, the probability that at least two packets have the same HASH value is less than 0.001.

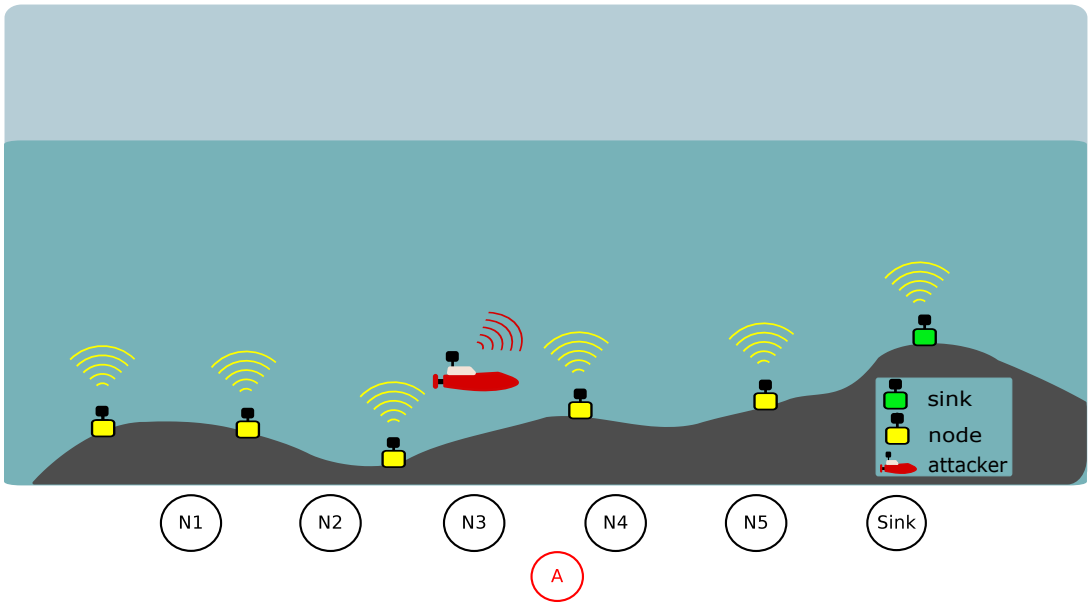


Figure 5.3: Simulation scenario and topology of `NET1`. An AUV acts as an attacker and tries to saturate the underwater network in order to reduce the packet delivery ratio at the sink (green node).

5.3 Simulation Scenarios and System Settings

In this chapter we consider two different scenarios, `NET1` and `NET2`. `NET1`, illustrated in Figure 5.3, consists of a static linear network composed by 5 underwater nodes (depicted in yellow). The nodes are moored on the sea bottom with a distance between two consecutive nodes of 1 km. Each node generates and forwards packets to a common collection node (i.e., the sink, depicted in green). The AUV (depicted in red) plays the role of the malicious node, recording the packets received by the surrounding nodes and replaying them into the network. The impact of the attacker is analyzed when deployed in different positions in the network, and for both contention-free and contention-based MAC protocols, specifically TDMA (with time frame 6.5 s, equally divided between the 5 nodes) and CSMA. All the nodes in `NET1` are equipped with a medium frequency acoustic modem, with carrier frequency $f_c = 25$ kHz, bandwidth $BW = 5$ kHz, bitrate 4.8 kbit/s and a maximum range of 2.25 km. All nodes generate packets according to Poisson traffic, with packet size 125 Bytes and average inter-packet generation time 60 s. The analysis of this simple scenario is crucial to understand the impact of different versions of the replay attack in the network, as well as of the defense mechanism, varying the position of the attacker and attacking both contention-based and contention-free MAC layers. This first step will provide us with an idea on how to attack more complex networks, like the one depicted in Figure 5.4.

The second scenario (`NET2`), illustrated in Figure 5.4, consists of a hybrid network composed by 4 underwater nodes moored on the sea bottom, a ship and two AUVs

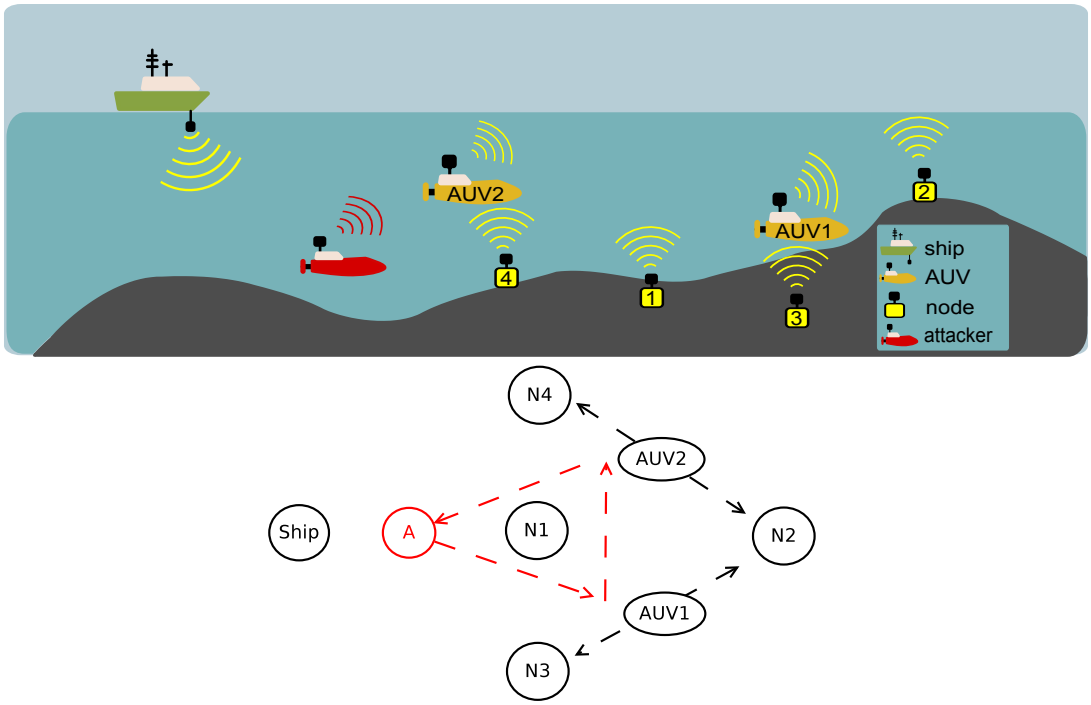


Figure 5.4: Simulation scenario and topology of `NET2`. An AUV acts as an attacker and tries to saturate the underwater network in order to reduce the packet delivery ratio of the nodes.

(depicted in yellow). The distance between two adjacent moored nodes is 3 km, while AUV1 and AUV2 move at a constant speed of 1 m/s following a linear trajectory between node 2 and node 3, and between node 2 and 4, respectively. All nodes broadcast data to all other nodes using a flooding routing protocol: in this configuration, the maximum number of hops between the static nodes is 2, while with up to 3 hops all static nodes are able to reach the AUVs. Also in this case, the AUV depicted in red plays the role of the malicious node, recording the packets received from the surrounding nodes and replaying them while moving around in the network, as shown in the topology depicted in Figure 5.4 (bottom part of the figure). All the nodes in `NET2` are equipped with a low frequency acoustic modem, with carrier frequency $f_c = 12$ kHz, bandwidth $BW = 10$ kHz, bitrate 500 bit/s and a maximum range of 4.5 km.

In `NET2`, different traffic types are generated, all according to a Poisson traffic, specifically:

- AUV sensed data: each AUV generates two packets with size 60 Bytes each, on average, every 40 s to be broadcast to all other nodes;
- AUV status for the ship: each AUV generates two packets with size 60 Bytes each, on average, every 120 s to be sent in unicast to the ship;
- node status transmission: each static node (i.e., both moored nodes and the ship)

generates a packet with size 32 Bytes, on average, every 120 s to be broadcast to all other nodes;

- ship position data: the ship generates one packet with size 60 Bytes, on average, every 90 s, to be broadcast to all other nodes;

- asynchronous messages: the ship generates one packet with size 60 Bytes, on average, every 120 s, to be sent in unicast to one of the AUVs;

- ranging information: each node generates one ranging packet with size 40 Bytes, on average, every 120 s, to be broadcast to all neighbors. This type of traffic is only transmitted to the one hop neighbors, without forwarding it to the next hops.

Table 5.1: Simulation settings

	`NET1`	`NET2`
Destination	unicast, single sink	broadcast
Nodes, hops	6, 5	7, 3
f_c, BW	25 kHz, 5 kHz	12 kHz, 10 kHz
Rate, range	4.8 kbit/s, 2.25 km	500 bit/s, 4.5 km
Traffic	each node Poisson:	7 types of traffic,
	125 Bytes, 60 s	total load: 565 bit/s
MAC	TDMA, CSMA	CSMA-ALOHA
Routing	static routing	flooding
Topology	linear, with	hybrid with AUVs
	consecutive nodes	
	1 km apart	

Both `NET1` and `NET2` have been evaluated using the DESERT Underwater Network simulator, and we used the channel model described in [5] with a spreading factor $k = 1.5$ (practical spreading), shipping activities equal to 0.5, no wind effects and a BPSK modulation. The BER is computed by applying the BER formula for the BPSK [215] to the SINR computed by DESERT, and the packet error rate assumes independent and uniform distribution of the bit errors. Although more realistic channel models can be used inside the DESERT Underwater Framework [150, 216], we decided to employ this simple model to focus our analysis more on the effects of the replay attack itself than on the performance degradation experienced on a real acoustic channel due to multipath, Doppler and time varying noise.

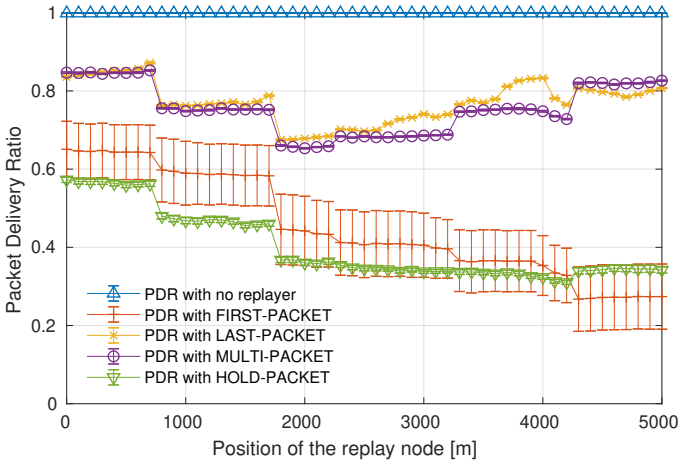


Figure 5.5: Packed delivery ratio of the network versus the replay node position in `NET1` with a TDMA MAC protocol. Position 0 m is close to the first node of the network, and position 5000 m is close to the network sink.

5.4 Results

We now analyze the impact of the replay attack and the proposed countermeasures in networks `NET1` and `NET2`.

5.4.1 Replay Attack and Countermeasures in `NET1`

For scenario `NET1`, we first analyze the impact of the replay attack when no security mechanisms are applied. Then, we evaluate the performance of the proposed countermeasures. With this scenario all nodes generate packets for the sink, with the specific intent of making the sink receiving all generated packets: for this reason we analyze the network performance in terms of PDR, i.e., the ratio between the number of packets received by the sink and the number of packets generated by all nodes of the network, without counting the duplicates.

5.4.1.1 Effect of the Replay Attack

The effect of the replay attacks in terms of PDR, when used in `NET1` with TDMA at the MAC layer, is reported in Figure 5.5: these results are presented with 95% confidence intervals. In this case the TDMA frame is 6.5 s, equally divided between the 5 nodes. The replay node does not have a MAC layer and transmits a recorded packet every 2 s. Without the attacker, almost every packet is successfully received at the destination, thanks also to the fact that a TDMA MAC is contention free, and therefore avoids any interference. The MULTI-PACKET replay attack is more destructive when the attacker is positioned in the middle of the network. This happens because the attacker simply replays only once all the recorded packets. Observing the

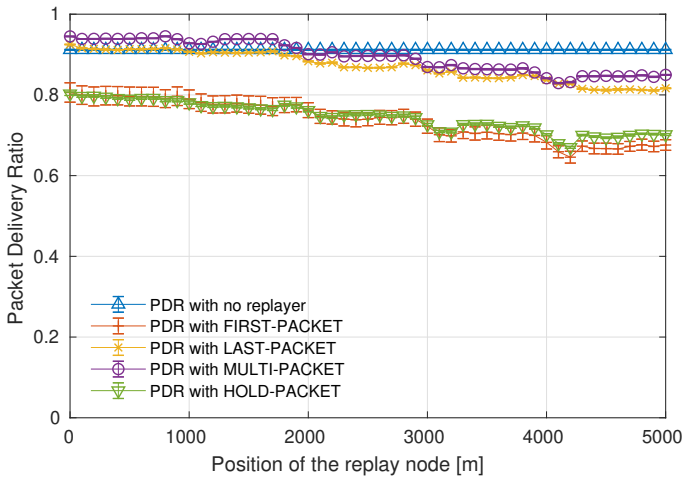


Figure 5.6: Packed delivery ratio of the network versus the replay node position in `NET1` with a CSMA MAC protocol. Position 0 m is close to the first node of the network, and position 5000 m is close to the network sink.

network from a central location allows the attacker to successfully record, and therefore transmit, more packets than when it observes the network from a peripheral location (e.g., close to the sink or to the first node of the network). In the FIRST-PACKET replay attack, the attacker always replays the first packet it recorded, by filling the MAC queue of the node that is required to forward that packet. The node forwards the attacker packet along the network, filling also the other nodes queue. This type of attack is more effective when the attacker is close to the sink, since the sink neighbors are the ones handling more packets. Therefore, the attacker can easily saturate the network exploiting this bottleneck. Depending on the node affected by the replay the effect of the attack on the network changes, therefore a different first recorded packet causes different performance. This is reflected in the the higher variability (wider confidence interval) of the FIRST-PACKET mechanism. LAST-PACKET transmits the last packet sent in the network. For this reason, when the attacker with LAST-PACKET is near the sink, it sends, most of the times, packets received from the last relay, and therefore intended for the sink, which does not need to forward any packet. If the attacker is in a more centralized location, it affects the network performance similarly to the MULTI-PACKET attack. HOLD-PACKET, instead, fills uniformly the queues of all nodes in range of the attacker, as it selects at random which packet to retransmit. This is the most effective attack if the malicious node is deployed between the first and the fifth node of the network: then FIRST-PACKET becomes the most harmful, because during its packet selection HOLD-PACKET transmits also packets for the sink, that are therefore not forwarded by any of the nodes of the network.

Differently from the previous case, in `NET1` with a CSMA protocol (Figure 5.6) 9% of the packets are lost even without the presence of the attacker, due to interference caused by the contention-based MAC protocol. Despite this disadvantage, in this network there

is a better use of the channel and, therefore, a bigger amount of traffic can be supported. The reason is due to the inefficiency introduced by the TDMA guard time, that is set equal to the propagation time experienced in the transmission between two adjacent nodes, and by the fact that in the first network a node can transmit only once within a time frame, and no parallel transmissions are scheduled, even if two nodes are separated by more than two hops. Therefore, with the same configuration analyzed before, the replay node does not overload the network as much as with the TDMA configuration. In addition, if the attacker is between the first and the third node of the network (i.e., when it is deployed between position 0 and position 2000 m) with LAST-PACKET and MULTI-PACKET it even improves the performance of the network, as it duplicates only packets that need to be transmitted for more hops, and, therefore, that have a higher probability of collision. Afterwords, it fills the packet queue of the nodes close to the sink, creating a bottleneck and thus causing a drop of 10% of the PDR. FIRST-PACKET and HOLD-PACKET, instead, always provide a drop of the PDR, even when the attacker is close to the first node, as they replicate old packets that have already been processed many hours before, therefore their retransmission does no provide any benefits to the network. Also in this case the attacker is more effective when placed close to the last relay of the network.

5.4.1.2 Replay Attack Countermeasures

In this subsection we focus on the replay attack countermeasures for `NET1` configured with a TDMA MAC layer. Figure 5.7a provides the PDR of the network under the FIRST-PACKET attack when changing the malicious node position without countermeasure (blue line), and with TIME (red line with X marker) and HASH (green line with round marker) defense mechanisms. With this attack, both countermeasures provide similar results, i.e., a 30% increase of the PDR compared to the case without defense mechanisms. A similar result is obtained for the HOLD-PACKET attack (Figure 5.7d), where the increase in PDR is 40% compared to the case without defense mechanisms. HOLD-PACKETS stores in a buffer the first 20 packets it receives, and after 3600 s it starts transmitting one of them at a time, selected at random, for the whole simulation. With this attack, the performance of TIME and HASH configured with a HASH list of 30 HASH values is very similar. In the case of a HASH list with size less than 20, instead, TIME outperforms HASH (Figure 5.8, yellow line), because if a very old packet is transmitted it will not be present in the HASH list anymore, while TIME is unaffected by this issue. On the other hand, the TIME countermeasure is substantially ineffective for LAST-PACKET and MULTI-PACKET attacks (depicted in Figure 5.7b and Figure 5.7c, respectively). Indeed, this countermeasure, with 70 s of packet validity time, is not able to drop the repetitions sent by the replay node. However setting the validity period to a lower value would lead the countermeasure to drop also legitimate packets, leading to even worse results: the value 70 s has been chosen as the minimum value of packet validity time to ensure no legitimate packets are discarded in `NET1`. HASH, instead, is immune to this phenomenon, however its PDR suddenly decreases when the distance between attacker and sink decreases, as in

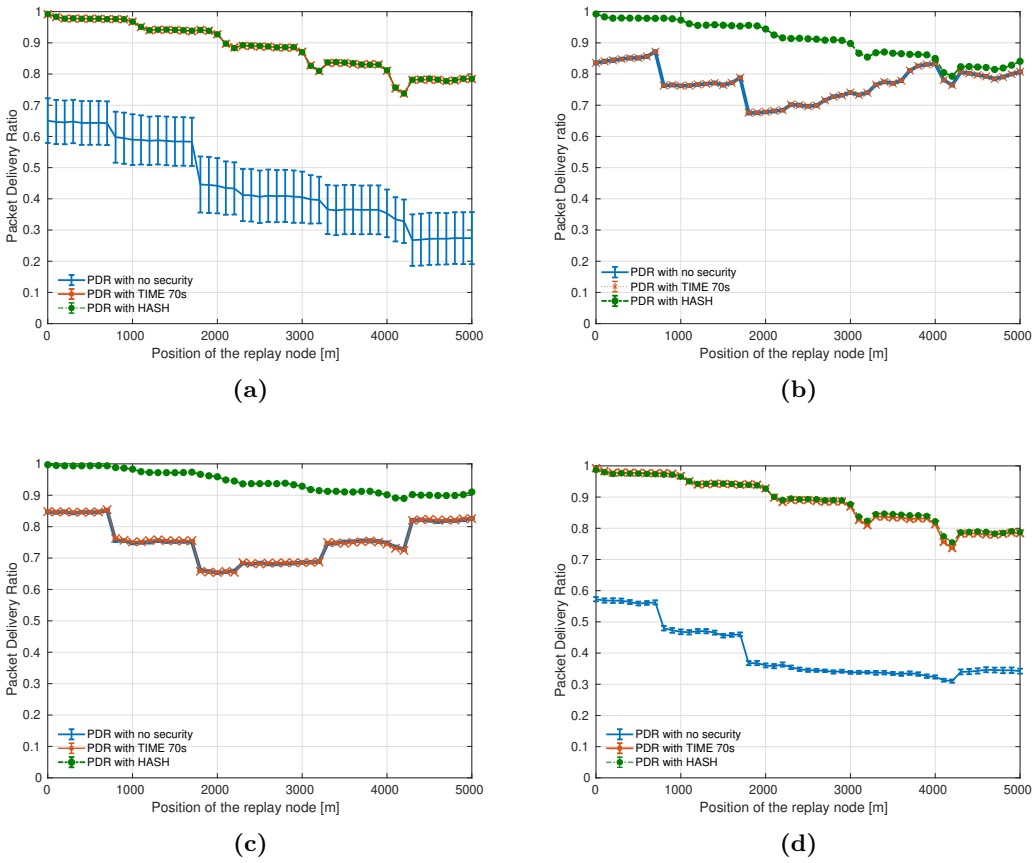


Figure 5.7: Packed delivery ratio of the network versus the replay node position in `NET1` with a TDMA MAC protocol and different countermeasures. (a): FIRST-PACKET; (b) LAST-PACKET; (c): MULTI-PACKET; (d) HOLD-PACKET.

this case the malicious node is attacking the network in an area that is closer to its bottleneck, i.e., the last node before the sink. Indeed, this node has more traffic to deliver to the destination. In this case the PDR decreases from 99% when the attacker is between the first and the second node, down to 80% when the attacker is placed close to the sink of the network, due to packet collisions caused by a collateral jamming effect. This effect is attenuated in the MULTI-PACKET attack (Figure 5.7c), as the packet transmissions are limited to the number of packets received by the attacker.

To investigate more in depth when HASH outperforms TIME as countermeasure of the HOLD-PACKET replay attack, we test different configurations of the attacker packet queue size and the HASH list size when the attacker is deployed between the sink and the last node before the sink, i.e., in the most advantageous configuration for the malicious node.

Figure 5.8 demonstrates that, as soon as the number of hashes stored in the HASH

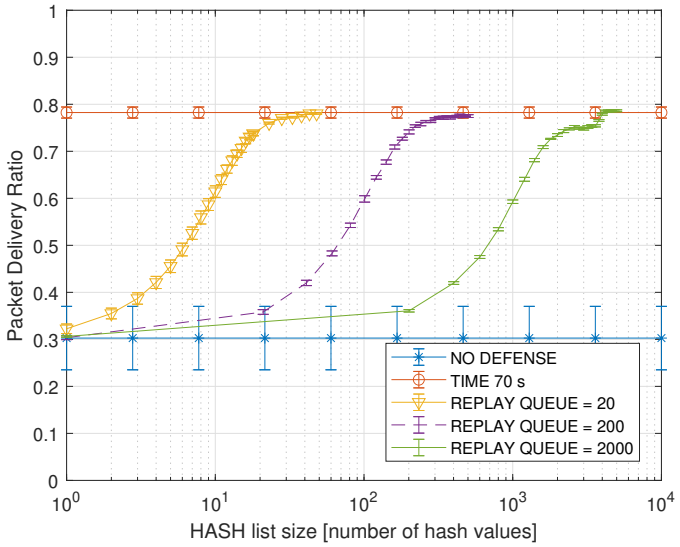


Figure 5.8: PDR of NET1 under HOLD-PACKET attack with different configurations of the attacker queue size and the HASH list stored in the nodes. The attacker is deployed between the sink and the last node before the sink.

list is greater than or equal to 2 times the number of packets recorded by the attacker, the performance of HASH and TIME are equal. We remark that while the attacker needs to store the whole packet, the defender requires to store only the HASH value, that is 4 Bytes per packet. For instance, if the attacker has a queue of 2000 packets, the nodes are required to allocate only 16 kBytes to store 4000 HASH values. Once this requirement is satisfied, in `NET1` HASH is never outperformed by TIME, for all attack strategies.

From our analysis, the same network with a CSMA MAC layer provides similar results, but is less interesting to analyze because such network configuration is less affected by the attack.

5.4.2 Replay Attack and Countermeasures in `NET2`

Also for scenario `NET2` we first analyze the impact of the replay attack when no security mechanisms are applied. Then, we evaluate the performance of the proposed countermeasures. With this scenario all nodes generate packets for all other nodes, with the intent of making the other nodes receive fresh position and data updates. In this scenario the data is carrying status updates and repeated over time, thus if some packets are lost a newer status with updated information can be obtained from subsequent messages. In this case we analyze the network performance in terms of average throughput per node, defined as the average number of payload bits that arrive to a node in a second, without counting the duplicates.

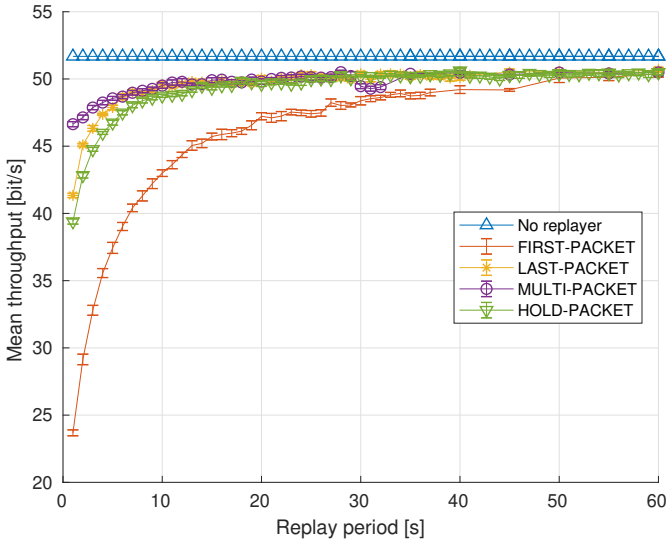


Figure 5.9: Average throughput received by one node versus the replay period in `NET2`.

5.4.2.1 Effect of the Replay Attack

The effect of the replay attacks in terms of average throughput received by one node when used in `NET2` is reported in Figure 5.9: these results are presented with 95% confidence intervals. The replay node does not have a MAC layer, and the effect of its attack is analyzed by varying the replay period, from 1 s to 60 s. Without the attacker, the throughput is 51.68 bit/s.

The MULTI-PACKET replay attack is the least destructive because it replays only once all the recorded packets, i.e., the maximum number of packets it transmits is bounded by the number of packets it receives. In addition, if the retransmitted packet was not received by one of the nodes in range due to packet collisions, when this packet is received by the application layer it is not discarded.

FIRST-PACKET, on the other hand, is the most destructive of the replay attacks considered in this chapter when no countermeasure is applied. This happens because this attack always injects the same old packet into the network, therefore increasing the network traffic without adding any packets that might have been lost by other nodes. In HOLD-PACKET, instead, after 6000 s of recordings where the attacker saves up to 2000 of the received packet in a buffer, it transmits packets randomly chosen from the 2000 packets stored: similarly to MULTI-PACKET, if the transmitted packet was not received by a node, it is not discarded by its application layer. HOLD-PACKET with a buffer size of 1 packet will behave instead like FIRST-PACKET: although at this point having a small buffer seems like the best choice, in the next section we will see that it is not, as the countermeasure of FIRST-PACKET is very easy.

Finally, with LAST-PACKET the attacker always replays the last packet it receives, almost acting as a relay of the flooding network. The node forwards the attacker packet

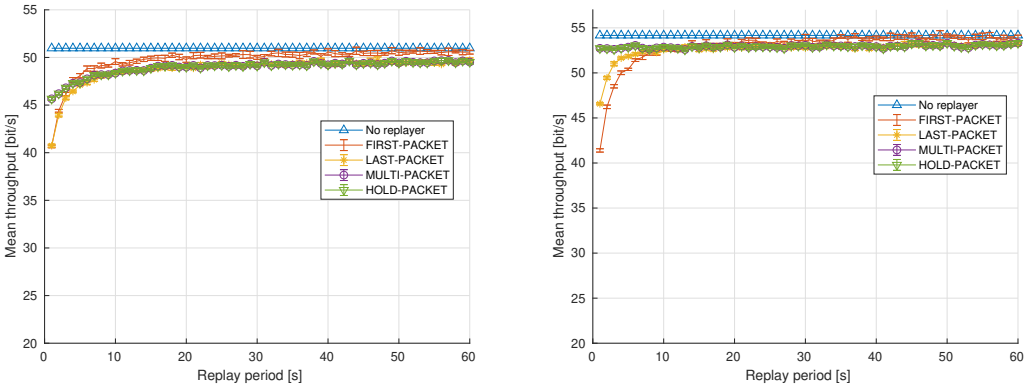


Figure 5.10: Average throughput received by one node versus the replay period in `NET2` with TIME (a) and HASH (b) countermeasures.

along the network, filling also the other nodes queue.

5.4.2.2 Replay Attack Countermeasures

Figures 5.10a and 5.10b demonstrate that TIME and HASH countermeasures are effective also in `NET2`, with flooding routing and mobile nodes. In this case, with TIME countermeasure the time validity of a packet is set to 150 s. TIME and HASH countermeasures perform similarly against FIRST-PACKET, providing a throughput increase up to 17 bit/s (70% throughput increase compared to the throughput without countermeasures) when the replay period is less than 5 s. Increasing the replay period, the benefits of the network with countermeasures compared to the network without countermeasures decrease, and so does the effect of the attack. A similar behavior is observed against the HOLD-PACKET attack, where both countermeasures provide a throughput increase of 5 bit/s (12.5% throughput increase compared to the throughput without countermeasures), when the replay period is less than 5 s. In this network, TIME does not provide benefits against LAST-PACKET and MULTI-PACKET, while for short replay period HASH provides a throughput increase of 5 bit/s (12%) against LAST-PACKET and of 8 bit/s (17.7%) against MULTI-PACKET.

To investigate when HASH outperforms TIME as countermeasure of the HOLD-PACKET replay attack, we test different configurations of the attacker packet queue size and the HASH list size also in the case of `NET2`, when the attacker uses a packet transmission period of 3 s. Figure 5.11 demonstrates that, as soon as the number of hashes stored in the HASH list is greater than or equal to the number of packets recorded by the attacker, HASH outperforms TIME. Once this requirement is satisfied, in `NET2` HASH is never outperformed by TIME, no matter the attack strategy.

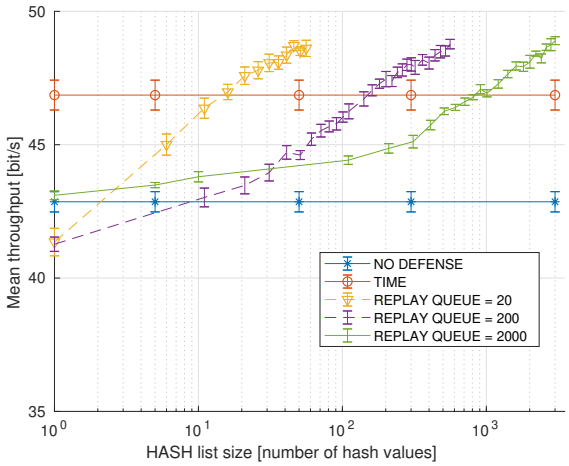


Figure 5.11: Throughput of NET2 under HOLD-PACKET attack with different configurations of the attacker queue size and the HASH list stored in the nodes.

5.5 Conclusion

In this chapter, we studied the effect of four different versions of the classical replay attack in UANs. We proposed two countermeasures, TIME and HASH. The former was based on the packet generation timestamp and the latter was based on the HASH value of the packet generation timestamp combined with the address of the source node. While the TIME solution is very effective if the attacker replays only old packets, generated several minutes (depending on the maximum packet delivery delay of the network) before the reception, the HASH countermeasure limits a lot the effect of the attack also in the case the attacker replays very recent packets. We demonstrated that the proposed countermeasures perform efficiently in a linear network with a unique sink and an static attacker as well as in a broadcast network with mobile/static nodes and a mobile attacker.

Chapter 6

Cross-Layer Communication System for Security in Underwater Acoustic Networks

6.1 Introduction

In this chapter we move a step forward with respect to Chapter 5, letting the attacker have full or partial knowledge about the protocol stack employed in the network and exploit this information to perform the attacks.

An extensive classification of DoS attacks in radio Wireless Sensor Networks (WSNs), along with the most common defensive techniques, is presented in [217] and [218]. Some of the DoS attacks that are performed in terrestrial radio networks are possible also in UANs. However, as discussed in Chapters 1 and 5, the countermeasures are often impossible to translate to the underwater acoustic domain, as its characteristics diverge too much from those of terrestrial radio. For example, countermeasures that are based on time validity of keys and certificates, that in general introduce a large overhead, cannot be directly applied to UANs due to the lack of a standard infrastructure for public key and certificate exchange. Therefore new studies and investigations are needed.

Countermeasures based on observations of the neighbor nodes' behavior [146], instead, can be easily applied thanks to the broadcast nature of the underwater acoustic channel, as they require low computational power and introduce a small overhead to the communication. In this chapter, we select two of the most common DoS attacks in radio WSN for which knowledge of the protocol stack is required: resource exhaustion and sinkhole attacks. We present a simulation study and a general countermeasure based on a `watchdog` layer able to overhear the packets transmitted by the neighbor nodes. A reputation system based on the analyzed behavior is then applied to identify possible attackers and exclude them from the network.

In Section 6.2 we present and describe types of attacks that are common in WSNs

and select those that are considered in this chapter, along with selecting the protocols to disrupt in the simulations. In Section 6.3 we propose countermeasures for each of these attacks and elaborate on the details for the selected scenarios. In Section 6.4 we illustrate the framework in which the simulations of both the attacks and their countermeasures are performed, and in Section 6.5 we present the simulation results. Finally, in Section 6.6 we draw our concluding remarks.

6.2 Protocol-Aware Attacks and Countermeasures

The attacks we investigate and analyze in this chapter fall under the categories of *resource exhaustion* and *sinkhole* attacks. This is a generic terminology that accounts for a broad array of different attacks, although they have similarities in the strategy, within the same group.

Resource exhaustion attacks, in particular, include a wide range of attacks in which the malicious node tries to deplete some resources that are necessary for the attacked nodes to operate. In [219] the authors define a resource exhaustion vulnerability as a specific type of fault that causes the consumption or allocation of a resource in an undefined or unnecessary way, or the failure to release it when no longer needed, eventually causing its depletion; they simulate a wide array of attacks for the Domain Name System (DNS) in order to discover vulnerabilities. From the survey performed in [113], it emerges that the limited resources available at the sensor nodes are often subject to attacks that try to exploit all these weaknesses. Channel access and availability play an important role in these types of attacks and on the ability of an attacker to disrupt normal operation of the network. Among the proposed taxonomy of defenses, the watchdog type of countermeasures emerge as one of the most effective and lightweight.

The second type of attack we consider is the sinkhole attack. In this type of attack, the malicious node tries to convince the nodes of the network to route the traffic though itself. Then it can perform dropping or more sophisticated traffic analysis. From this attack, more complex strategies can be mounted such as the *wormhole* attack [209, 210]. The survey in [220] remarks the necessity to design routing protocols with security in mind and argues that this is something rarely addressed. The work performs a survey of some of the most common WSN routing protocols, common attacks and countermeasures: the conclusion is that a defense mechanism against the sinkhole attack is unlikely to be designed, and that the only possibility is to design the routing protocols around the sinkhole attack countermeasure. The countermeasure taken in [112], which is one of the most common in the literature, is quite expensive in terms of overhead as it is based on specific signaling messages dedicated to the attacker detection. Specifically, it proposes an algorithm that relies on the presence of a data collection point (sink node) to detect a possible ongoing sinkhole attack involving flooding the network and retrieving information.

6.2.1 Resource Exhaustion Attack on UW-POLLING

To test the resource exhaustion attack and its defense, we chose UW-POLLING, a polling-based MAC protocol that can be used for data muling applications (see Chapter 2).

We simulated two types of attacks. In the first type, the attacker is only able to store the overheard packets and replay a subset of them; in detail, the attacker is able to discriminate the different types of signaling packets and performs the attack by replaying: trigger packets, poll packets and probe packets. In the second type of attack, a smart attacker is able to generate legitimate signaling packets; in this case, the attacker's goal is to exhaust the channel access allocation by asking the sink the permission to transmit a large amount of packets, without actually transmitting them; therefore, the sink always polls the attacker first, due to its fair policy, thus resulting in less time for the other nodes in range to transmit their data.

We assume two types of attacks for this protocol:

- a malicious node replaying only a selected subset of the recorded packets;
- a malicious node generating deceptive legitimate packets that cause the sink to assign excessive resources to the attacker.

6.2.2 Sinkhole Attack on SUN protocol

To test the *sinkhole* attack we chose the SUN protocol [151]. This is a Dynamic Source Routing (DSR) protocol based on hop-count for route determination and includes two types of nodes: a sink node that collects information and normal nodes that generate this information. The basic mechanism of SUN is described below:

1. the sink sends probe packets to allow in-range nodes to be aware of being one-hop away from the sink;
2. nodes that receive probe packets, called *end-nodes*, have now a route to the sink node, of length one hop;
3. a node that has data to send, and does not have any route to the sink broadcasts a path request;
4. a node receiving a path request can perform one of the following actions:
   (a) if it has no route to the sink, it adds its own address to the request packet and broadcasts again the packet;
   (b) if it is an end-node, it adds its own address to the packet and sends back a path establishment reply, using the reverse route;
   (c) if it has a route to the sink but it is not an end-node, it proceeds as in case (a);

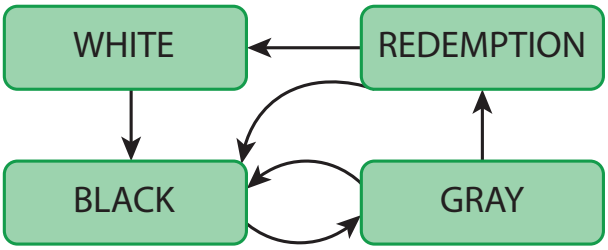


Figure 6.1: State machine of the reputation system.

5. the path establishment reply packet, containing all the nodes that the corresponding request passed through, reaches the source node, allowing it to have a route to the sink node.

The malicious node will always advertise itself as an end-node, i.e., as being one hop away from the sink node. Then, the attacker will drop either all or some of the packets received by the other nodes, instead of forwarding them: in this chapter, the effect of the attack is analyzed by varying the position of the malicious node and the percentage of packets that it drops.

6.3 Defense framework and strategies

The defense strategy we present in this chapter develops upon two fundamental mechanisms: a `watchdog` layer, able to overhear and send the overheard data to the upper layers, and a reputation mechanism, able to label nodes as *suspicious* or *trustful* and thus to choose which ones to rely on.

The `watchdog` layer is placed between the MAC and the physical layer and is able to overhear all communications of the nodes in range. Via cross-layer interaction, this layer can notify all other layers of the stack when it overhears a packet and is the base of the security architecture employed. The interested layers collect this information and use it in their specific defense strategy.

In order to detect and counteract the ongoing attack, a reputation system which makes use of the `watchdog` security layer and of the shared cross-layer information is employed. Each protocol of the communication stack implementing the required security mechanism needs to perform the following operations:

1. check which node transmitted the overheard packet received via cross-layer;
2. decide, according to its protocol rules, whether that node is behaving in a good (`G`) or in a bad (`B`) way;
3. report the node's behavior to the reputation system.

We want to highlight that the good (`G`) and bad (`B`) ways are protocol dependent, and the reputation system only needs to know the final results of the behavior, i.e., `G` or

B. Figure 6.1 shows the state machine employed by the reputation system illustrated in this chapter. Then the reputation system, based on its implementation, decides whether that node should be trusted or not by using white, grey, redemption and black lists. The transition between white, gray, redemption and black is generic and can change depending on the implementation of the reputation system. We present the implementation we used for the state transitions as follows.

- Initially, all nodes are in the white list and have a total trust $s_i = S_{max}$, that is the maximum trust score.
- Each time the behavior of a node x is B, s_x is decremented, and each time the node behavior is G for n_{res} consecutive times, s_x is restored to the maximum value S_{max}.
- If s_x becomes 0, node x is blacklisted.
- Depending on the layer rules, after a certain event (e.g., after a timeout elapses, or after 10 selections of other nodes as the next hop of the network), a blacklisted node x is moved to the gray list in order to give x a second chance.
- A node x in the gray list is moved to the black list as soon as it behaves in B way, instead if its behavior is G for n_{gr} times it is moved to the redemption list
- A node x in the redemption list has a total score of $s_x = S_{max}/2$: each time its behavior is B, s_x is decremented, and if s_x becomes 0, x is blacklisted. Instead, if its behavior is G for n_{rw} times, it is moved to the white list and gains back a total score of S_{max}.

6.3.1 Resource Exhaustion Countermeasures

To counteract the first attacker strategy based on the replay of recorded packets, a security mechanism based on the HASH freshness index is applied, as described in Chapter 5: a security layer computes the XOR operation between the HASH of the packet generation time and the HASH of the network address of the source node, for a total size of 4 Bytes, and inserts it in the packet header, for the receiver to certify its validity. For the second type of attack, based on the generation of legitimate packets, two complementary and subsequent phases of the countermeasure system are employed. The first phase is a preliminary check used to immediately exclude from the network those nodes that ask to transmit too many data packets (thus, with a highly suspicious behavior); specifically, when the amount of time to be allocated to a single node is below a given threshold $T_{tx,ths}$ the packet is directly inserted in the poll list, whereas, if the amount of time exceeds the threshold, the node is either not inserted in the poll list (BAN-NODE mechanism), or inserted but reducing the amount of allocated time for it (POLL-NODE mechanism). The second phase, that consists in the actual countermeasure, is based on the reputation system presented in Section 6.3 and comes into play when the first countermeasure is not applicable, i.e., when the number of packets the

node is asking permission to transmit is not immediately suspicious. Indeed, a smarter attacker can bypass the former countermeasure by reducing the number of packets it asks to transmit. The reputation system gives a score to each node and reduces it when there is a difference between the intended transmissions and the actual packets transmitted. Eventually, the malicious node is blacklisted by the reputation system and no longer considered in the polling phase.

6.3.2 Sinkhole Countermeasures

The defense mechanism is based on overheard packets: after a node asks a neighbor to forward a packet, it observes whether that neighbor forwards that packet within a certain timeout by using the `watchdog` layer. If the packet is not overheard, the node assumes that the neighbor dropped that packet and decreases the trust score of this neighbor. Conversely, if the source node overhears a correctly forwarded packet, it increases the score of the relay node. The increase/decrease of reputation score operation is performed using the reputation mechanism presented in Section 6.3. When a source node needs to send data, there could be two cases: in case the node has no route to the sink it begins the path establishment process that will be analyzed later; in case the node has a route to the sink, it verifies the trust of the first hop: if the node is blacklisted (i.e., it has a bad reputation), the route is discarded and another path establishment process is started. The path establishment process begins with a path search packet to which, hopefully, a number of nodes reply with a path answer packet. Once receiving the reply messages, the source node will verify the trust of the replying nodes and discard all the routes coming from blacklisted nodes. Among the remaining ones, the path with the minimum number of hops is selected. Otherwise, if no trusted route is received within a fixed timeout, a new path establishment process is started.

6.4 Simulation Scenarios and System Settings

The DESERT Underwater simulator [149] is used to simulate the two selected types of attacks and their countermeasures. The characteristics of both the devices and the environment are slightly different between the resource exhaustion attack and the sinkhole attack simulations, to better highlight the key behaviors.

In the simulation of the *resource exhaustion* attack, all nodes are equipped with low frequency acoustic modems operating in the 7-17 kHz bandwidth, their transmission rate is 500 bit/s and their simulated range is 4 km.

In the first scenario the attacker is not able to create legitimate packets but can discriminate their type, namely: `data`, `trigger`, `probe` and `poll` packet. We simulate the disruption of the UW-POLLING protocol replaying, separately, `trigger`, `probe` and `poll` packets: the topology is presented in Figure 6.2. The nodes are deployed in 4 clusters composed by 3 legitimate nodes each. In the second cluster an attacker is placed close to the legitimate nodes, in order to be able to record and repeat the signaling packets generated within its cluster or by the AUV. This setup allows to

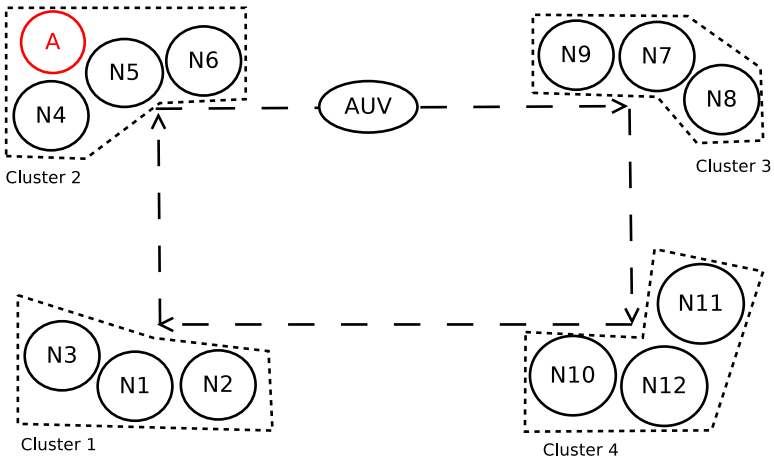


Figure 6.2: UW-POLLING attack simulation topology.

quickly compare the behavior of the attacked cluster with the clusters close to it and the one not reached by the attacker. The AUV moves at a constant speed of 1.67 m/s in the network following a square path, moving among the clusters and collecting data from the nodes. In this configuration, all nodes generate a packet every 120 seconds with a size of 60 Bytes. We considered 3 cases in which the attacker retransmits packets of different sizes: in the first case, only `trigger` packets are retransmitted; in the second case, only `poll` packets are retransmitted; finally, in the third case, only `probe` packets are retransmitted.

In a second scenario, a smarter attacker is able to forge legitimate packets. In this case, the attacker's goal is to exhaust the channel access allocation by asking the sink permission to transmit a large number of packets, and never transmitting them. Therefore, at each trigger cycle when the attacker is in range with the sink, the sink always polls the attacker first due to its fair policy, giving the other nodes in range less time to transmit their data.

For what concerns the simulation of the *sinkhole* attack, we test it on the SUN network protocol using the topology shown in Figure 6.3. The attacker, depicted in red, is first placed in position A1 and then in position A2 to inspect if the attacker's position affects the network in different ways. All the devices involved are operating in the 7-17 kHz frequency range, their bitrate is 500 bit/s and the communication range for each node is about 4 km, therefore nodes 1, 2 and 3 are in communication range with the sink while node nodes 4, 5 and 6 require the usage of relay nodes. Node 4 is in communication range with 1, 2 and 3. For nodes 5 and 6 (two AUVs), the link quality to reach the other nodes is impacted by the vehicles' mobility. As previously specified, the goal of the attacker is to attract the largest possible amount of traffic from the network: then, it can decide to drop some of this traffic.

The nodes generate packets at a constant rate and size: the characteristics of the various types of traffic, generated by the nodes in the scenario in Figure 6.3, are detailed in Table 6.1.

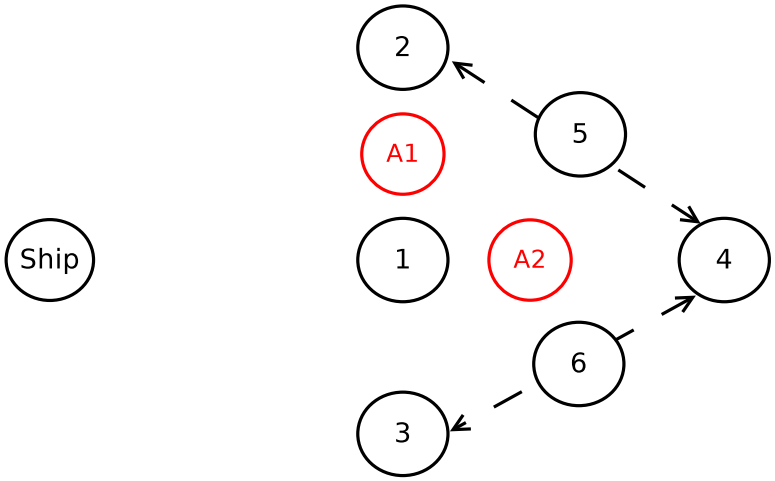


Figure 6.3: Sinkhole attack simulation topology.

Table 6.1: Traffics of the nodes deployed in the scenario of Figure 6.3

Traffic	Generating Node	Packet size [Bytes]	Generation period [s]
T1	1, 2, 3, 4	32	120
T2	5, 6	60	120
T3	5, 6	120	80

6.5 Results

6.5.1 Resource Exhaustion Attacks

We now analyze the impact of the *first resource exhaustion attack*, performed by a malicious node that replays signaling packets, on the throughput of the network as a function of the replay time T_{replay}, i.e., the average time between two consecutive attacker's transmissions.

Figure 6.4a shows the overall throughput, for Cluster 2 of the topology depicted in Figure 6.2, as a function of T_{replay}, when the attacker retransmits trigger packets. This type of attack has small effects for $T_{replay} > 15$ s (green line). Conversely, the throughput of Cluster 2 drops to 7 bit/s when $T_{replay} = 10$ s, and further decreases down to 3 bit/s when T_{replay} decreases to values < 10 s. When the security mechanism based on the HASH freshness index is enabled (red line), the throughput of the network is almost equivalent to the case without attack (blue line), confirming the effectiveness of this countermeasure.

Figure 6.4b depicts the throughput of the node affected by the periodic replay of the same poll packet. The poll packet is sent in unicast, therefore when a poll packet is replayed only one node is attacked, i.e., only the intended node of the original poll packet. Figure 6.4b shows that the throughput of the attacked node is affected even

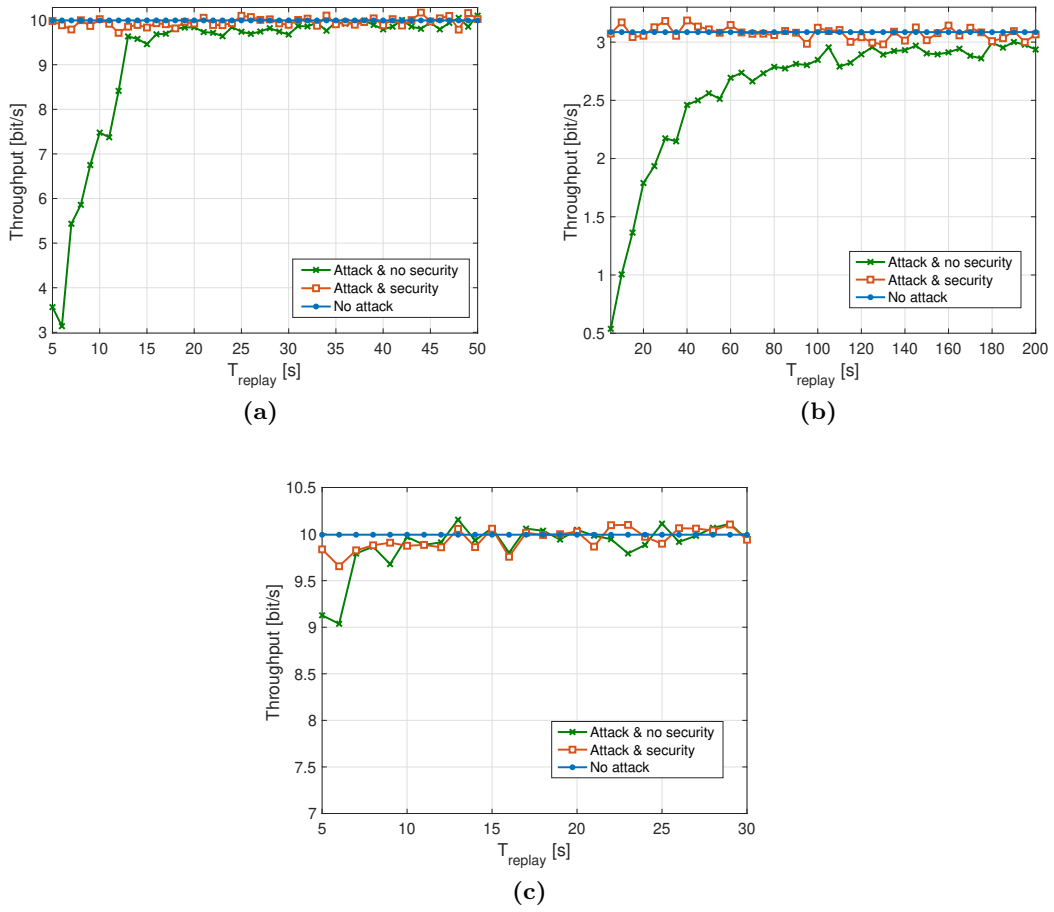


Figure 6.4: (a) Throughput of Cluster 2 when the malicious node repeats trigger packets; (b) Throughput of the attacked node when the malicious node repeats poll packets; (c) Throughput of Cluster 2 when the malicious node repeats probe packets

with a relatively high replay period, i.e., with $T_{replay} < 120$ s. As for the trigger packet attack, the security mechanism based on HASH freshness index completely mitigates the effectiveness of this attack. Differently from the previous scenarios, the replay of a probe packet does not affect the performance of the network (Figure 6.4c) except for small values of T_{replay}, where there is a small drop in the overall throughput. Also in this case, the HASH index provides a valid countermeasure, preventing the effects of the attack.

The effects of the *second resource exhaustion attack*, performed by a node able to generate legitimate probe packets, are presented in Figure 6.5. The figure shows the overall throughput of Cluster 2, i.e., the cluster where the attacker is located. The green line represents the overall throughput of Cluster 2 when the network is under attack and without defense. The analysis has been done as a function of the number of packets

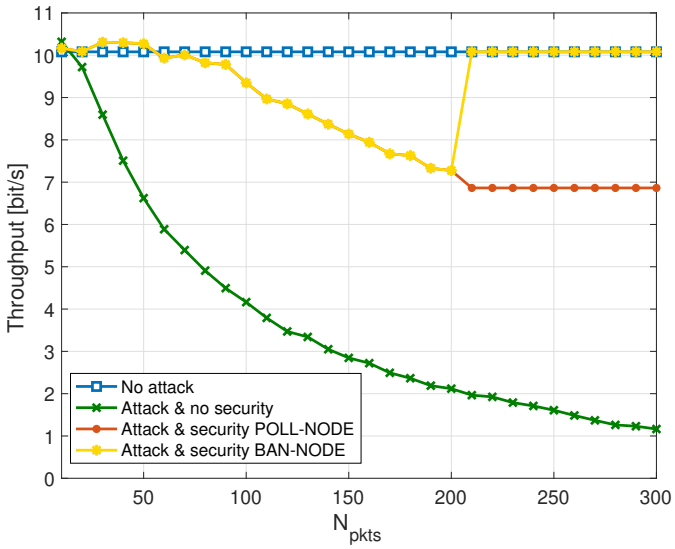


Figure 6.5: Overall throughput of the second cluster for the resource exhaustion attack

the attacker claims to transmit in the probe packets (N_{pkts}). The larger the value of N_{pkts}, the higher the amount of time reserved to the attacker, and, therefore, the lower the time available for the other nodes to be served by the AUV. In this scenario, the overall throughput of the cluster is reduced by 30% when $N_{pkts} = 50$ and by 60% when $N_{pkts} = 100$. Using the security mechanisms, the drop in performance is limited. When $N_{pkts} > 200$, the amount of time that should be reserved to the attacker exceeds the time threshold $T_{tx,ths}$. If the BAN-NODE mechanism is employed (yellow line), the node is automatically excluded from the network. Indeed, the performance becomes comparable to the performance without an attacker (blue line). When the POLL-NODE mechanism is employed, there is a reduction in the overall throughput, since an amount of time equal to the threshold $T_{tx,ths}$ could be reserved for the attacker. Still, the effects of the attacker are mitigated thanks to the reputation system. Indeed, when no data packets are received from the attacker, its reputation is reduced as long as the node is inserted in the black list. From that point onward, except for the redemption attempts, the node is not considered and $T_{tx,ths}$ s are no longer reserved. When $N_{pkts} \leq 200$, the barrier does not come into play, and only the reputation system can prevent the malicious node from affecting the network performance. When $N_{pkts} \leq 100$, the performance of the network with the security mechanism is almost equivalent to the network without attack. For higher values of N_{pkts} there is a drop in performance, but still much lower than the drop without any security countermeasure. The drop with the security mechanism is due to the fact that the reputation system takes time to identify the malicious node and put it in the black list. In the meantime, the attacker still induces the AUV to reserve the channel to it, thus reducing the channel availability for the other nodes in the cluster.

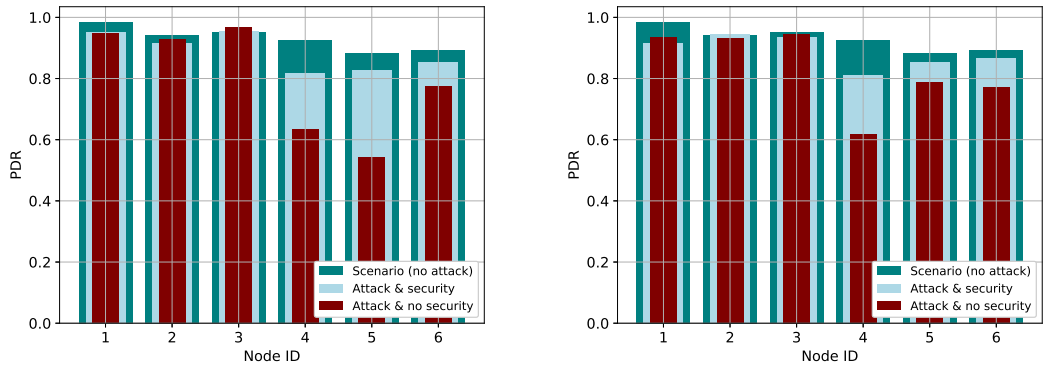
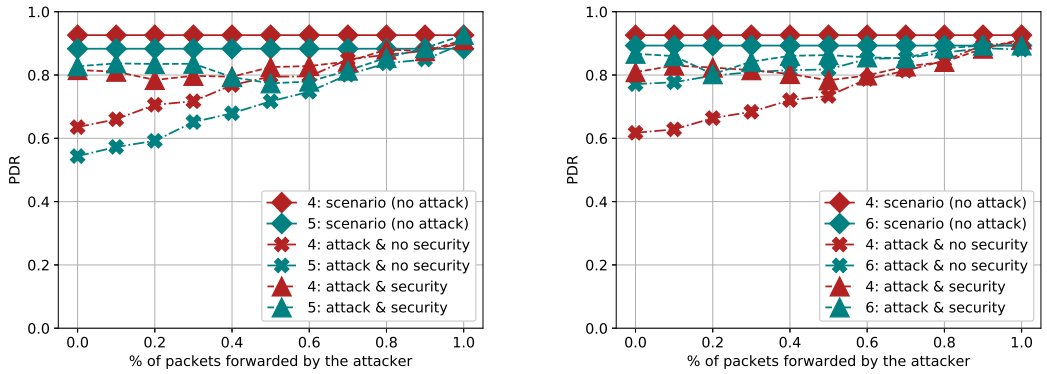


Figure 6.6: Sinkhole attack: PDR with attacker in position A1 (a) and A2 (b)

6.5.2 Sinkhole Attack

Concerning the sinkhole attack against the SUN network protocol, Figure 6.6a and Figure 6.6b present the PDR of the nodes in the network when the attacker is in position A1 and A2, respectively. In both plots the attacker drops 100% of the received packets. When the attacker is in position A1, the most affected nodes are node 4 (static node) and node 5 (the first AUV), whose PDR drops from 93% to 64%, and from 88% to 54%, respectively, when the network is under attack. With the countermeasure enabled, the PDR increases to 82% for node 4 and to 83% for node 5, thanks to the fact that the malicious node is detected and blacklisted by the reputation system. When the attacker is in position A2, the most affected node is node 4, whose PDR drops from 93% (without attack) down to 62% (ongoing attack). With the security enabled, the node's PDR increases to 81%. For both cases, the nodes most affected by the attack are those that are not in range with the sink, and need to find possible routes. In this scenario, the most affected node is node 4, while the other nodes are only partially affected by the attacker since most of them have a direct path toward the sink (node 1, 2 and 3). Nodes 5 and 6, instead, are sometimes able to obtain a valid path toward the sink also thorough nodes 2 and 3 thanks to their mobility, therefore avoiding the attacker in A2. The capability to detect the presence of the malicious node and avoid its selection as relay node has a significant impact on the data delivery. From these figures, it can be seen that some nodes improve their performance in case of attack and no countermeasure deployed: this happens because the network is loaded with heavy traffic, thus, nodes close to the sink have to both send their packets and relay packets for other nodes. When the attacker is in position A1, and drops most or all of the packets, its behavior is beneficial to those nodes whose traffic is mostly relayed for other nodes, namely node 1, node 2 and node 3: in the first two cases the PDR does not undergo a noticeable change, whereas, in case of node 3, it is even increased.



(a) **(b)**

Figure 6.7: Sinkhole attack: PDR of the most affected nodes when the attacker is in position A1 (a) and A2 (b)

When the attacker is in position A2, the PDR of the nodes close to the sink does not vary notably.

Figure 6.7a and Figure 6.7b show the PDR of the nodes most affected by the attack when the attacking node varies the percentage of dropped packets, respectively, in position A1 and position A2. When the attacker is in position A1 it is able to effectively draw traffic from both node 4 and node 5. Figure 6.7a shows that the defense mechanism is able to restore the PDR to values very close to the scenario without attack. Figure 6.7b shows that, when the attacker is in position A2, only node 4 is heavily affected by the attack, as node 5 is now able to obtain a valid route from node 2 more frequently. In addition, the defense mechanism restores the PDR to values close to the no-attack scenario. In both cases, it can be seen that the most challenging situation for the defense mechanism is when the dropped percentage is in mid-range. When the attacker is dropping all packets it is easier for the other nodes to detect the ongoing attack. Conversely, when the attacker relays most or all of the packets, it is harder for other nodes to detect the attack but the PDR is closer to no-attack values.

6.6 Conclusions

The proposed defense mechanism, based on a `watchdog` layer and a *reputation system*, proved to be effective in counteracting the two types of attacks analyzed. In particular, it is able to prevent the attacked nodes from being excluded completely from the network, guaranteeing a minimum level of participation in the network communication. This includes defending against attacks that exploit knowledge of the communication protocol stack. Specifically, a countermeasure based on a packet freshness index has been proved as a valuable solution for replay attack of the signaling packets of the UW-POLLING protocol, whereas a reputation based system can limit the effect of sinkhole

and resource exhaustion attacks. Furthermore, the defense framework, implemented in DESERT Underwater [149], proved to be very extensible and configurable, allowing to tailor the general structure to the attacks under analysis, in both the `watchdog` layer and the trust mechanism.

6. CROSS-LAYER COMMUNICATION SYSTEM FOR SECURITY IN UNDERWATER ACOUSTIC NETWORKS

Chapter 7

Trust Model for Security in Underwater Acoustic Networks

7.1 Introduction

To infer the reputation, a node needs to observe the behavior of its neighbors by overhearing transmissions by the other nodes. This solution is widely used in terrestrial wireless networks, where a node can exploit the broadcast nature of the channel to understand the behavior of its neighbors, by overhearing the packets they transmit. This mechanism can be referred to as watchdog [146], implicit ack, or overhear forwarding mechanism. In this chapter we present a trust mechanism for underwater networks, where we adapt approaches used in terrestrial wireless networks to the disruptive nature of the acoustic channel by using subjective logic [147] to take into consideration the uncertainty caused by the acoustic channel quality evolution. The use of subjective logic to provide a trust measure for the nodes of a network has been already employed in terrestrial networks [116, 221], however the peculiarity of the acoustic channel poses a further challenge in the analysis of the node trust. Indeed, bursts of packet errors characterize acoustic communications due to the multipath, high delay spread and ambient noise of the acoustic channel. For these reasons the trust models designed for the terrestrial counterpart do not take into account channel measures to compute the reputation. On the other hand, other methods based on bayesian models, often employed to compute the trust in terrestrial networks [148, 222], are not applicable to underwater networks because they do not consider any link disruption caused by the variability of the acoustic channel. The risk of this approach is to estimate the reliability of the channel rather than the trustworthiness of a node.

The main contribution of this chapter is the design of a trust module that provides a measure of trust of the one-hop neighbors. Our proposed framework is: *i)* general enough to tackle different types of attacks; *ii)* specifically tailored for underwater acoustic networks. Indeed, as soon as the concepts of correct behavior and misbehavior are well defined, the model can be employed to discover different attacks, such as

sinkhole and resource exhaustion. Moreover, the unique characteristics of the underwater acoustic channel are taken into account in the trust model by considering channel based metrics, such as noise and SNR, and modeling the acoustic channel variability with a Hidden Markov Model (HMM) [223]. The proposed system is evaluated both analytically and through simulations based on real field measurements, using the flooding network protocol in the DESERT Underwater network simulator [149]. The trust model can also be used in real-field application such as in underwater acoustic networks employed for coast surveillance that make use of the Gossiping in Underwater Acoustic Mobile Ad-hoc Networks (GUWMANET) protocol [19] specifically tailored for military networks.

The rest of this chapter is organized as follows. Section 7.2 presents the channel model used to characterized the variability of the acoustic channel, while Section 7.3 describes the trust model based on subjective logic and provides the analytical formulation of the problem. Section 7.4 presents the scenario and Section 7.5 discusses about the analytical and the simulation results in terms of attacker detection capability of the presented trust model. Finally, Section 7.6 draws some conclusions.

7.2 Channel Model

Given the disruptive nature of the acoustic channel, where an acoustic link between two nodes may present only a small packet loss for several hours, then present a high packet loss for a few hours, and then return stable again, it is not trivial to understand when a drop of performance of an acoustic network is caused by a DoS attack or by bad channel conditions. The increase of packet loss can be caused by several factors [5], for example the increase of noise caused by a ship travelling close to the network deployment, by the presence of strong rain and wind, or by the presence of shallow zones caused by a temperature drop and the consequent change of sound speed profile [224].

In this chapter, we model the acoustic channel according to the statistics of sea trial measurements. Indeed, in the last fifteen years researchers [223, 225] demonstrated that the time evolution of underwater acoustic channels can be statistically well characterized with two- and four-state Markov models [226] and with a two-state HMM [227]. The use of Markov models to characterize the behavior of the channel is well-known also in terrestrial networks [227, 228, 229, 230] where the transition probabilities from the states of the Markov Chain are usually obtained exploiting well-established statistical channel characterizations such as Rayleigh fading or Rician fading channel models [231]. In underwater acoustic networks, instead, there is no commonly accepted statistical model for the channel behavior, since the channel is strongly affected by the local environmental conditions of the network deployment. Therefore, in acoustic networks the parameters of the Markov model are often inferred from experimental measurements. An evaluation of the three Markov models (the two Markov models of [226] and the HMM of [227]) compared with sea trial measurements is presented in [223]. The discussion on which model best fits the experimental data is carried out considering relevant metrics for networking, i.e., PER, length of error bursts and corre-

lation of errors after a given number of packet transmissions. Results show that HMMs yield an accurate reproduction of the channel metrics, tracking well long term channel behaviors.

For this reason, we decided to model the acoustic channel with a two-state HMM, following the work presented in [223]. The trust model presented in this chapter is built on top of the Markov channel model. In an HMM, the observable events stay on the top of a non-observable structure, the MC. The underlying, non-observable link model is a two-state MC that defines two states for the goodness of the channel, specifically a GOOD (g) and a BAD (b) state, collected in the set $\mathcal{S} = \{g, b\}$. The probability of receiving a transmitted packet is o_g in GOOD state, and o_b in BAD state, with $o_g > o_b$. The MC is described through the transition probability matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \begin{pmatrix} P_{gg} & P_{gb} \\ P_{bg} & P_{bb} \end{pmatrix}, \tag{7.1}$$

where P_{ij} is the probability of moving from state i to state j in one step, with $i, j \in \mathcal{S}$. Figure 7.1 shows the two-state MC, where $P_{bb} = 1 - P_{bg}$ and $P_{gg} = 1 - P_{gb}$.

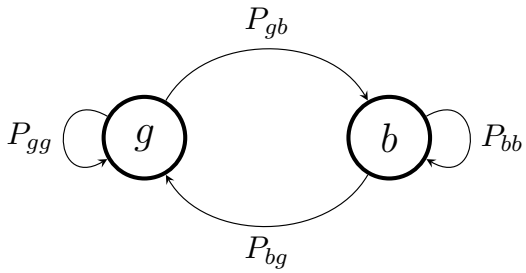


Figure 7.1: Two-state MC.

The n-step transition matrix $\boldsymbol{P}^n$, can be computed as described in [232]

$$\boldsymbol{P^n} = \frac{1}{P_{gb} + P_{bg}} \begin{pmatrix} P_{bg} & P_{gb} \\ P_{bg} & P_{gb} \end{pmatrix} + \frac{(1 - P_{gb} - P_{bg})^n}{P_{gb} + P_{bg}} \begin{pmatrix} P_{gb} & -P_{gb} \\ -P_{bg} & P_{bg} \end{pmatrix}, \tag{7.2}$$

with n the number of steps after which the system described by the MC is observed again. We denote the state visited at step n as $X_n = s \in \mathcal{S}$. The steady state probability vector $\boldsymbol{\pi} = [\pi_g, \pi_b]$, with $\pi_g + \pi_b = 1$, does not depend on the initial state, and can be computed from the transition probability matrix, specifically,

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} \pi_g & \pi_b \end{pmatrix} = \lim_{n \to \infty} \boldsymbol{P^n} = \frac{1}{P_{gb} + P_{bg}} \begin{pmatrix} P_{bg} & P_{gb} \\ P_{bg} & P_{gb} \end{pmatrix} \tag{7.3}$$

where the last equality holds because $1 - P_{gb} - P_{bg}$ is smaller than 1 and therefore the second term in Equation (7.2) goes to 0, thus obtaining $\pi_b = \frac{P_{gb}}{P_{gb}+P_{bg}}$ and $\pi_g = \frac{P_{bg}}{P_{gb}+P_{bg}}$. P_{gb}, P_{bg}, o_g and o_b can be either set manually, in order to study the system in a synthetic channel, or obtained from sea trials measurements [223].

7.3 Trust Models

7.3.1 Subjective Logic

The trust model presented in this chapter is based on the observation of the behavior of the neighbors, through the so called watchdog mechanism. In general, when a node of the network receives a packet, it needs to perform a task. For example, if node A sends a packet to node B, and B is not the final destination, B could be required to forward that packet either once or several times, depending on the routing protocol used in the network stack. The packet forwarded by B will be overheard by all of B's neighbors, including A. From A's point of view, B's task will be correctly accomplished if A overhears B's packet, or result in a misbehavior if A does not. Depending on the running protocols and applications, tasks can be different, from the forwarding of a single packet, to the transmission of a series of packets whose number and inter-transmission times depend on the running application. Independently of the task, the final result will always be a decision by node A about whether B's action corresponds to a correct behavior (C) or a misbehavior (M). In this chapter, a misbehavior is defined as the observation (or the lack of an observation) by a node about an action from the neighbor not compliant with the protocol rules, regardless of whether the action was intentionally carried out by the neighbor or caused by bad channel conditions. For example, if a neighbor is required to forward a packet and the node does not overhear its neighbor's transmission, this will be considered as a misbehavior regardless of whether the packet was intentionally dropped by the neighbor or it was actually transmitted by the neighbor but not overheard by the node because of bad channel condition. The goal of trust model is to be able to distinguish between intentional misbehavior and unintended misbehavior caused by channel loss that in principle are not discernible by the overhearing node.

In an underwater environment, directly applying the output of the watchdog mechanism to compute the trustworthiness of a node could cause misleading conclusions due to the variability of the channel described in Section 7.2. Indeed, a result purely based on the observation of a neighbor without any distinction on the channel quality could lead to a judgement related to the channel quality rather than to the actual node behavior.

In our model, we distinguish when a certain behavior occurs in GOOD and BAD channel state, weighing the two cases differently to obtain an opinion about the node's behavior according to subjective logic [147]. Considering the channel model described in Section 7.2, we model the behavior of a neighbor node through an HMM in which the observable events are the correct behavior (C) or the misbehavior (M) of a node and we denote this set as $\mathcal{E} = \{C, M\}$. For each state $s \in \mathcal{S}$ and each observable event $e \in \mathcal{E}$, the probability of observing the event e in state s is defined as $o_s(e)$, with the constraint $\sum_{e \in \mathcal{E}} o_s(e) = 1$ for each value of $s \in \mathcal{S}$. We emphasize that for a well behaving node, the values of $o_s(M)$ and $o_s(C)$ are related to the probability of not overhearing the packet transmitted by the neighbor, and therefore depend on the packet error probability. Consider as an example a node that has to forward a packet

due to the routing protocol rules: $o_g(C)$ is the probability of overhearing a packet transmitted by a legitimate forwarding node in GOOD channel conditions, and $o_b(C)$ the probability of overhearing a packet transmitted by a legitimate forwarding node in BAD channel conditions.

To compute the trustworthiness of a node we use subjective logic. Subjective logic deals with uncertainty and can be used to represent an *opinion* about a given statement (in our case whether a node can be trusted or not). The opinion is defined as the tuple $\mathbf{o} = \{b, d, u\}$, where $b, d, u \in [0, 1]$ and $b+d+u = 1$. Specifically, the three terms refer to belief, disbelief and uncertainty, respectively. The main idea is to update belief, disbelief and uncertainty based on the outcome, i.e., a correct behavior or a misbehaviors, of the analyzed nodes. The opinion depends on the number of misbehaviors $m = m_b + m_g$ and correct behaviors $c = c_b + c_g$, where m_i and c_i with $i \in \mathcal{S}$ are the number of correct behaviors and misbehaviors observed by a node in channel state i. Belief, disbelief and uncertainty can be computed as:

$$\begin{cases} b = \dfrac{w_{cg}c_g + w_{cb}c_b}{c+m} \\ d = \dfrac{w_{mg}m_g + w_{mb}m_b}{c+m} \\ u = \dfrac{(1-w_{cg})c_g + (1-w_{cb})c_b + (1-w_{mg})m_g + (1-w_{mb})m_b}{c+m} \end{cases} \tag{7.4}$$

where $w_{ij} \in [0, 1]\ \forall i \in \{M, C\}\ \forall j \in \{b, g\}$ are the weights to use for a correct behavior (C) or a misbehavior (M) in a GOOD (G) or a BAD (B) channel state.

In addition, the weights used to compute belief, disbelief and uncertainty can be composed by a fixed part, decided a priori, and a variable part that takes into account the trend of the behaviors of the neighbor to adjust the overall weights for correct behaviors and misbehaviors. Specifically, the weights related to a misbehavior, in both GOOD and BAD channel, are defined as

$$w_{mi} = \alpha\tilde{w}_{mi} + (1-\alpha)w_{var} \qquad i \in g, b, \tag{7.5}$$

where w_{var} is a function of the behaviors of a neighbor, whose goal is to grasp some signs about anomalous behavior (e.g., number of misbehaviors in GOOD channel higher than number of misbehaviors in BAD channel) that could be the manifestation of a misbehaving node, and therefore to penalize that neighbor increasing the weight for each misbehavior. Similarly, the weights for the correct behavior, in both GOOD and BAD channel, can be defined as

$$w_{ci} = \alpha\tilde{w}_{ci} + (1-\alpha)(1-w_{var}) \qquad i \in g, b. \tag{7.6}$$

7.3.2 Trustworthiness

We define a random variable T to describe whether a node is trustworthy ($T = 1$) or not ($T = 0$). Based on subjective logic and on the trust model described in Section 7.3.1, we can decide if a node is trustworthy by observing belief, disbelief and uncertainty. These

three values can be differently combined to infer trustworthiness, e.g., by considering in different ways the role of the uncertainty. For example, in the following we will use

$$\begin{aligned} T &= 1 \qquad \text{if } b + \beta u > d + (1-\beta)u \\ T &= 0 \qquad \text{otherwise,} \end{aligned} \tag{7.7}$$

with $\beta \in [0,1]$. Based on the HMM we can compute the probability that a node is considered trustworthy after N_t observations. We compute

$$P[T = 1 \mid N_t] = P[b + \beta u > d + (1-\beta)u \mid N_t]. \tag{7.8}$$

We define $\mathcal{M}$ as the set of all the m_g values for which $T = 1$, for a given number of misbehaviors in BAD channel and for a given number of visits to the BAD state (N_b)

$$\mathcal{M} = \{m_g : b + \beta u > d + (1-\beta)u\} \tag{7.9}$$

We remember that the number of visits to the GOOD state (N_g) can be obtained from the equation $N_t = N_b + N_g$, and that $N_s = m_s + c_s$ with $s \in \mathcal{S}$.

The probability in Equation (7.8) can be computed by conditioning on the number of misbehaviors in the BAD channel m_b and on the number of visits either to the GOOD state (N_g) or to the BAD state (N_b).

$$\begin{aligned} P[T = 1 \mid N_t] = \sum_{N_b=0}^{N_t} \sum_{m_b=0}^{N_b} \sum_{m_g \in \mathcal{M}} & P\Big[m_g \;\Big|\; N_t, m_b, N_b \Big] \\ & P\Big[m_b \mid N_b, N_t \Big] P\Big[N_b \mid N_t \Big]. \end{aligned} \tag{7.10}$$

where m_g and m_b follow a binomial distribution: $m_g \sim \text{Bin}(N_g, o_g(M))$ and $m_B \sim \text{Bin}(N_b, o_b(M))$, respectively. Therefore, the trust probability becomes

$$\begin{aligned} P[T = 1 \mid N_t] = \sum_{N_b=0}^{N_t} \sum_{m_b=0}^{N_b} \sum_{m_g \in \mathcal{M}} & \binom{N_t - N_b}{i} o_g(M)^i (1 - o_g(M))^{(N_t - N_b - i))} \\ & \binom{N_b}{m_b} o_b(M)^{m_b} (1 - o_b(M))^{(N_b - m_b)} P\Big[N_b \mid N_t \Big]. \end{aligned} \tag{7.11}$$

The last step is to compute the probability of visiting the BAD state N_b times, in a given number of steps N_t (or equivalently the number of visits in the GOOD state). We define $\phi_s(k,n) = P[k \text{ visits to B in } n \text{ steps} \mid X_0 = s]$ as the probability of visiting k times the BAD state in n steps, given that we start in the initial state $s \in \mathcal{S}$. We can recursively compute $\phi_s(k,n)$ exploiting the properties of a MC by conditioning on the first step

$$\begin{aligned} \phi_g(k,n) &= P_{gg}\phi_g(k, n-1) + P_{gb}\phi_b(k, n-1) \\ \phi_b(k,n) &= P_{bg}\phi_g(k-1, n-1) + P_{bb}\phi_b(k-1, n-1)\,, \end{aligned} \tag{7.12}$$

with the initial conditions $\phi_s(0,0) = 1$ and $\phi_s(k,n) = 0$ if $k > n$, with $s \in \mathcal{S}$. In the first row only the number of steps is decreased because, starting from the GOOD state, there is no visit to the BAD state in the first step. On the other hand, in the second row, given that we start from the BAD state both the number of remaining visits and the number of steps are decreased by one. Finally, $P[N_b \mid N_t]$ can be computed as

$$P[N_b \mid N_t] = \pi_g \phi_g(N_b, N_t) + \pi_b \phi_b(N_b, N_t) \tag{7.13}$$

7.3.3 Variable Weights

To compute the set $\mathcal{M}$ for which the node is considered trustworthy, we need to define w_{var}. If we consider a scenario with both GOOD and BAD channel states, we define w_{var} as a function of the estimated misbehavior probability in GOOD and BAD channel, $p_{m,g} = m_g/N_g$ and $p_{m,b} = m_b/N_b$, respectively. A GOOD channel should be characterized by a small number of misbehaviors, while in BAD channel misbehaviors are more likely to be observed due to the higher packet error rate. Comparing the misbehavior rate in GOOD and BAD channel, we can gain some insight about the behavior of the neighbor. Indeed, a number of misbehaviors in the GOOD channel comparable or even higher than the number of misbehaviors in the BAD channel could be the manifestation of an attacking node, and therefore the weights for the misbehavior should be increased, while the weight for the correct behavior should be decreased. We define

$$\begin{aligned} w_{var} &= \frac{2p_{m,g}}{p_{m,g} + p_{m,b}} && \text{if } p_{m,g} < p_{m,b} \\ w_{var} &= 1 && \text{otherwise.} \end{aligned} \tag{7.14}$$

Using this definition, w_{var} is equal to 1 when the misbehaviors in GOOD channel are much higher than the misbehaviors in the BAD channel, and close to 0 when there are only few misbehaviors in GOOD channel as would be expected by a node behaving normally. This definition needs to be slightly modified for those scenarios in which the channel quality is favorable, and thus the channel always remains in GOOD state (i.e., $N_b = 0$). In this case, since $p_{m,b}$ cannot be computed, we consider a target value equal to 0.5, therefore the definition becomes:

$$\begin{aligned} w_{var} &= \frac{2p_g}{p_g + 0.5} && \text{if } p_g < 0.5 \\ w_{var} &= 1 && \text{otherwise.} \end{aligned} \tag{7.15}$$

The last step is to find the set $\mathcal{M}$, i.e., the number of misbehaviors in GOOD channel for which the node can be trusted, by solving the inequality

$$b + \beta u > d + (1 - \beta)u\,. \tag{7.16}$$

Substituting the expression for belief and disbelief defined in Equation (7.4), remembering that $u = 1 - b - d$, and considering that $c_s = N_s - m_s$ with $s \in \mathcal{S}$, we obtain the

following inequality

$$\begin{aligned} m_g((1-\beta)w_{cg}+\beta w_{mg})+m_b((1-\beta)w_{cb}+\beta w_{mb}) < \\ (1-\beta)w_{cb}N_b+(1-\beta)w_{cg}N_g-N_t(1/2-\beta) \end{aligned} \tag{7.17}$$

In addition, we assume equal weights for the correct behaviors with GOOD and BAD channel, i.e., $\tilde{w}_{cb}=\tilde{w}_{cg}=\tilde{w}_c$ and therefore $w_{cb}=w_{cg}=w_c$. Considering that $N_t=N_g+N_b$, the inequality becomes

$$m_g((1-\beta)w_c+\beta w_{mg})+m_b((1-\beta)w_c+\beta w_{mb}) < N_t((1-\beta)w_c+\beta-1/2) \tag{7.18}$$

Substituting Equations (7.5) and (7.6) in the previous inequality, we obtain

$$a_g m_g + a_b m_b + k_1 w_{var}(m_g+m_b) + k_2 w_{var} - k_3 < 0 \tag{7.19}$$

with the coefficients defined as

$$\begin{aligned} a_g =& \alpha((1-\beta)\tilde{w}_c+\beta\tilde{w}_{mg})+(1-\alpha)(1-\beta) \\ a_b =& \alpha((1-\beta)\tilde{w}_c+\beta\tilde{w}_{mb})+(1-\alpha)(1-\beta) \\ k_1 =& (1-\alpha)(2\beta-1) \\ k_2 =& N_t(1-\alpha)(1-\beta) \\ k_3 =& N_t(\beta-1/2+\alpha(1-\beta)\tilde{w}_c+(1-\alpha)(1-\beta)) \end{aligned} \tag{7.20}$$

The last step is to substitute the value of w_{var} in the two cases as defined in Equation (7.14), $p_{m,g}<p_{m,b}$ and $p_{m,g}\geq p_{m,b}$, obtaining

$$c_1 m_g^2 + c_2 m_g + c_3 < 0 \tag{7.21}$$

with

$$\begin{aligned} c_1 =& \frac{a_g+2k_1}{N_g} \\ c_2 =& a_g p_{m,b} + \frac{2k_1 m_b + 2k_2 + a_b m_b - k_3}{N_g} \\ c_3 =& a_b m_b - k_3 \end{aligned} \tag{7.22}$$

with $p_{m,g}<p_{m,b}$, and

$$m_g < \frac{k_3-k_2-m_b(a_b+k_1)}{a_g+k_1} \tag{7.23}$$

with $p_{m,g}\geq p_{m,b}$. This inequality holds for the general case with both GOOD and BAD channel. If we consider the case with only GOOD channel, the solutions can be obtained by substituting $N_b=0$ (and therefore $m_b=0$) and $p_{m,b}=0.5$.

In addition to variable weights, in principle also different values of β could be used for different scenarios. As an example, where the channel remains always in the GOOD state ($N_b=0$) the role of uncertainty would be different since less uncertainty is expected from a node experiencing good channel condition. In such a scenario β could be set to a low value (or at least lower than in the general case with both GOOD and BAD channel condition), to help discover malicious nodes performing weaker attacks (e.g., not performing the intended task only occasionally).

7.3.4 Malicious Node

In the case of a malicious node the trust model remains the same, with the only exception of the probability of behaving correctly or maliciously. An attacker intentionally acts to damage the network, therefore the probability of misbehaving is not only related to the channel quality but also to the strength and type of attack the node is going to perform. If we consider a malicious node that performs an attack not accomplishing intentionally its task with a given probability p_d, e.g., does not forward the packet according to the protocol rules, the probability of behaving correctly becomes

$$\tilde{o}_s(C) = o_s(C)(1 - p_d) \qquad \forall s \in \mathcal{S} \tag{7.24}$$

and therefore, the probability of misbehaving is

$$\tilde{o}_s(M) = 1 - \tilde{o}_s(C) \qquad \forall s \in \mathcal{S} \tag{7.25}$$

By substituting $o_s(M)$ with $\tilde{o}_s(M)$ in Equation (7.11) we obtain the trust probability of an attacker acting maliciously with probability p_d.

7.4 Scenario Description and Parameter Settings

We analyze the trust model, both through an analytical formulation based on HMM and through simulation with the DESERT Underwater Network simulator [149].

The HMM described in Section 7.2 is characterized by the transition probabilities $p_{gg} = 0.87$ and $p_{bb} = 0.72$ when both GOOD and BAD channel are considered. For the theoretical analysis we compare the results with different probabilities of observing a misbehavior ($o_s(M)$) or a correct behavior ($o_s(C)$) for each channel state $s \in \mathcal{S}$, mimicking different channel qualities for a normal node acting according to the protocol rules. We assess the trust probability of both a normal node and a malicious node performing attacks of different strength, i.e., with different values of p_d. In the scenario with both GOOD and BAD channel states we assume a value of $\beta = 0.7$, meaning that uncertainty is mostly considered as part of the trustworthiness of the node. In the scenario with only GOOD channel state we consider $\beta = 0$, therefore the uncertainty computed with the subjective logic is considered as the sign of an untrustworthy node. Indeed, in this second case, the channel condition is favorable and in principle the possibility of observing a misbehavior is lower, therefore each misbehavior needs to be carefully taken into account in the trust model. A value of $\beta = 0$ allows us to better detect attackers in such a scenario.

As a second step we test the trust model in an underwater network. We assess our model in topologies similar to the one depicted in Figure 7.2. Specifically, the network is composed by 10 nodes, 9 of them generating data and sending it to the sink placed in the center of the network using flooding as the routing protocol. Each node receiving a packet and running the flooding protocol is expected to forward the data to all its neighbors, until reaching the sink node. When a node overhears the packet forwarded by the neighbor a correct behavior is considered, while if the forwarding is

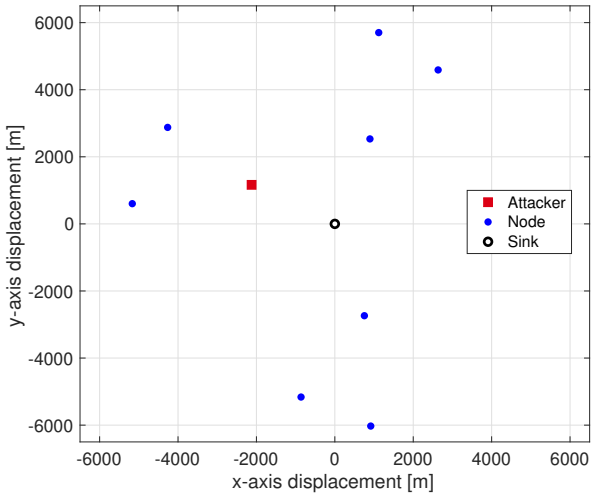
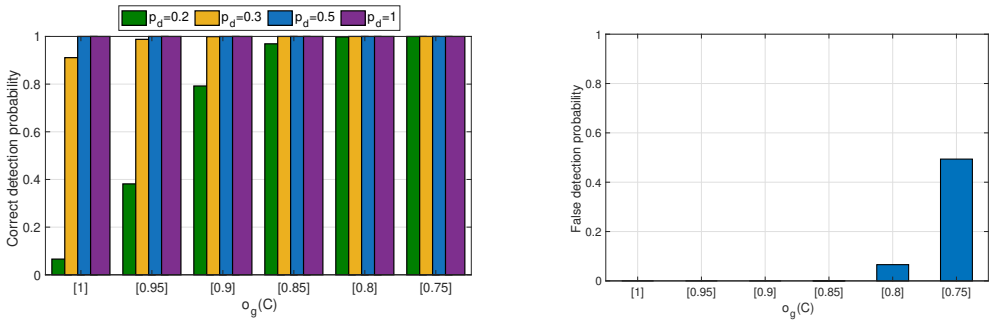


Figure 7.2: Example of topology with normal nodes (blue circles), attacker (red square) and sink node (black circle).

not overheard within a predefined time interval a misbehavior is counted. We remark that the observed misbehaviors can in principle be either intentionally caused by an attacker, or unintended due to the channel condition. The count of correct behaviors and misbehaviors will be used to compute the trust of a node considering Equations (7.4) and (7.7). To test our trust model we select one of the nodes close to the sink to act as the attacker from the beginning of the simulation. Specifically, the attacker does not always forward the packets received from its neighbor, but drops them with a given probability p_d. In this scenario we consider that all the nodes have the same trust level at the beginning of the simulation. In the network simulator the channel state is obtained by looking at the SINR of each received packet or at the noise level when the node does not overhear the forwarding of a packet by one of its neighbors. We use two thresholds to detect the channel state, one for the transition from GOOD state to BAD state $S_{th,g}$, and the second one for the transition from BAD state to GOOD state $S_{th,b}$. The hysteresis is useful to avoid continuous jumps from one state to the other with small changes in the SINR value. In our scenario we set $S_{th,g} = 6.3$ dB and $S_{th,b} = 7$ dB.

To simulate the behavior of a two-state channel model as that described in Section 7.2, in the simulator we use the Urick propagation model [224], but changing the noise level every $T = 180$ s between two values, according to the transition probability of a MC. For consistency, we set the same transition probabilities used in the theoretical analysis.

In the simulated scenario, each node generates a packet of 24 Bytes every 100 s, on average. The transmission power is equal to $P_{tx} = 180$ dB re μPa, the central frequency used for the transmission is $f_0 = 26$ kHz and the bandwidth is $B = 16$ kHz.



(a) **(b)**

Figure 7.3: Correct detection (a) and false detection (b) probabilities as a function of the correct behavior probability for GOOD channel scenario.

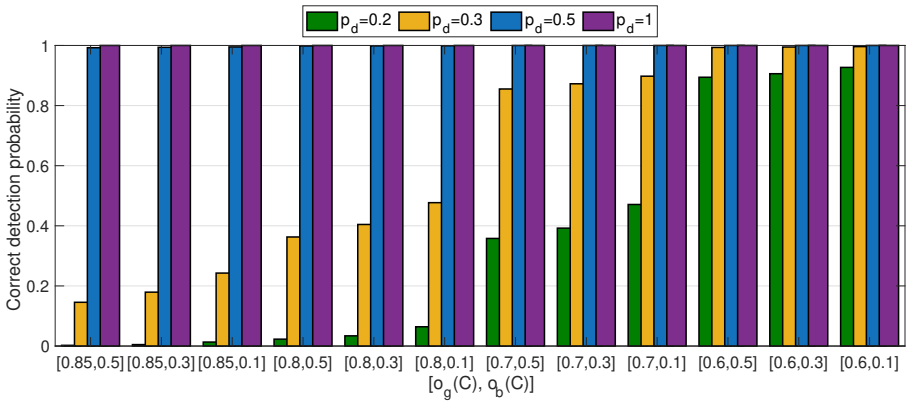
7.5 Results

7.5.1 Analytical Results

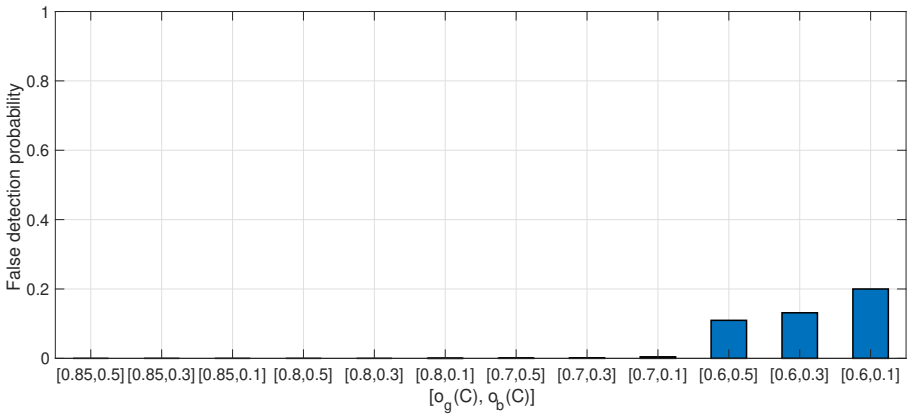
In this Section we present the results obtained from the theoretical trust model described in Section 7.3. We compute the probability of correct detection for a malicious node, i.e., the probability of not trusting the attacker (a node with $p_d > 0$), and the probability of false detection, i.e., the probability of not trusting a correctly behaving node (with $p_d = 0$), after $N_t = 150$ steps. We computed the correct detection and the false detection probabilities in two different scenarios: the first one in which the channel always remains in GOOD state, with the goal of analyzing the trustworthiness of a node under very favorable channel conditions, the second one with a more general behavior where both GOOD and BAD channel states are considered.

Figures 7.3a and 7.3b show the correct detection and the false detection probabilities in the scenario with only GOOD channel. The analysis has been carried out as a function of the probability of observing a correct behavior $o_g(C)$ (that is related to the packet delivery ratio) in a GOOD channel state and considering different attack strengths (i.e., different probabilities p_d of intentional misbehavior). For each analyzed channel quality, an attacker behaving intentionally maliciously with a probability of performing an attack of $p_d \geq 0.3$ can be easily identified and marked as an untrustworthy node. However, when the correct behavior probability $o_g(C)$ drops to 0.75, the false detection probability rapidly increases to 0.5, meaning that even the well behaving nodes are marked as untrustworthy half of the time. This is not true if the channel quality is higher, as is expected for this particular scenario with only GOOD channel condition. Indeed, for better channel conditions the false detection probability remains lower than 0.1 and even close to 0 for a correct behavior probability lower than or equal to 0.85.

Figures 7.4a and 7.4b show the correct detection and false detection probabilities in the scenario with both GOOD and BAD channel. The analysis has been performed



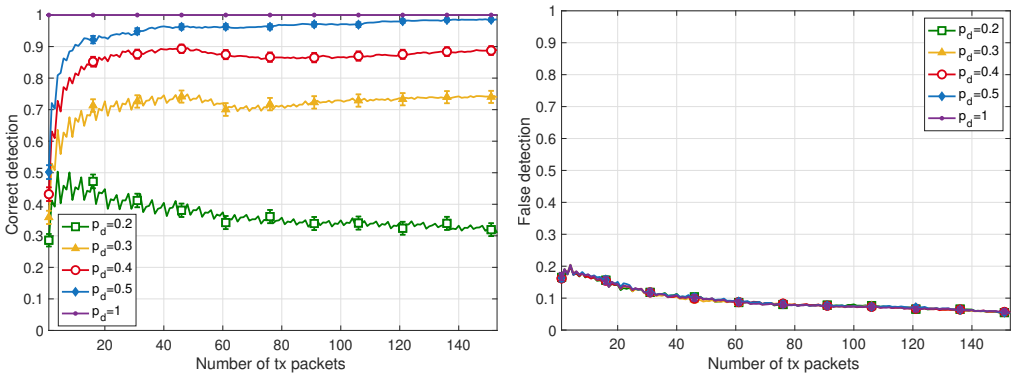
(a)



(b)

Figure 7.4: Correct detection (a) and false detection (b) probabilities as a function of the correct behavior probability for GOOD and BAD channel scenario

as a function of the correct behavior probabilities in both GOOD and BAD channel $[o_g(C), o_b(C)]$, and considering different attack strengths p_d. In this scenario, since the channel quality is lower than in the previous scenario with only GOOD channel, the detection of attackers with a low p_d becomes more difficult. Indeed, when $p_d \geq 0.5$ the attacker is always detected after 150 steps, while for $p_d \leq 0.3$ the correct detection depends on the channel quality. An increasing error probability due to channel losses seems to help in detecting attackers with lower strength. This is also followed by an increment of the false detection probability (7.4b), however the increment in the false detection probability in our scenario is limited when $o_g(C) = 0.6$ and always remains lower than 0.2.



(a) (b)

Figure 7.5: Correct detection (a) and false detection (b) probabilities as a function of the number of transmitted packets and for different attack strength p_d.

7.5.2 Simulation Results

In this Section we present the results obtained through simulation of the scenario described in Section 7.4. We assessed the trustworthiness of the nodes, considering 50 runs for each of the 20 analyzed topologies, similar to that presented in Figure 7.2. Specifically, we considered the trust computed by each node in the most external set with respect to its neighbor closer to the sink. Figures 7.5a and 7.5b show the correct detection (for an attacker) and false detection (for a normal node) probabilities as a function of the number of transmitted packets and for different drop probabilities p_d, taking into account the results obtained for each run and each topology. Figure 7.5a shows the same trend observed with the theoretical results. We want to highlight that, depending on the topologies and thus on the actual distance between neighbors, the performance takes into account both nodes with favorable conditions (i.e., with only GOOD channel) and nodes with more unfavorable conditions (i.e., nodes that alternate GOOD and BAD channel states). For a drop probability $p_d = 0.2$, the system cannot easily identify an attacker because the drops caused intentionally by the malicious node can be confused with the losses caused by bad channel conditions or collisions with other transmissions. With $p_d \geq 0.3$ the overall performance improves, going from a correct detection probability of 0.7 with $p_d = 0.3$, to a detection of almost 100% with $p_d \geq 0.5$. On the other hand, Figure 7.5b shows the estimated false detection probability for a normal node as a function of the number of transmitted packets. As expected, the result does not depend on the the drop probability p_d of the attacker. The false detection probability is close to 0.2 at the very beginning of the simulations, when few packets have been exchanged, while it decreases to 0.05 when more information becomes available.

As mentioned before, this analysis considers all topologies, therefore it takes into account different channel quality for both attackers and normal nodes. For a better understanding of the behavior of the trust model for different simulated channel qualities,

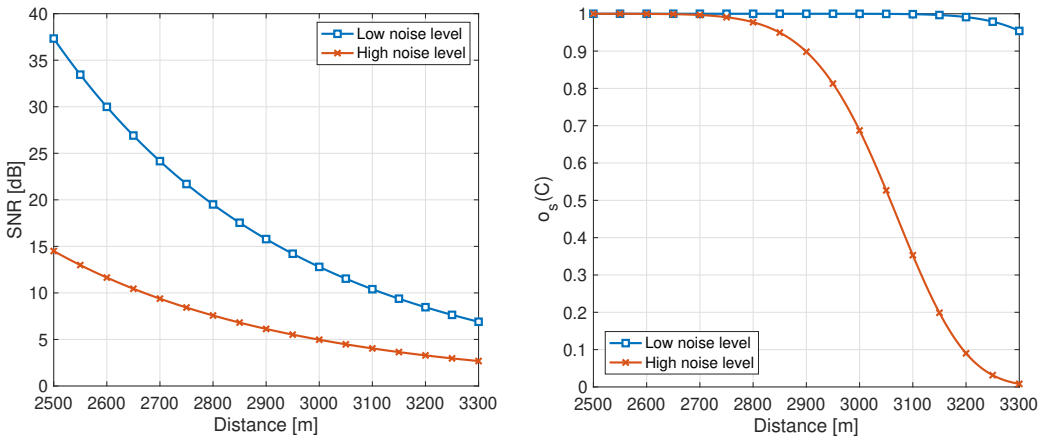


Figure 7.6: SNR (a) and probability of observing a correct behavior (b) as a function of the distances for low (blue) and high (red) noise level.

we plot the estimated correct and false detection probabilities at the end of the simulation as a function of the distance between the node and its neighbors. Figures 7.6a and 7.6b are showing the SNR experienced by a node as a function of the distance for low and high noise level and the probability of observing a correct behavior from the neighbor based on the channel, respectively. Since different distances correspond to different channel qualities, this allows us to understand the behavior of the trust model for different channels. Figures 7.7a and 7.7b depict the estimated correct and false detection probabilities, respectively. According to the propagation model used in the simulator, a node placed at a distance lower than 3 km is always in GOOD channel, since the SNR is higher than the threshold $S_{th,g}$. In this scenario the trend obtained for the correct detection is similar to the trend observed with the analytical results with always GOOD channel condition (Figure 7.3a). An attacker with $p_d \geq 0.3$ is correctly detected with a probability close to 1. Considering $p_d = 0.2$, the correct detection probability is very low for closer nodes, but increases as the distance increases because the packet losses due to channel errors help in the attacker identification. At the same time, for a distance lower than 3 km the false detection probability is close to 0, meaning that the normal nodes are not wrongly marked as attackers. Nodes within a range of 3 to 3.1 km can experience both situations, i.e., either always GOOD channel condition or both GOOD and BAD channel. In this case attackers dropping packets with $p_d \geq 0.5$ are always correctly detected, while for an attack strength of $p_d = 0.2$ and $p_d = 0.3$ the correct detection probability is 0.7. In this range of distances the false detection probabilities increase up to 0.2, due to those nodes that are always in GOOD channel condition but with an increased packet error rate. When the distance is bigger than 3.1 km the nodes always experience both GOOD and BAD channel conditions. Also in this case the trend for correct detection and false detection is similar to what observed

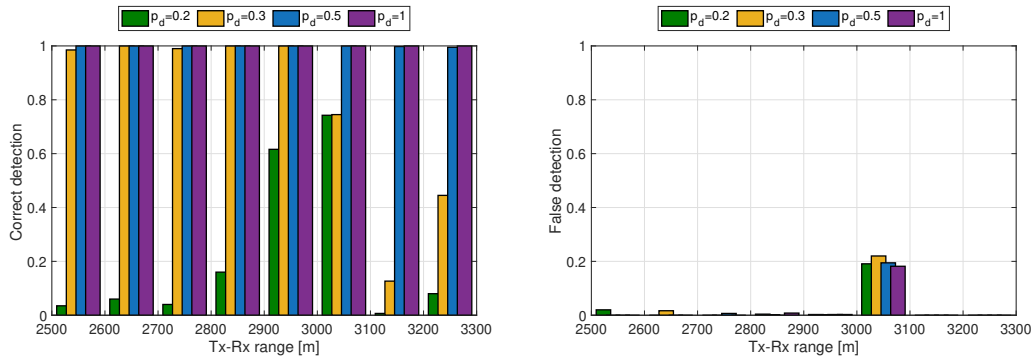


Figure 7.7: Correct detection (a) and false detection (b) probabilities as a function of the number of transmitted packets and for different attack strength p_d.

in the theoretical analysis. For the considered distances, a lower attack strength is more difficult to detect, therefore the correct detection probability with $p_d \leq 0.3$ is lower than 0.5, while it remains close to 1 for $p_d \geq 0.5$. In this situation the estimated false detection probability is close to 0.

7.6 Conclusions

In this chapter we presented a trust model for underwater acoustic networks to detect suspicious behaviors of possible attackers. The main problem in acoustic communication is to understand whether a misbehavior is due to channel loss conditions or to a malicious behavior. The dynamic quality of an acoustic channel can be described through a two-state HMM and we exploited this characteristic to weigh differently misbehaviors in GOOD and BAD channel conditions. We analyzed the trust model both analytically and through simulations. Specifically, we computed the correct detection and false detection probabilities for different attack strengths p_d, observing that when $p_d \geq 0.5$ the malicious node is always detected, while for lower values of p_d in scenarios with both GOOD and BAD channel states the detection is more challenging since the intentional misbehavior is difficult to distinguish from a misbehavior caused by a channel drop. If the channel quality is more favorable, i.e., with only GOOD channel, even a value of $p_d = 0.3$ can be detected. The simulations of the trust model with the flooding routing protocols confirm the same trend observed with the analytical results.

As future work, we will extend the trust model by letting the nodes exchange information about their trust level of a node, thus combining local information with the received information and obtain a more accurate result. In addition, we will design and evaluate countermeasures to exclude the attacker from the network which will exploit the trust model as a base for the detection of malicious nodes.

7. TRUST MODEL FOR SECURITY IN UNDERWATER ACOUSTIC NETWORKS

Chapter 8

Conclusions

The development of underwater networks still faces many challenges, from the design of effective protocols to counteract the channel impairments, to a deep analysis of possible attacks and countermeasures in network security. In the first part of the thesis we reviewed the possible technologies that can be employed for underwater communications, specifying pros and cons of each of them and the most suitable scenarios in which they can be employed. We then reviewed the state of the art of network security with a focus on underwater acoustic networks, discussing the limits of applying to underwater networks the countermeasures designed for their terrestrial counterpart.

We then investigated possible applications for underwater networks in the future smart port scenario. We focused on the E2E data collection service to gather data, by means of an underwater vehicle, from underwater sensors deployed in the harbor area to monitor the underwater environment. We designed a polling-based MAC protocol to collect data from the sensors and then to forward it to buoys. We assessed how the parameters of the protocol, such as the maximum backoff time, affect the network performance as a function of the network node density. We also tested the UW-POLLING protocol in an experiment at lake Kreidesee in Hemmoor (Germany). Using the DESERT Framework we were able to test the same protocol stack used in the simulation analysis also in the lake test, where we employed the AHOI modems to transmit the actual data. In the data collection service, the data obtained from underwater sensors needs to be forwarded to the shore through an above water network. In this thesis we finally investigated the feasibility of using LoRaWAN for the above water data transmission. Since the LoRa nodes are constrained by duty-cycle restrictions, we analyzed the performance as a function of the number of buoys equipped with a LoRa device. We also analyzed whether the bottleneck of the E2E performance is the underwater network or the LoRaWAN, showing that most of the time LoRaWAN is an enabling technology for the data collection service.

In the second part of the thesis we studied attacks and countermeasures specifically tailored for underwater acoustic networks. We started from simple attacks, such as jamming and replay attacks, for which no specific knowledge about the network protocol

stack is needed. We analyzed jamming attacks through a game-theoretical framework, taking into account the propagation characteristics of acoustic waves. First, we studied the effectiveness of a blind jammer, which randomly chooses the slots to jam, obtaining the optimal strategies to use for the transmissions of both the sender and the jammer. We also evaluated, through a bayesian game, whether the performance of a transmitter with no information about the jammer position gets worse. We discovered that, observing the outcome of its transmissions, the transmitter can quickly identify the jammer distance and start to play like it had complete information. We then compared reactive and blind jammer for different geometries of the network deployment. Indeed, varying the relative positions between the transmitter, the jammer, and the receiver, the portion of packet a reactive jammer can hit changes and so its effectiveness in blocking the communication. We studied whether it is more convenient for a transmitter to use less robust modulations, with a lower packet duration and energy consumption, but being less vulnerable to reactive jammer, or conversely it is better to increase the robustness of its transmissions, increasing the energy consumption and the packet duration, but being more prone to reactive jammer. We also observed that after a given threshold on the angle, blind jamming becomes a more effective solution than the reactive one. In the thesis, we also analyzed the effect of replay attack in underwater acoustic networks and some possible countermeasures based on either a timestamp or a HASH mechanism. We observed that, in acoustic networks, which are often characterized by large packet delivery delay, the mechanism based on timestamp is not the best solution against replay attack and it is always outperformed by the HASH mechanism as long as the queue to store the HASH values is long enough. We then moved a step forward, by increasing the knowledge and the capabilities of the attackers. We studied the effect of a simple countermeasure based on a reputation system against sinkhole and resource exhaustion attacks. Finally, we designed a trust model specifically tailored for underwater acoustic networks. The model makes use of subjective logic to consider the uncertainty due to the acoustic channel variability into the trustworthiness computation. The model has been validated both analytically, modeling the acoustic channel thorough an HMM, and through simulations, studying its effectiveness in discovering attackers which perform a drop attack in a scenario where flooding protocol is employed in the routing phase. We observed that most of the time our trust model is able to correctly identify the attackers without erroneously detecting as attackers normal nodes, therefore being able to discern whether the observed misbehaviors are due to the channel or intentionally caused by malicious node. Still the trust model could be further studied and investigated to design and evaluate countermeasures to exclude the attacker from the network which will exploit the trust model as a base for the detection of malicious nodes.

Bibliography

[1] Porter M. *et al.*, "Bellhop code," http://oalib.hlsresearch.com/Rays/index.html, Last time accessed: Sep. 2021.

[2] K. Pelekanakis and L. Cazzanti, "On adaptive modulation for low SNR underwater acoustic communications," in *Proc. MTS/IEEE OCEANS*, Charleston, SC, Oct. 2018.

[3] S. I. Inácio, M. R. Pereira, H. M. Santos, L. M. Pessoa, F. B. Teixeira, M. J. Lopes, O. Aboderin, and H. Salgado, "Dipole antenna for underwater radio communications," in *Proc. IEEE UComms*, Lerici, Italy, August 2016.

[4] "Seatooth RF modem," https://www.csignum.com/product/seatooth/, Last time accessed: Sep. 2021.

[5] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, Oct. 2007.

[6] "EvoLogics S2C Acoustic Modems," https://evologics.de/acoustic-modems, Last time accessed: Sep. 2021.

[7] A. Signori, F. Campagnaro, and M. Zorzi, "Modeling the performance of optical modems in the desert underwater network simulator," in *Proc. IEEE UComms*, Lerici, Italy, August 2018.

[8] "Sonardyne bluecomm optical modem," https://www.sonardyne.com/product/bluecomm-200-wireless-underwater-video/, Last time accessed: Sep. 2021.

[9] H. Dol, "EDA-SALSA: Towards smart adaptive underwater acoustic networking," in *Proc. MTS/IEEE OCEANS*, Marseille, France, 2019.

[10] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: applications, advances and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 158–175, Jan. 2012.

[11] A. Signori, F. Campagnaro, F. Steinmetz, B.-C. Renner, and M. Zorzi, "Data gathering from a multimodal dense underwater acoustic sensor network deployed in shallow fresh water scenarios," *MDPI Journal of Sensor and Actuator Networks*, vol. 8, no. 4, p. 55, Dec. 2019.

[12] "Robotic Vessels as-a-Service," https://www.martera.eu/projects/robovaas, Last time accessed: Sep. 2021.

[13] F. Campagnaro, R. Francescon, P. Casari, R. Diamant, and M. Zorzi, "Multimodal underwater networks: Recent advances and a look ahead," in *Proc. ACM WUWNet*, Halifax, Canada, 2017.

[14] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 103–116, Spring 2009.

[15] E. Coccolo, F. Campagnaro, A. Signori, F. Favaro, and M. Zorzi, "Implementation of AUV and ship noise for link quality evaluation in the DESERT Underwater framework," in *Proc. ACM WUWNet*, Shenzhen, China, Dec. 2018.

[16] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proc. OCEANS - Asia Pacific*, Singapore, May 2006, pp. 1–7.

[17] F. Guerra, P. Casari, and M. Zorzi, "A performance comparison of MAC protocols for underwater networks using a realistic channel simulator," in *Proc. OCEANS 2009*, Biloxi, MS, USA, Oct. 2009, pp. 1–8.

[18] G. Toso, R. Masiero, P. Casari, O. Kebkal, M. Komar, and M. Zorzi, "Field experiments for dynamic source routing: S2C evologics modems run the SUN protocol using the DESERT Underwater libraries," in *Proc. MTS/IEEE OCEANS*, Hampton Roads, VA, Oct. 2012.

[19] M. Goetz and I. Nissen, "GUWMANET — multicast routing in underwater acoustic networks," in *Proc. Military Communications and Information Systems Conference (MCC)*, Gdansk, Poland, Oct 2012, pp. 1–8.

[20] R. Diamant, P. Casari, F. Campagnaro, O. Kebkal, V. Kebkal, and M. Zorzi, "Fair and throughput-optimal routing in multimodal underwater networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1738–1754, Mar. 2018.

[21] P. van Walree, H. Buen, and R. Otnes, "A performance comparison between DSSS, M-FSK, and frequency-division multiplexing in underwater acoustic channels," in *Proc. IEEE UComms*, Sestri Levante, Italy, Sep. 2014, pp. 1–5.

[22] O. Kebkal, M. Komar, and K. Kebkal, "D-MAC: Hybrid media access control for underwater acoustic sensor networks," in *Proc. IEEE ICC Workshops*, Cape Town, South Africa, 2010.

[23] C. Lal, R. Petroccia, K. Pelekanakis, M. Conti, and J. Alves, "Towards the development of secure underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 1075–1087, October 2007.

[24] G. Yang, L. Dai, and Z. Wei, "Challenges, threats, security issues and new trends of underwater wireless sensor networks," *MDPI Sensors*, vol. 18, no. 11, p. 3907, Nov. 2018.

[25] F. Campagnaro, A. Signori, and M. Zorzi, "Wireless remote control for underwater vehicles," *MDPI Journal of Marine Science and Engineering*, vol. 8, no. 10, p. 736, Oct. 2020.

[26] N. Miller, "An underwater communication system," *IEEE IRE Transactions on Communications Systems*, vol. 7, no. 4, pp. 249–251, Dec. 1959.

[27] "Teledyne-benthos acoustic modems," http://www.teledynemarine.com/acoustic-modems/, last time accessed: Sep. 2021.

[28] "LinkQuest Underwater Acoustic Modems," http://www.link-quest.com/html/models1.htm, last time accessed: Sep. 2021.

[29] "Communication Design," http://www.aquatecgroup.com/19-solutions/107-communication-design, Last time accessed: Sep. 2021.

[30] "Develogic Subsea Systems," http://www.develogic.de/, Last time accessed: Sep. 2021.

[31] "MATS 3G multi-modulation acoustic telemetry system," http://www.sercel.com/products/Pages/mats3g.aspx, last time accessed: Sep. 2021.

[32] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. MTS/IEEE OCEANS*, Washington, DC, USA, Sep. 205.

[33] "Modems for underwater communication," https://www.kongsberg.com/maritime/products/Acoustics-Positioning-and-Communication/modems/, last time accessed: Sep. 2021.

[34] L. Freitag, K. Ball, J. Partan, P. Koski, and S. Singh, "Long range acoustic communications and navigation in the Arctic," in *Proc. MTS/IEEE OCEANS*, Washington, DC, USA, Oct. 2015.

[35] "Popoto modem," http://popotomodem.com/, Last time accessed: Sep. 2021.

[36] G. Cario, A. Casavola, M. Lupia, and C. Rosace, "SeaModem: A Low-Cost Underwater Acoustic Modem for Shallow Water Communication," in *Proc. MTS/IEEE OCEANS*, Genova, Italy, May 2015.

[37] Sonardyne, "Underwater acoustic modem 6," https://www.sonardyne.com/product/underwater-acoustic-modems/, last time accessed: Sep. 2021.

[38] "Low cost underwater acoustic modem for makers of underwater things and oems!" https://dspcommgen2.com/news-flash-low-cost-acoustic-modems-and-transducers-available-for-sale-now/, last time accessed: Sep. 2021.

[39] "Subnero," https://subnero.com/, last time accessed: Sep. 2021.

[40] "SeaTrac Technology," https://www.blueprintsubsea.com/seatrac/technology.php, Last time accessed: Sep. 2021.

[41] T. Shimura, Y. Kida, and M. Deguchi, "High-rate acoustic communication at the data rate of 69 kbps over the range of 3,600 m developed for vertical uplink communication," in *Proc. MTS/IEEE OCEANS*, Marseille, France, Jun. 2019.

[42] B. Binnerts, I. Mulders, K. Blom, M. Colin, and H. Dol, "Development and demonstration of a live data streaming capability using an underwater acoustic communication link," in *Proc. MTS/IEEE OCEANS*, Kobe, Japan, May 2018.

[43] R. Otnes, J. Locke, A. Komulainen, S. Blouin, D. Clark, H. Austad, and J. Eastwood, "Dflood network protocol over commercial modems," in *Proc. IEEE UComms*, Lerici, Italy, Aug. 2018.

[44] H. Dol, M. Colin, P. van Walree, and R. Otnes, "Field experiments with a dual-frequency-band underwater acoustic network," in *Proc. IEEE UComms*, Lerici, Italy, Aug. 2018.

[45] "Underwater communication systems," https://www.wartsila.com/marine/build/sonars-naval-acoustics/underwater-communication-systems, Last time accessed: Sep. 2021.

[46] "L3HARRIS," https://www.l3harris.com/all-capabilities/acoustic-general-purpose-modem, last time accessed: Sep. 2021.

[47] C. Tapparello, P. Casari, G. Toso, I. Calabrese, R. Otnes, P. van Walree, M. Goetz, I. Nissen, and M. Zorzi, "Performance evaluation of forwarding protocols for the RACUN network," in *Proc. ACM WUWNet*, Kaohsiung, Taiwan, Nov. 2013.

[48] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The janus underwater communications standard," in *Proc. IEEE UComms*, Sestri Levante, Italy, 2014.

[49] B. Sherlock, J. A. Neasham, and C. C. Tsimenidis, "Implementation of a spread-spectrum acoustic modem on an android mobile device," in *Proc. MTS/IEEE OCEANS*, Aberdeen, UK, Jun. 2017.

[50] N. Morozs, P. D. Mitchell, Y. Zakharov, R. Mourya, Y. R. Petillot, T. Gibney, M. Dragone, B. Sherlock, J. A. Neasham, C. C. Tsimenidis, M. E. Sayed, A. C. McConnell, S. Aracri, and A. A. Stokes, "Robust TDA-MAC for practical underwater sensor network deployment: Lessons from USMART sea trials," in *Proc. ACM WUWNet*, Shenzhen, China, Dec. 2018.

[51] B. Benson, Y. Li, B. Faunce, K. Domond, D. Kimball, C. Schurgers, and R. Kastner, "Design of a Low-Cost Underwater Acoustic Modem," *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 58–61, May 2010.

[52] "Tritech micron modem - acoustic modem," https://www.tritech.co.uk/product/micron-data-modem, last time accessed: Sep. 2021.

[53] Desert Star Systems, "SAM-1 Technical Reference Manual," https://desertstarsystems.nyc3.digitaloceanspaces.com/Manuals/SAM-1TechnicalReferenceManual.pdf, last Time Accessed: Sep. 2021.

[54] "DiveNET: Sealink," https://www.divenetgps.com/sealink, last time accessed: Sep. 2021.

[55] B.-C. Renner, J. Heitmann, and F. Steinmetz, "AHOI: Inexpensive, low-power communication and localization for underwater sensor networks and AUVs," *ACM Transactions on Sensor Networks*, vol. 16, no. 2, Jan. 2020.

[56] "Water Linked Modem M64," https://waterlinked.com/product/modem-m64/, Last time accessed: Sep. 2021.

[57] A. Sanchez, S. Blanc, P. Yuste, A. Perles, and J. J. Serrano, "An Ultra-Low Power and Flexible Acoustic Modem Design to Develop Energy-Efficient Underwater Sensor Networks," *Sensors, Special Issue on Underwater Sensor Nodes and Underwater Sensor Networks*, vol. 12, no. 6, pp. 6837–6856, Jun. 2012.

[58] C. Pelekanakis, M. Stojanovic, and L. Freitag, "High rate acoustic link for underwater video transmission," in *Proc. MTS/IEEE OCEANS*, San Diego, CA, USA, Oct 2003.

[59] T. Riedl and A. Singer, "Towards a video-capable wireless underwater modem: Doppler tolerant broadband acoustic communication," in *Proc. IEEE UComms*, Sestri Levante, Italy, Sep. 2014.

[60] F. Campagnaro, R. Francescon, D. Tronchin, and M. Zorzi, "On the feasibility of video streaming through underwater acoustic links," in *Proc. IEEE UComms*, Lerici, Italy, Aug. 2018.

[61] M. Rahmati, A. Gurney, and D. Pompili, "Adaptive underwater video transmission via software-defined MIMO acoustic modems," in *Proc. MTS/IEEE OCEANS*, Charleston, US, May 2018.

[62] P. P. Beaujean, J. Spruance, E. A. Carlson, and D. Kriel, "HERMES - A high-speed acoustic modem for real-time transmission of uncompressed image and status transmission in port environment and very shallow water," in *Proc. MTS/IEEE OCEANS*, Québec City, Canada, Sep. 2008.

[63] E. Demirors, B. G. Shankar, G. E. Santagati, and T. Melodia, "SEANet: A software-defined acoustic networking framework for reconfigurable underwater networking," in *Proc. ACM WUWNet*, Washington DC, US, Oct. 2015.

[64] "Underwater Wireless Acoustic Video Communications Channel," http://www.baltrobotics.com/index.php/products-services-mnu/item/269-underwater-wireless-acoustic-video-communications-channel, last time accessed: Sep. 2021.

[65] "Robust and Reliable, Broadband Underwater Acoustic Communications," http://marecomms.ca/index.html, last time accessed: Sep. 2021.

[66] S. Singh, S. E. Webster, L. Freitag, L. L. Whitcomb, K. Ball, J. Bailey, and C. Taylor, "Acoustic communication performance of the whoi micro-modem in sea trials of the nereus vehicle to 11,000 m depth," in *Proceedings IEEE OCEANS*, Biloxi, US, Oct. 2009.

[67] D. Anguita, D. Brizzolara, G. Parodi, and Q. Hu, "Optical wireless underwater communication for auv: Preliminary simulation and experimental results," in *IEEE OCEANS*, Santander, Spain, Jun. 2011.

[68] F. Hanson and S. Radic, "High bandwidth underwater optical communication," *Applied Optics*, vol. 47, no. 2, pp. 277–283, Jan. 2008.

[69] "LUMA," https://www.hydromea.com/underwater-wireless-communication/, last time accessed: Sep. 2021.

[70] M. Doniec, A. Xu, and D. Rus, "Robust real-time underwater digital video streaming using optical communication," in *Proc. ICRA*, Karlsruhe, Germany, May 2013.

[71] M.-A. Khalighi, T. Hamza, S. Bourennane, P. Léon, and J. Opderbecke, "Underwater Wireless Optical Communications Using Silicon Photo-Multipliers," *IEEE Photonics Journal*, vol. 9, no. 4, pp. 1–14, Aug 2017.

[72] A. Caiti, E. Ciaramella, G. Conte, G. Cossu, D. Costa, S. Grechi, R. Nuti, D. Scaradozzi, and A. Sturniolo, "Optocomm: introducing a new optical underwater wireless communication modem," in *Proc. IEEE UComms*, Lerici, Italy, Sep. 2016.

[73] "Release of MC100 underwater optical wireless communication modem," https://www.shimadzu.com/news/g16mjizzgbhz3--y.html, last time accessed: Sep. 2021.

[74] "Sonardyne Bluecomm Underwater Wireless Optical Communication System," https://www.sonardyne.com/app/uploads/2016/06/BlueComm.pdf, Last time accessed: Sep. 2021.

[75] X. Liu, S. Yi, X. Zhou, Z. Fang, Z.-J. Qiu, L. Hu, C. Cong, L. Zheng, R. Liu, and P. Tian, "34.5 m underwater optical wireless communication with 2.70 gbps data rate based on a green laser diode with NRZ-OOK modulation," *Optics Express*, vol. 25, no. 22, pp. 27 937–27 947, Oct. 2017.

[76] J. Wang, C. Lu, S. Li, and Z. Xu, "100 m/500 mbps underwater optical wireless communication using an NRZ-OOK modulated 520 nm laser diode," *Optics Express*, vol. 27, no. 9, pp. 12 171–12 181, Apr. 2019.

[77] "Ultra subsea high-bandwith comms," http://www.oceanit.com/products/ultra, last time accessed: Sep. 2021.

[78] "Study of adaptive underwater optical wireless communication with photomultiplier tube suruga bay," http://www.godac.jamstec.go.jp/catalog/data/doc_catalog/media/KR17-11_leg2_all.pdf, last time accessed: Sep. 2021.

[79] "AQUAmodem Op1 Optical Modem page," http://www.aquatecgroup.com/aquamodem/aquamodem-op1, Last time accessed: Sep. 2021.

[80] F. Campagnaro, M. Calore, P. Casari, V. S. Calzado, G. Cupertino, C. Moriconi, and M. Zorzi, "Measurement-based Simulation of Underwater Optical Networks," in *Proc. MTS/IEEE OCEANS*, Aberdeen, UK, Oct. 2017.

[81] "About Penguin Automated Systems," http://www.penguinasi.com/, last time accessed: Sep. 2021.

[82] G. Baiden and Y. Bissiri, "High bandwidth optical networking for underwater untethered telerobotic operation," in *Proc. MTS/IEEE OCEANS*, Vancouver, Canada, Sep. 2007.

[83] G. Ardelt, M. Mackenberg, J. Markmann, T. Esemann, and H. Hellbruck, "A flexible and modular platform for development of short-range underwater communication," in *Proc. ACM WUWNet*, Shangai, China, Oct. 2016.

[84] P. Gois, N. Sreekantaswamy, N. Basavaraju, M. Rufino, L. S. J. Botelho, J. Gomes, and A. Pascoal, "Development and validation of Blue Ray, an optical modem for the MEDUSA class AUVs," in *Proc. IEEE UComms*, Lerici, Italy, Sep. 2016.

[85] "Optical Communications," https://www.saphotonics.com/communications-sensing/optical-communications/, last time accessed: Sep. 2021.

[86] Z. Zeng, S. Fu, H. Zhang, Y. Dong, and J. Cheng, "A Survey of Underwater Optical Wireless Communications," *IEEE Communications Survey and Tutorials*, vol. 19, no. 1, pp. 204–238, 2017.

[87] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks," *IEEE Communications Magazine*, vol. 48, no. 12, pp. 143–151, Dec. 2010.

[88] M. I. Sani, S. Siregar, A. P. Kurniawan, and M. A. Irwan, "FIToplankton: Wireless Controlled Remotely-operated Underwater Vehicle (ROV) for Shallow Water Exploration," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3325–3332, Oct. 2018.

[89] "IMM (Inductive Modem Module)," https://www.seabird.com/communications/imm-inductive-modem-module/family?productCategoryId=54627870427, last time accessed: Sep. 2021.

[90] "RBR Inductive Modem," https://rbr-global.com/products/systems/inductive-modem, last time accessed: Sep. 2021.

[91] "Ulti-modem Underwater Inductive Modem," https://www.soundnine.com/inductive-modems/, last time accessed: Sep. 2021.

[92] R. Diamant, F. Campagnaro, S. Dahan, R. Francescon, and M. Zorzi, "Development of a submerged hub for monitoring the deep sea," in *Proc. UACE2017*, Skiathos, Greece, July 2017.

[93] "Inductive Modems on Ice," https://www.nortekgroup.com/assets/documents/Nortek-Aquadopp-and-Ross-Ice-Shelf-research-Eco-Magazine-March-2019.pdf, last time accessed: Sep. 2021.

[94] "WFS Seatooth Mark IV," https://www.sut.org/wp-content/uploads/2015/09/Brendan-Hyland-v-3-Subsea-Control-Down-Under-Subsea-Internet-of-Things-final-v2.pdf, last time accessed: Sep. 2021.

[95] "EXRAY," https://www.hydromea.com/exray-wireless-underwater-drone/, last time accessed: Sep. 2021.

[96] "WiSub Maelstrom," http://www.wisub.com/products/maelstrom/, Last time accessed: Sep. 2021.

[97] "Power And Communication - Electrical Interfaces," https://www.bluelogic.no/products/electrical-interfaces, last time accessed: Sep. 2021.

[98] B. W. Hobson, R. S. McEwen, J. Erickson, T. Hoover, L. McBride, F. Shane, and J. G. Bellingham, "The development and ocean testing of an auv docking station for a 21" auv," in *Proc. IEEE/OES OCEANS*, Vancouver, Canada, Sep. 2007.

[99] N. Ahmed, J. Hoyt, A. Radchenko, D. Pommerenke, and Y. R. Zheng, "A multi-coil magneto-inductive transceiver for low-cost wireless sensor networks," in *Proc. IEEE UComms*, Sestri Levante, Italy, Sep. 2014.

[100] J.-F. Bousquet, A. A. Dobbin, and Y. Wang, "A compact low-power underwater magneto-inductive modem," in *Proc. ACM WUWNet*, Shanghai, China, Oct. 2016.

[101] N. Ahmed, A. Radchenko, D. Pommerenke, and Y. R. Zheng, "Design and evaluation of low-cost and energy-efficient magneto-inductive sensor nodes for wireless sensor networks," *IEEE Systems Journal*, vol. 131, no. 2, pp. 1135–1144, Jun. 2019.

[102] J. J. Sojdehei, P. N. Wrathall, and D. F. Dinn, "Magneto-inductive (MI) communications," in *Proc. MTS/IEEE OCEANS*, Honolulu, USA, Aug. 2001.

[103] "World Heritage Grimeton Radio Station," https://grimeton.org/?lang=en, last time accessed: Sep. 2021.

[104] "The World's Largest 'Radio' Station," https://pages.hep.wisc.edu/~prepost/ELF.pdf, last time accessed: Sep. 2021.

[105] "Extremely Low Frequency Transmitter Site Clam Lake, Wisconsin," https://fas.org/nuke/guide/usa/c3i/fs_clam_lake_elf2003.pdf, last time accessed: Sep. 2021.

[106] "ZEVS, The Russian 82 Hz Elf Transmitter," http://www.vlf.it/zevs/zevs.htm, last time accessed: Sep. 2021.

[107] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, USA, May 2005.

[108] F. Chiariotti, A. Signori, F. Campagnaro, and M. Zorzi, "Underwater jamming attacks as incomplete information games," in *Proc. IEEE International Workshop on Wireless Communications and Networking in Extreme Environments (WCNEE INFOCOM Workshop)*, Jul. 2020.

[109] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Proc. The Computer Security Foundations Workshop*, Franconia, NH, USA, Jun. 1994, pp. 187–191.

[110] S. Na, D. Hwang, W. Shin, and K.-H. Kim, "Scenario and countermeasure for replay attack using join request messages in lorawan," in *Proc. IEEE International Conference on Information Networking (ICOIN)*, Da Nang, Vietnam, Jan. 2017.

[111] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sinkhole attack in wireless sensor networks; the intruder side," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Avignon, France, Oct. 2008, pp. 526–531.

[112] E. C. H. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Proc. IEEE International Conference on Communications*, vol. 8, Istanbul, Turkey, Jun. 2006, pp. 3383–3389.

[113] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, "Denial of service defence for resource availability in wireless sensor networks," *IEEE Access*, vol. 6, pp. 6975–7004, Jan. 2018.

[114] F. Campagnaro, D. Tronchin, A. Signori, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, and M. Zorzi, "Replay-attack countermeasures for underwater acoustic networks," in *Proc. MTS/IEEE OCEANS*, Virtual, Global Oceans Singapore – U.S. Gulf Coast, Oct. 2020.

[115] A. Signori, F. Chiariotti, F. Campagnaro, and M. Zorzi, "A game-theoretic and experimental analysis of energy-depleting underwater jamming attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9793–9804, Oct. 2020.

[116] X. Li, M. R. Lyu, and J. Liu, "A trust model based routing protocol for secure ad hoc networks," in *Proc. IEEE Aerospace Conference*, vol. 2, Big Sky, MT, USA, 2004, pp. 1286–1295.

[117] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[118] C. Lal, R. Petroccia, M. Conti, and J. Alves, "Secure underwater acoustic networks: Current and future research directions," in *Proc. IEEE UComms*, Lerici, Italy, Aug. 2016.

[119] L. Ma, C. Fan, W. Sun, and G. Qiao, "Comparison of jamming methods for underwater acoustic DSSS communication systems," in *Proc. IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Xi'an, China, Mar. 2018.

[120] M. Samir, M. Kowalski, S. Zhou, and Z. Shi, "An experimental study of effective jamming in underwater acoustic links," in *Proc. IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, Philadelphia, PA, USA, Oct. 2014, pp. 737–742.

[121] M. C. Domingo, "Securing underwater wireless communication networks," *IEEE Wireless Communications*, vol. 18, no. 1, pp. 22–28, Feb. 2011.

[122] K. Grover, A. Lim, and Q. Yang, "Jamming and anti–jamming techniques in wireless networks: a survey," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 17, no. 4, pp. 197–215, Jan. 2014.

[123] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *Proc. IEEE Information Assurance Workshop (IAW)*, West Point, NY, USA, Jun. 2005, pp. 356–364.

[124] L. Chen and J. Leneutre, "Fight jamming with jamming–a game theoretic analysis of jamming attack in wireless networks and defense strategy," *Computer Networks*, vol. 55, no. 9, pp. 2259–2270, Mar. 2011.

[125] G. Alnifie and R. Simon, "A multi-channel defense against jamming attacks in wireless sensor networks," in *Proc. ACM workshop on QoS and Security for Wireless and Mobile Networks*, Chania, Crete Island, Greece, Oct. 2007.

[126] R. K. Mallik, R. A. Scholtz, and G. P. Papavassilopoulos, "Analysis of an on-off jamming situation as a dynamic game," *IEEE Transactions on Communications*, vol. 48, no. 8, pp. 1360–1373, Aug. 2000.

[127] B. DeBruhl, C. Kroer, A. Datta, T. Sandholm, and P. Tague, "Power napping with loud neighbors: Optimal energy-constrained jamming and anti-jamming," in *Proc. ACM Conference on Security and Privacy in Wireless & Mobile Networks*. Oxford, United Kingdom: ACM, Jul. 2014.

[128] F. Chiariotti, C. Pielli, N. Laurenti, A. Zanella, and M. Zorzi, "A game-theoretic analysis of energy-depleting jamming attacks with a learning counterstrategy," *ACM Transactions on Sensor Networks (TOSN)*, vol. 16, no. 1, pp. 1–25, Nov. 2019.

[129] L. Xiao, D. Jiang, Y. Chen, W. Su, and Y. Tang, "Reinforcement-learning-based relay mobility and power allocation for underwater sensor networks against jamming," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 1148–1156, Jul. 2020.

[130] Y. Ye, Z. Peng, and X. Hong, "Active jamming for eavesdropping prevention in underwater wireless networks," in *Proc. ACM WUWNet*, Atlanta, GA, USA, Oct. 2019, pp. 1–5.

[131] Y. Huang, P. Xiao, S. Zhou, and Z. Shi, "A half-duplex self-protection jamming approach for improving secrecy of block transmissions in underwater acoustic channels," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4100–4109, Jun. 2016.

[132] M. Zuba, Z. Shi, Z. Peng, and J.-H. Cui, "Launching denial-of-service jamming attacks in underwater sensor networks," in *Proc. ACM WUWNet*, Seattle, Washington, Dec. 2011.

[133] M. Khatua and S. Misra, "CURD: Controllable reactive jamming detection in underwater sensor networks," *Pervasive and Mobile Computing*, vol. 13, pp. 203–220, Aug. 2014.

[134] K. Murakami, H. Suemitsu, and T. Matsuo, "Classification of repeated replay-attacks and its detection monitor," in *Proc. IEEE Global Conference on Consumer Electronics (GCCE)*, Nagoya, Japan, Oct. 2017.

[135] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proc. IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, 2009, pp. 911–918.

[136] S. Malladi, J. Alves-Foss, and R. Heckendorn, "On preventing replay attacks on security protocols," *Proc. International Conference on Security and Management*, Jun. 2002.

[137] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Commununications ACM*, vol. 24, no. 8, p. 533–536, Aug. 1981.

[138] F. Farha and H. Ning, "Enhanced timestamp scheme for mitigating replay attacks in secure zigbee networks," in *Proc. IEEE International Conference on Smart Internet of Things (SmartIoT)*, Tianjin, China, Aug. 2019, pp. 469–473.

[139] F. Farha, H. Ning, S. Yang, J. Xu, W. Zhang, and K. R. Choo, "Timestamp scheme to mitigate replay attacks in secure zigbee networks," *IEEE Transactions on Mobile Computing*, Jul. 2020, Early Access.

[140] D. Jinwala, D. Patel, S. Patel, and K. S. Dasgupta, "Replay protection at the link layer security in wireless sensor networks," in *Proc. WRI World Congress on Computer Science and Information Engineering*, vol. 1, Los Angeles, CA, USA, Mar. 2009, pp. 160–165.

[141] J. Cho, A. Swami, and I. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 562–583, Oct. 2011.

[142] A. Boukerche, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11, pp. 2413–2427, Sep 2007.

[143] F. Oliviero and S. P. Romano, "A reputation-based metric for secure routing in wireless mesh networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 2008.

[144] J. Du, G. Han, C. Lin, and M. Martinez-Garcia, "ITrust: An anomaly-resilient trust model based on isolation forest for underwater acoustic sensor networks," *IEEE Transactions on Mobile Computing (Early Access)*, 2020.

[145] M. M. Arifeen, A. A. Islam, M. M. Rahman, K. A. Taher, M. M. Islam, and M. S. Kaiser, "ANFIS based trust management model to enhance location privacy in underwater wireless sensor networks," in *Proc. IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh, 2019.

[146] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. ACM international conference on Mobile computing and networking (MobiCom)*, 2000, pp. 255–265.

[147] A. Jøsang, "Artificial reasoning with subjective logic," in *Proc. the second Australian workshop on commonsense reasoning*, vol. 48, 1997, p. 34.

[148] S. Buchegger and J. L. Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proc. Second Workshop on Economics of P2P Systems*, Boston, MA, USA, 2004.

[149] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. IEEE UComms*, Lerici, Italy, Sep. 2016.

[150] F. Guerra, P. Casari, and M. Zorzi, "World Ocean Simulation System (WOSS): a simulation tool for underwater networks with realistic propagation modeling," in *Proc. ACM WUWNet*, Berkeley, CA, Nov 2009.

[151] G. Toso, R. Masiero, P. Casari, M. Komar, O. Kebkal, and M. Zorzi, "Revisiting source routing for underwater networking: The sun protocol," *IEEE Access*, vol. 6, pp. 1525–1541, Dec. 2018.

[152] International Chamber of Shipping, "Key Facts," http://www.ics-shipping.org/shipping-facts/key-facts, Last time accessed: Sep. 2021.

[153] T. Zugno, F. Campagnaro, and M. Zorzi, "Controlling in real-time an ASV-carried ROV for quay wall and ship hull inspection through wireless links in harbor environments," in *Proc. MTS/IEEE OCEANS*, Virtual, Global Oceans Singapore – U.S. Gulf Coast, Sep. 2020.

[154] D. Zordan, F. Campagnaro, and M. Zorzi, "On the feasibility of an anti-grounding service with autonomous surface vessels," in *Proc. MTS/IEEE OCEANS*, Marseille, France, Apr. 2019.

[155] F. Favaro, P. Casari, F. Guerra, and M. Zorzi, "Data upload from a static underwater network to an AUV: Polling or random access?" in *Proc. MTS/IEEE OCEANS*, Yeosu, Republic of Korea, May 2012.

[156] W. Liu, J. Weaver, L. Weaver, T. Whelan, R. Bagrodia, P. A. Forero, and J. Chavez, "APOLL: Adaptive polling for reconfigurable underwater data collection systems," in *Proc. MTS/IEEE OCEANS*, Kobe, Japan, May 2018.

[157] W. Rizzo, A. Signori, F. Campagnaro, and M. Zorzi, "Auvs telemetry range extension through a multimodal underwater acoustic network," in *Proc. IEEE/MTS OCEANS*, Charleston, US, Oct. 2018.

[158] R. Diamant, R. Francescon, and M. Zorzi, "Topology-efficient discovery: A topology discovery algorithm for underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 1200–1214, Jun. 2018.

[159] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of mobile data networks*. Cambridge University Press, 2016.

[160] R. Diamant, "Robust interference cancellation of chirp and cw signals for underwater acoustics applications," *IEEE Access*, vol. 6, pp. 4405–4415, Jan 2018.

[161] Aquarian Audio & Scientific, "AS-1 Hydrophone," http://www.aquarianaudio.com/as-1-hydrophone.html, last time accessed: Sep. 2021.

[162] F. Steinmetz and C. Renner, " Resilience against Shipping Noise and Interference in Low-Power Acoustic Underwater Communication ," in *Proc. MTS/IEEE OCEANS*, Seattle, WA, USA, Oct. 2019.

[163] F. Campagnaro, F. Steinmetz, A. Signori, D. Zordan, B.-C. Renner, and M. Zorzi, "Data collection in shallow fresh water scenarios with low-cost underwater acoustic modems," in *Proc. UACE2019*, Crete, Greece, Jul. 2019.

[164] A. Signori, F. Campagnaro, D. Zordan, F. Favaro, and M. Zorzi, "Underwater acoustic sensors data collection in the robotic vessels as-a-service project," in *Proc. MTS/IEEE OCEANS*, Marseille, France, Jun. 2019.

[165] J.-Y. Le Boudec, *Performance evaluation of computer and communication systems*. Epfl Press, 2011.

[166] C. Delea, E. Coccolo, S. F. Covarrubias, F. Campagnaro, F. Favaro, R. Francescon, V. Schneider, J. Oeffner, and M. Zorzi, "Communication infrastructure and cloud computing in robotic vessel as-a-service application," in *Proc. MTS/IEEE OCEANS*, Virtual, Global Oceans Singapore – U.S. Gulf Coast, Sep. 2020.

[167] R. Francescon, F. Campagnaro, E. Coccolo, A. Signori, F. Guerra, F. Favaro, and M. Zorzi, "An event-based stack for data transmission through underwater multimodal networks," in *Proc. UComms*, Lerici, Italy, Sep. 2021.

[168] "Mikrotik," last time accessed: Sep. 2021. [Online]. Available: https://mikrotik.com/

[169] "Mikrotik Manual:Interface/Wireless," last time accessed: Sep. 2021. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:Interface/Wireless

[170] "Mikrotik Metal 52AC," last time accessed: Sep. 2021. [Online]. Available: https://mikrotik.com/product/RBMetalG-52SHPacn

[171] "Mikrotik mANTBox," last time accessed: Sep. 2021. [Online]. Available: https://mikrotik.com/product/mantbox_2_12s

[172] "Mikrotik hEX POE lite," last time accessed: Sep. 2021. [Online]. Available: https://mikrotik.com/product/RB750UPr2

[173] "Mikrotik wAP R," last time accessed: Sep. 2021. [Online]. Available: https://mikrotik.com/product/RBwAPR-2nD

[174] V. Kebkal, K. Kebkal, O. Kebkal, and M. Komar, "Experimental results of delay-tolerant networking in underwater acoustic channel using s2c modems with embedded sandbox on-board," in *Proc. MTS/IEEE OCEANS*, Genova, Italy, 2015, pp. 1–6.

[175] F. Bekkadal, "Emerging maritime communications technologies," in *Proc. International Conference on Intelligent Transport Systems Telecommunications (ITST)*, Lille, France, Oct. 2009.

[176] Y. Xu, "Quality of service provisions for maritime communications based on cellular networks," *IEEE Access*, vol. 5, pp. 23 881–23 890, Oct. 2017.

[177] LoRa alliance, "LoRaWAN™1.1 Specification," https://www.lora-alliance.org/, Last time accessed: Sep. 2021.

[178] "ns3 network simulator," https://www.nsnam.org/, Last time accessed: Sep. 2021.

[179] D. Magrin, M. Capuzzo, and A. Zanella, "A Thorough Study of LoRaWAN Performance Under Different Parameter Settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, Jan. 2020.

[180] G. Tuna and V. C. Gungor, "A survey on deployment techniques, localization algorithms, and research challenges for underwater acoustic sensor networks," *International Journal of Communication Systems*, vol. 30, no. 17, Nov. 2017, e3350.

[181] M. Zuba, Z. Shi, Z. Peng, J.-H. Cui, and S. Zhou, "Vulnerabilities of underwater acoustic networks to denial-of-service jamming attacks," *Security and Communication Networks*, vol. 8, no. 16, pp. 2635–2645, Nov. 2015.

[182] G. Han, J. Jiang, N. Sun, and L. Shu, "Secure communication for underwater acoustic sensor networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 54–60, Aug. 2015.

[183] H. Luo, K. Wu, and F. Hong, "Ocean barrier: A floating intrusion detection ocean sensor networks," in *Proc. IEEE International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Hefei, China, Dec. 2016, pp. 390–394.

[184] S. Bagali and R. Sundaraguru, "Efficient channel access model for detecting reactive jamming for underwater wireless sensor network," in *Proc. IEEE International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, Mar. 2019, pp. 196–200.

[185] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, Dec. 2009.

[186] L. Xiao, X. Wan, W. Su, Y. Tang *et al.*, "Anti-jamming underwater transmission with mobility and learning," *IEEE Communications Letters*, vol. 22, no. 3, pp. 542–545, Jan. 2018.

[187] J. Lin, W. Su, L. Xiao, and X. Jiang, "Adaptive modulation switching strategy based on Q-learning for underwater acoustic communication channel," in *13th International Conference on Underwater Networks & Systems*. ACM, Dec. 2018, pp. 1–5.

[188] P. Casari and M. Rossi and M. Zorzi, "Towards Optimal Broadcasting Policies for HARQ based on Fountain Codes in Underwater Networks," in *Proc. IEEE/IFIP WONS*, Garmisch-Partenkirchen, Germany, Jan. 2008.

[189] A. Rubinstein, A. Tversky, and D. Heller, "Naive strategies in competitive games," in *Understanding strategic interaction*. Springer, 1997, pp. 394–402.

[190] D. Abreu, "On the theory of infinitely repeated games with discounting," *Econometrica: Journal of the Econometric Society*, pp. 383–396, Mar. 1988.

[191] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, Sep. 1951.

[192] Y.-p. Lee, S. Yoo, S. Y. Kim, and S. Yoon, "Anti-jamming performance analysis of CSS-based communication systems," in *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, 2008, pp. 101–104.

[193] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[194] C. E. Lemke and J. T. Howson, Jr, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, Jun. 1964.

[195] R. Savani and B. Von Stengel, "Hard-to-solve bimatrix games," *Econometrica*, vol. 74, no. 2, pp. 397–429, Mar. 2006.

[196] F. Campagnaro, R. Francescon, O. Kebkal, P. Casari, K. Kebkal, and M. Zorzi, "Full reconfiguration of underwater acoustic networks through low-level physical layer access," in *Proc. ACM WUWNet*, Halifax, Canada, Nov. 2017.

[197] A. Signori, C. Pielli, F. Chiariotti, F. Campagnaro, M. Giordani, N. Laurenti, and M. Zorzi, "Jamming the underwater: a game-theoretic analysis of energy-depleting jamming attacks," in *Proc. ACM WUWNet*, Atlanta, GA, USA, Oct. 2019.

[198] J. C. Harsanyi, "Games with incomplete information played by "Bayesian" players part II. Bayesian equilibrium points," *Management Science*, vol. 14, no. 5, pp. 320–334, Jan. 1968.

[199] O. Kebkal, M. Komar, K. Kebkal, and R. Bannasch, "D-MAC: Media access control architecture for underwater acoustic sensor networks," in *Proc. IEEE/OES OCEANS*, Santander, Spain, Jun. 2011.

[200] L. Wu, J. Trezzo, D. Mirza, P. Roberts, J. Jaffe, Y. Wang, and R. Kastner, "Designing an adaptive acoustic modem for underwater sensor networks," *IEEE Embedded Systems Letters*, vol. 4, no. 1, pp. 1–4, Mar. 2012.

[201] R. Petroccia, G. Cario, M. Lupia, V. Djapic, and C. Petrioli, "First in-field experiments with a "bilingual" underwater acoustic modem supporting the janus standard," in *Proc. MTS/IEEE OCEANS*, Genova, Italy, 2015, pp. 1–7.

[202] R. Petroccia, J. Alves, and G. Zappa, "Janus-based services for operationally relevant underwater applications," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 994–1006, Oct. 2017.

[203] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The JANUS underwater communications standard," in *Proc. IEEE UComms.* IEEE, Sep. 2014.

[204] W. W. Peterson and E. J. Weldon, *Error-correcting codes.* MIT press, 1972.

[205] R. Askey, *Orthogonal polynomials and special functions.* SIAM, Jan. 1975.

[206] L. Xiao, Q. Li, T. Chen, E. Cheng, and H. Dai, "Jamming games in underwater sensor networks with reinforcement learning," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[207] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, May 2016.

[208] M. Goetz, S. Azad, P. Casari, I. Nissen, and M. Zorzi, "Jamming-resistant multi-path routing for reliable intruder detection in underwater networks," in *Proc. ACM WUWNet*, Seattle, Washington, Dec. 2011.

[209] W. Wang, J. Kong, B. Bhargava, and M. Gerla, "Visualisation of wormholes in underwater sensor networks: A distributed approach," *ACM International Journal of Security and Networks*, vol. 3, no. 1, pp. 10–23, Jan. 2008.

[210] R. Zhang and Y. Zhang, "Wormhole-resilient secure neighbor discovery in underwater acoustic networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[211] G. Dini and A. L. Duca, "A secure communication suite for underwater acoustic sensor networks," *MDPI Sensors*, vol. 12, no. 11, pp. 15 133–15 158, Nov. 2012.

[212] A. Caiti, V. Calabro, G. Dini, A. Lo Duca, and A. Munafo, "Secure cooperation of autonomous mobile sensors using an underwater acoustic network," *MDPI Sensors*, vol. 12, no. 2, pp. 1967–1989, Feb. 2012.

[213] Y. Liu, J. Jing, and J. Yang, "Secure underwater acoustic communication based on a robust key generation scheme," in *Proc. IEEE International Conference on Signal Processing*, Beijing, China, Oct. 2008, pp. 1838–1841.

[214] G. Ateniese, A. Capossele, P. Gjanci, C. Petrioli, and D. Spaccini, "SecFUN: Security Framework for Underwater acoustic sensor Networks," in *Proc. MTS/IEEE OCEANS*, Genova, Italy, May 2015, pp. 1–9.

[215] N. Benvenuto and M. Zorzi, *Principles of Communications Networks and Systems*, 1st ed. Wiley, 2011.

[216] P. Casari, F. Campagnaro, E. Dubrovinskaya, R. Francescon, A. Dagan, S. Dahan, M. Zorzi, and R. Diamant, "ASUNA: A topology dataset for underwater network emulation," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 1, pp. 307–318, Jan. 2021.

[217] A. D. Wood and J. A. Stankovic, "A taxonomy for denial-of-service attacks in wireless sensor networks," *Handbook of sensor networks: compact wireless and wired sensing systems*, pp. 739–763, 2004.

[218] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, Jan. 2008.

[219] J. Antunes, N. F. Neves, and P. J. Veríssimo, "Detection and prediction of resource-exhaustion vulnerabilities," in *Proc. IEEE International Symposium on Software Reliability Engineering (ISSRE)*, Seattle, WA, USA, 2008, pp. 87–96.

[220] A. Mathew and J. S. Terence, "A survey on various detection techniques of sinkhole attacks in WSN," in *Proc. IEEE International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2017, pp. 1115–1119.

[221] S. Dietzel, R. van der Heijden, H. Decke, and F. Kargl, "A flexible, subjective logic-based framework for misbehavior detection in V2V networks," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Sydney, NSW, Australia, Jun. 2014.

[222] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Network*, vol. 4, no. 3, Jun. 2008.

[223] B. Tomasi, P. Casari, L. Finesso, G. Zappa, K. McCoy, and M. Zorzi, "On modeling JANUS packet errors over a shallow water acoustic channel using Markov and hidden Markov models," in *Proc. IEEE Military Communications Conference (MILCOM)*, San Jose, CA, USA, Nov. 2010.

[224] R. J. Urick, *Principles of Underwater Sound*, 3rd ed. McGraw-Hill, 1983.

[225] F. Pignieri, F. De Rango, F. Veltri, and S. Marano, "Markovian approach to model underwater acoustic channel: Techniques comparison," in *Proc. IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, Nov. 2008.

[226] D. A. Sanchez-Salas and J. L. Cuevas-Ruiz, "N-states channel model using Markov chains," in *Proc. IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, Cuernavaca, Mexico, Sep. 2007.

[227] W. Turin and R. van Nobelen, "Hidden Markov modeling of flat fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1809–1817, Dec. 1998.

[228] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, Sep. 1960.

[229] M. Zorzi, R. Rao, and L. Milstein, "Error statistics in data transmission over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468–1477, Nov. 1998.

[230] ——, "On the accuracy of a first-order markov model for data transmission on fading channels," in *Proc. IEEE International Conference on Universal Personal Communications*, Tokyo, Japan, 1995, pp. 211–215.

[231] J. G. Ruiz, B. Soret, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "On finite state Markov chains for Rayleigh channel modeling," in *Proc. IEEE International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, Aalborg, Denmark, May 2009.

[232] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, 1999.

List of Publications

Publications on International Journals

[J1] A. Signori, F. Campagnaro, F. Steinmetz, C. Renner, M. Zorzi, "Data gathering from a multimodal dense underwater acoustic sensor network deployed in shallow fresh water scenarios," *MDPI Journal of Sensor and Actuator Networks (JSAN)*, vol. 8, no. 4, p. 55, Dec. 2019.

[J2] A. Signori, F. Chiariotti, F. Campagnaro, M. Zorzi, "A Game-Theoretic and Experimental Analysis of Energy-Depleting Underwater Jamming Attacks," *IEEE Internet of Things Journal (IOTJ)*, vol. 7, no. 10, pp. 9793–9804, Oct. 2020.

[J3] F. Campagnaro, A. Signori, M. Zorzi, "Wireless Remote Control for Underwater Vehicles," *MDPI Journal of Marine Science and Engineering (JMSE)*, vol. 8, no. 10, p. 736, Oct. 2020.

[J4] E. Coccolo, C. Delea, F. Steinmetz, R. Francescon, A. Signori, C. N. Au, F. Campagnaro, V. Schneider, F. Favaro, J. Oeffner, C. Renner, M. Zorzi, "System Architecture and Communication Infrastructure for the RoboVaaS project," *submitted to IEEE Journal of Oceanic Engineering (JOE)*, 2021.

[J5] A. Signori, F. Chiariotti, F. Campagnaro, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, M. Zorzi, "A Geometry-Based Game Theoretical Model of Blind and Reactive Underwater Jamming," *IEEE Journal on Wireless Communications (TWC)*, Early Access, 2021.

[J6] A. Signori, F. Campagnaro, I. Nissen, M. Zorzi, "Channel-Based Trust Model for Security in Underwater Acoustic Networks," *submitted to IEEE Internet of Things Journal (IOTJ)*, 2021.

Publications on Conference and Workshop Proceedings

[C1] E. Coccolo, F. Campagnaro, A. Signori, F. Favaro, M. Zorzi "Implementation of AUV and Ship Noise for Link Quality Evaluation in the DESERT Underwater

Framework", in *Proc. ACM WUWNet*, Shenzhen, China, Dec. 2018.

[**C2**] A. Signori, F. Campagnaro, D. Zordan, F. Favaro, and M. Zorzi, "Underwater acoustic sensors data collection in the robotic vessels as-a-service project," in *Proc. MTS/IEEE OCEANS*, Marseille, France, Jun. 2019.

[**C3**] F. Campagnaro, F. Steinmetz, A. Signori, D. Zordan, C. Renner, and M. Zorzi, "Data collection in shallow fresh water scenarios with low-cost underwater acoustic modems," in *Proc. UACE*, Crete, Greece, Jul. 2019.

[**C4**] A. Signori, C. Pielli, F. Chiariotti, F. Campagnaro, M. Giordani, N. Laurenti, and M. Zorzi, "Jamming the underwater: a game-theoretic analysis of energy-depleting jamming attacks," in *Proc. ACM WUWNet*, Atlanta, GA, USA, Oct. 2019.

[**C5**] F. Chiariotti, A. Signori, F. Campagnaro, M. Zorzi, "Underwater Jamming Attacks as Incomplete Information Games," in *Proc. IEEE International Workshop on Wireless Communications and Networking in Extreme Environments (WCNEE INFOCOM Workshop)*, Jul. 2020.

[**C6**] D. Magrin, A. Signori, D. Tronchin, F. Campagnaro, M. Zorzi, "Collaboration of LoRaWAN and Underwater Acoustic Communications in Sensor Data Collection Applications," in *Proc. MTS/IEEE OCEANS*, Oct. 2020.

[**C7**] A. Signori, F. Campagnaro, K. Wachlin, I. Nissen, M. Zorzi, "On the Use of Conversation Detection and Traffic Classification to Improve the Security of Underwater Acoustic Networks," in *Proc. MTS/IEEE Oceans*, Oct. 2020.

[**C8**] F. Campagnaro, D. Tronchin, A. Signori, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, and M. Zorzi, "Replay-Attack Countermeasures for Underwater Acoustic Networks," in *Proc. MTS/IEEE OCEANS*, Oct. 2020.

[**C9**] R. Francescon, F. Campagnaro, E. Coccolo, A. Signori, F. Guerra, F. Favaro, M. Zorzi, "An Event-Based Stack For Data Transmission Through Underwater Multimodal Networks," in *Proc. IEEE UComms*, Aug. 2021.

[**C10**] D. Tronchin, R. Francescon, F. Campagnaro, A. Signori, R. Petroccia, K. Pelekanakis, R. Paglierani, J. Alves, M. Zorzi, "A Secure Cross-Layer Communication Stack for Underwater Acoustic Networks," in *Proc. MTS/IEEE Oceans*, Sep. 2021.

[**C11**] F. Campagnaro, A. Signori, R. Otnes, M. Goetz, D. Sotnik, A. Komulainen, I. Nissen, F. Favaro, F. Guerra, M. Zorzi, "A Simulation Framework for Smart Adaptive Long-and Short-range Acoustic Networks," in *Proc. MTS/IEEE Oceans*, Sep. 2021.

[**C12**] A. Signori, E. Coccolo, F. Campagnaro, I. Nissen, M. Zorzi, "Trustworthiness in the GUWMANET Protocol for Underwater Acoustic Mobile Ad-Hoc Networks," in *Proc. ACM WUWNet*, Nov. 2021.