frontiers | Frontiers in Communications and Networks

TYPE Original Research
PUBLISHED 30 May 2023
DOI 10.3389/frcmn.2023.1179626

Network intelligence *vs.* jamming in underwater networks: how learning can cope with misbehavior

J. S. Mertens[1,2], A. Panebianco[2,3], A. Surudhi[4], N. Prabagarane[5] and L. Galluccio[1,2]*

[1]Department of Electrical Electronic and Computer Engineering, University of Catania, Italy, [2]CNIT Research Unit at University of Catania, Catania, Italy, [3]Department of Engineering, University of Palermo, Italy, [4]Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, United States, [5]Department of ECE Sri Sivasubramaniya Nadar College of Engineering, Chennai, India

Check for updates

OPEN ACCESS

EDITED BY
Filippo Campagnaro,
University of Padua, Italy

REVIEWED BY
Sara Falleni,
Northeastern University, United States
Paolo Casari,
University of Trento, Italy
Stefano Basagni,
Northeastern University, United States

*CORRESPONDENCE
L. Galluccio,
laura.galluccio@unict.it

RECEIVED 04 March 2023
ACCEPTED 15 May 2023
PUBLISHED 30 May 2023

CITATION
Mertens JS, Panebianco A, Surudhi A, Prabagarane N and Galluccio L (2023), Network intelligence *vs.* jamming in underwater networks: how learning can cope with misbehavior.
Front. Comms. Net 4:1179626.
doi: 10.3389/frcmn.2023.1179626

In this paper, we present a machine-learning technique to counteract jamming attacks in underwater networks. Indeed, this is relevant in security applications where sensor devices are located in critical regions, for example, in the case of national border surveillance or for identifying any unauthorized intrusion. To this aim, a multi-hop routing protocol that relies on the exploitation of a Q-learning methodology is presented with a focus on increasing reliability in data communication and network lifetime. Performance results assess the effectiveness of the proposed solution as compared to other efficient state-of-the-art approaches.

KEYWORDS

underwater networks, jamming attacks, Q-learning, routing, Markov model

1 Introduction

War events in the last year have highlighted the relevance of supporting surveillance applications in both terrestrial and underwater scenarios. Today, this is a primary concern from the perspective of international cooperation groups that work to guarantee the security of national borders against external aggressions. Situation awareness and prompt alert communication have, thus, become a priority in the context of terrestrial and underwater scenarios. From this perspective, recently, some relevant papers have been published. As an example, in Akyildiz et al. (2005), the exploitation of both static and mobile autonomous underwater vehicles (AUVs) equipped with heterogeneous sensors is proposed to monitor either the seabed or the sea surface to guarantee prompt situation awareness. Further work in this direction was proposed in PAO (2017) by CMRE NATO. In this case, a hybrid multistatic network is proposed; this includes active sonars placed on the surface communicating directly with AUVs moving underwater for controlling purposes.

By way of acoustic, optical, or magnetic sensors, these underwater devices can identify unsafe and abnormal conditions and promptly send alerts to security centers to stimulate appropriate actions. These sensors, which we will simply denote as *nodes*, will autonomously and naturally set up a communication network around one or more surface gateway nodes to invoke the support of surface vessels to execute appropriate security operations.

Despite the different types of security threats, jamming attacks are among the most critical since they cause communication disruption and can block transmissions at the physical level. Although different underwater communication technologies are available (acoustic, optical, or radio frequency), in this paper, we focus specifically on acoustic solutions. We consider different types of potential jamming attacks, caused, for example, by intruders or jammers who deliberately try to disrupt communications and contrast transmissions at the physical layer. To counteract this action, we propose to exploit path redundancy to increase the probability of successfully detecting an intrusion or a hazard that threatens a region. To this aim, a technique that relies on the use of machine learning and, specifically, reinforcement learning, to preserve reliable delivery, while reducing energy consumption, is proposed. Considering that the batteries of underwater nodes cannot be replaced or recharged, and should be long-lasting, it is important to decrease the energy consumption of individual nodes, thus increasing the network lifetime. This is also in consideration of the fact that jammers intentionally send acoustic signals with the aim of depleting target sensor batteries, leading to denial-of-service attacks. Furthermore, due to the intrinsic complexity of underwater scenarios, characterized by long propagation delay, low bit rate, and high error probability, jointly solving communication and security issues is a challenging problem and a primary concern.

From this perspective, in this work, we present a machine-learning framework to support efficient underwater communications in noisy and insecure environments. Specifically, underwater devices that endure several channel impairments, for example, possible bad channel conditions, noise, and/or ongoing jamming actions, employ a Q-learning approach relying on Markov decision processes to make optimal relay choices by taking into account not only their residual energy but also the average residual energy at nodes in the area. This is the aim of also pursuing fair energy balancing inside the network. Accordingly, a trade-off between energy consumption, delivery delay, and network lifetime is achieved. The efficiency of the proposed joint approach is finally assessed through simulations and comparisons with state-of-the-art efficient solutions.

Compared to previous literature on jamming in underwater networks, our contributions are multiple:

- We consider the problem of jamming in underwater networks from the perspective of multi-hop routing designs;
- We introduce a Q-learning routing methodology for underwater networks impaired by an ongoing jamming action, while also detailing the set of procedures characterizing its functioning;
- Our solution jointly addresses the problem of data delivery and energy consumption fairness among network nodes, to allow increasing network lifetime, while not unfairly exhausting node batteries;
- We incorporate an underwater channel model and test the effectiveness of the proposed approach;
- We provide an extensive study of the impact of the jammer position on the efficacy of the anti-jamming procedure while also comparing the effectiveness of our methodology to the one achievable by other state-of-the-art solutions, such as Zhang et al. (2021);
- We analyze the impact of the proposed approach on multiple performance metrics, such as energy efficiency, latency, and delivery rate, showing the stability of the proposed approach compared to existing alternative solutions.

The rest of the paper is organized as follows. Some preliminary literature in the field of underwater jamming is discussed in Section 2. In Section 3, we provide a description of the considered system. In Section 4, we illustrate, in detail, the jamming action. In Section 5, we detail the distributed communication and routing protocol employed by underwater nodes. In Section 6, we present the considered Q-learning framework, while in Section 7, we describe the Markov model assumed for describing the underwater channel. In Section 8, we provide numerical results to assess the effectiveness of the proposed approach, also in comparison with state-of-the-art solutions. Finally, in Section 9, conclusions and considerations on future directions of the work are drawn.

2 Related work

Underwater acoustic network applications typically span from national border security and control to environmental and marine wildlife monitoring.

Due to the critical challenges posed by marine scenarios, the design of underwater (UW) communication networks is hard, both from the point of view of hardware features and from the perspective of reliable and robust communication protocols (Akyildiz et al., 2005). At the physical level, three main technologies can be employed (acoustics, optical, and radio frequency technologies), but all highlight the need to trade off hardware costs, bandwidth, and coverage. To this aim, the most widespread commercial approach is to employ acoustic communications. However, the low available bandwidth offered by acoustic communications poses challenges in terms of the amount of information that could be transmitted. At the channel level, the UW medium itself is highly time-varying, unstable, and impaired by multipath propagation and fading. Furthermore, the long propagation delay (in the order of 1,500 m/s) and the Doppler effect, which are intrinsic to the relative mobility of nodes, can make the design even more complex. Concerning also the practical deployment of underwater networks, compared to terrestrial ones, additional difficulties related to equipment and installation costs emerge. An additional feature is related to robustness to corrosion, environmental disturbances, and damage, which must be supported upon acting in a wet, unattended, saline environment.

Within the area of UW network design, security and privacy have also attracted the interest of researchers. To this aim, in Dini and Lo Duca (2012), Caiti et al. (2012), and Liu et al. (2008), the authors present a comprehensive gateway security suite to protect the system from internal attacks, such as spoofing, replay, and Sybil attacks (Aman et al., 2023); (S.A.H.Mohsan et al., 2023). To protect the integrity and confidentiality of messages, cryptographic-based authentication can be used.

In Kulhandjian et al. (2014), the authors consider the problem of securing an underwater network through a cooperative jammer that employs CDMA-based analog network coding. The cooperative jammer sends information that is known to the receiver, which, thus, suppresses the interfering contribution and reconstructs the original information. Along the same line of reasoning, in Ye et al. (2020), the authors deal with the problem of preventing eavesdropping in underwater wireless networks, where it is not possible to use cryptographic techniques. To this aim, fictitious interference is created in the vicinity of the transmitter and receiver.

Coordinated multipoint communication (CoMP) is then used to ensure that the intended recipient receives the data correctly.

In Aman et al. (2023), an overview of techniques aimed at supporting security in underwater scenarios is presented. Similarly in Ahmad et al. (2021), the type of security threats and the solution mechanisms that can be employed at different levels of the protocol stack to solve those threats are discussed. In Samir et al. (2014), the authors also focus on vulnerabilities in underwater acoustic networks by specifically applying this to the case of commercial modems which use heterogeneous transmission schemes.

However, when focusing on privacy attacks, the jamming attack should also be specifically accounted for. Jamming is the act of deliberately disrupting signal transmission by emitting signals that either resemble the standard network traffic or disrupt authorized traffic. The effect of jamming is similar to that of noise, meaning that when the jamming effect increases, packets can be forced to travel longer routes or be diverted. As an effect of the jamming action, packets can be corrupted and/or sometimes lost, which leads to a decrease in the packet delivery ratio (PDR).

In Zuba et al. (2015), the authors investigate the characteristics of different jamming attack patterns on multiple commercial brand acoustic modems and a prototype multiplexing modem.

In Kalita and Sahu (2015), an uncoordinated direct sequence spread spectrum technique for managing receivers in the context of anti-jamming multi-channel underwater communications is presented, and multiple metrics are investigated in a simulated environment.

To cope with jamming, for example, *RACUN*, Wang et al. (2017) propose an online learning anti-jamming algorithm called a multi-armed bandit (MAB)-based acoustic channel access algorithm to achieve jamming-resilient acoustic communication. It implements a hidden Markov model (HMM), based on which the potential reward is estimated, along with an appropriate probability that the channel is jammed. Each underwater acoustic channel will be sensed according to this probability, the value of which is lower for those channels more prone to be jammed.

Jamming scenarios in underwater networks have been theoretically analyzed and modeled by way of different approaches. For example, in Xiao et al. (2014), a game theoretic framework is studied to model a jamming game. The communication between jammers and sensors is modeled as a game, and an anti-jamming power control policy is proposed to assist the sensors in choosing their transmission power without having any knowledge of the channel gain. A Bayesian zero-sum game is formulated in Vadori et al. (2015) to enable the sensor network to maximize the transmission capacity, despite the presence of a jammer that tries to disrupt communication. In this contribution, the resulting equilibrium due to the effects of the nodes' positions is investigated. In Goetz et al. (2011), a multipath routing protocol, in a jamming scenario for an underwater network working in the frequency band of 4–8 kHz, is addressed. Restricted flooding and adaptive source routing are selected to enable multipath transmissions and achieve jamming resilience by also taking into account the noise from the boat propellers, which interferes with this frequency band.

Reinforcement learning and deep learning have also been used to describe and model such jammed systems. For example, in Schmidhuber (2015), a historical survey summarizing relevant works on supervised and unsupervised learning, reinforcement learning, and evolutionary computation is presented. In Erpek et al. (2019), a cognitive transmitter uses a pre-trained classifier to predict the current channel status based on recent sensing results, based on which it decides whether to transmit or not. In Shi and Sagduyu (2017), the jammer node is modeled through a deep learning classifier, and the received signal strength is analyzed under different attack mechanisms.

In Di Valerio et al. (2019), a reinforcement learning methodology for forwarding data based on varying channel conditions is presented. This is based on the exploitation of multiple paths, which can be switched adaptively based on energy considerations and packet delivery ratio estimation. In Xiao et al. (2018), reinforcement learning is also considered in the framework of anti-jamming applications. More specifically, control of transmission power and mobility of nodes are used to counteract jamming. In Xiao et al. (2020), a deep reinforcement learning (RL)-based relay scheme is further employed to improve relay performance. In Signori et al., (2021) and Signori et al., (2020), a multistage game is presented to model a jamming scenario where the jammed node may use packet-level coding as a countermeasure against the attack. The authors also consider real experimental data and perform a sensitivity analysis to compare the effect of the real channel model compared to that of the modeled one. In Signori et al. (2022), the same authors analyze the effectiveness of a reactive and a blind jammer using a game theoretical model considering different scenario geometries. Various anti-jamming strategies are also compared (e.g., using additional energy to protect the communication or avoiding jamming signals by randomizing the transmission pattern).

In Xiong et al. (2020), jamming is also addressed from the perspective of the jammer's effectiveness. A distributed barrage jamming layout strategy is proposed. Detection performance is estimated using signal processing methods. Accordingly, the Cramer–Rao bound (CRB) of multiple targets estimated by an underwater sensor network with distributed jammers is calculated, which applies independently of the specific signal processing method. The proposed optimization is then solved numerically by using heuristics.

In Bagali and Sundaraguru (2019), cooperative jamming detection is proposed. In particular, a cross-layer Efficient Channel Access (ECA) model using cross-layer design for mitigating reactive jammer action is presented. By optimizing the cooperative hopping probabilities and channel accessibility probabilities of an authenticated sensor device, the effectiveness of this cooperative strategy, compared to that of non-cooperative ones, is proven.

In Su et al. (2022) and Xiao et al. (2015), collaborative jamming as a countermeasure to security threats is considered as well.

In Zhang et al. (2021), a reinforcement learning-based opportunistic routing protocol (RLOR) is proposed by combining opportunistic routing and the reinforcement learning algorithm. The RLOR consists of a distributed routing approach, which considers the peripheral location of the nodes to select the appropriate relay nodes and employs a recovery mechanism to enable the packets to bypass void areas efficiently.

However, none of the aforementioned works combines either consideration of the real features of the underwater communication

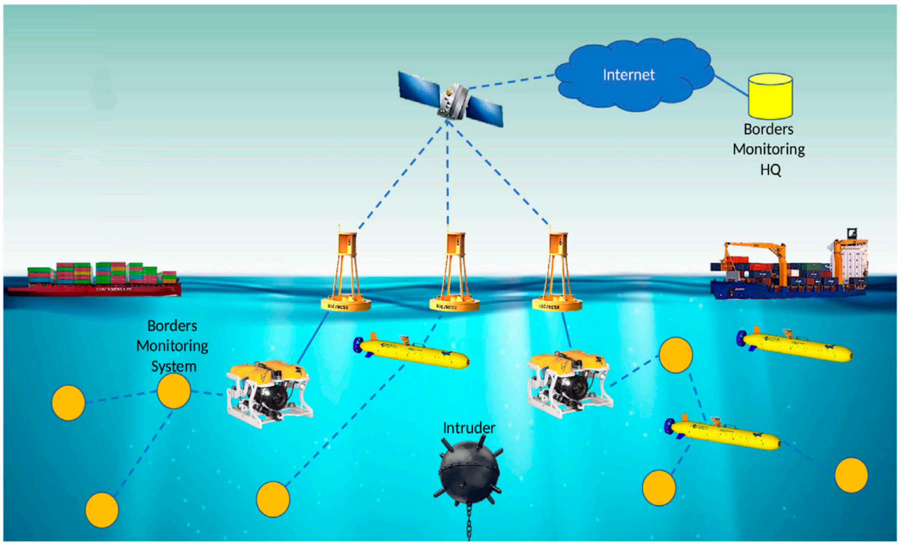

FIGURE 1
System scenario.

channel or estimates the effect of different jamming actions and jammers' positions on end-to-end data delivery. In this work, instead, in line with Hu and Fei (2010), we consider an underwater network under an intermittent jamming attack. By taking into account the realistic channel time variability of the underwater scenario, similar to the model presented in Pignieri et al. (2008), which considers realistic measurements and traces, we design a resilient protocol that facilitates the support of efficient and long-lasting data communication under ongoing jamming attacks.

In the following section, we will start by detailing the considered system scenario.

3 System architecture

Figure 1 illustrates the scenario we consider in our work. The network comprises diverse underwater devices that can sense, interpret, and respond to external conditions when they are activated remotely. Furthermore, we have considered the devices as heterogeneous in terms of their capabilities (e.g., complex underwater vehicles or simple sensing devices that have acoustic transducers embedded in them), as well as in the type of data they sense and process (e.g., images detailing the marine wildlife or data that correspond to the water salinity level and data revealing the differences in seawater acidity as a result of pollution due to fossil fuels, temperature, etc.). Furthermore, some of the considered devices are vehicles that are operated remotely by a surface vessel with the aid of a cable (e.g., a remotely operated vehicle (ROV)), and others are unmanned underwater vehicles (UUVs) that can move closer to the surface or deeper without any human intervention; other devices can also be static or cabled to a depth buoy. Moreover, the network of underwater devices is assumed to be interconnected with a terrestrial network or with border surveillance stations that are equipped with servers where the gathered data are elaborated and processed. Moreover, to meet these requirements, we have assumed the availability of surface buoys equipped with long-distance connection capabilities (e.g., cellular or satellite networks) heading towards coastal facilities that enable the connections of the underwater network with the terrestrial one. Even if optical and radio frequency technologies are also available for use in underwater scenarios at the cost of a very low transmission range due to attenuation in water or expensive hardware costs, in the rest of this work, we specifically consider the scenario where only acoustic technology is employed to support communication in the underwater network. This is justified by the relative hardware simplicity and cost reduction, as well as by the longer communication range they provide. This is the most widespread underwater technology that is employed for commercial purposes by major vendors (Evologics, 2023); (TeledyneMarine, 2023).

In the considered scenario, a malicious adversary node tries to avoid being identified as an intruder by network nodes and generates a jamming signal to disrupt communications in the underwater surveillance network. Accordingly, this jamming action makes it difficult to exercise any countermeasure to ensure border protection. Moreover, the scenario can be characterized by a high level of marine noise due to cargo, vessels, and maritime platforms navigating across the area. It should be noted that this type of noise can have relevant effects, especially in the range of low frequencies.

In the next section, we will detail how a jammer can execute a jamming action to disrupt communications in the addressed scenario. Then, we will detail the communication protocol executed by network nodes to cope with this type of risk and the mathematical framework that relies on Q-learning, which can be promiscuously combined with the communication protocol to increase the chances of performing successful data delivery at the destination.

4 Jamming

In a communication environment, jamming results in the deliberate disruption of signal transmissions by emitting interfering signals that either resemble standard network traffic or disrupt authorized traffic by acting similarly to noise.

Jammers use powerful devices to prevent the proper functioning of network nodes. If a jamming action is aimed at paralyzing a central node, for example, a base station, an access point, or a gateway, this can lead to a collapse of the entire network (Cong et al., 2010). In the context of underwater networks, jamming is, thus, a type of denial-of-service (DoS) attack that intentionally causes interference over the range of acoustic frequencies used by legitimate underwater network nodes (Shi and Perrig, 2004), thus prohibiting reliable data transfer.

Due to the high impact of this type of attack, the use of defense mechanisms to deal with jamming in underwater networks has gained the utmost importance. Furthermore, while designing countermeasures, one should take into account the constraints the underwater networks exhibit, like limited energy, low processing capability, limited memory, and vulnerability to physical capturing when insecure communication channels are used. In the following section, we will briefly recall the main types of jamming attacks which can threaten an underwater network.

4.1 Type of jamming attacks

Usually, in jamming attacks, jammers aim to decrease the signal-to-noise ratio (SNR) of legitimate transmissions.

One of the most popular methods used by jammers is *spot jamming* (Mpitziopoulos et al., 2009). In this case, a jammer works on a single frequency channel and employs all its transmitted power to make the original signal ineffective. It is a powerful jamming technique, which can, however, be easily overcome by switching to another frequency channel, once the legitimate users realize there is an ongoing attack. Another popular jamming technique is *sweep jamming*. It is quite different from spot jamming; in this case, the power of a jammer is employed to rapidly jam over different frequency channels. Although this jamming method is capable of jamming a range of frequencies, all of them are not affected simultaneously; hence, the jamming technique is not very effective. Nevertheless, this type of jamming may lead to sizeable packet loss, thereby resulting in re-transmissions that consume precious energy resources of legitimate users. This jamming technique on its own is much more complex and energy-consuming for the jammer node itself. Another type of widespread jamming attack is *barrage jamming*, which affects a range of frequencies simultaneously. However, as the number of frequencies increases, this jamming attack may become ineffective as the power emitted by the jammer on each frequency channel may be significantly reduced. *Deceptive jamming* is an alternative jamming attack, where a jammer can either jam over a single frequency or multiple frequency channels. In this case, a jammer that does not want to disclose its presence simply floods the network with legitimate data that resemble those of the network traffic to deceive the network defense mechanism, thereby misleading network functioning and causing a waste of bandwidth available for legitimate nodes that remain in the receiving mode only.

Another relevant classification among different types of jamming techniques can be carried out upon considering the activation of the jamming action (see Figure 2). From this perspective, different jamming actions are possible: constant jamming, random jamming, and reactive jamming. A *constant jammer* continuously radiates random radio signals. The jammer, indeed, intends to keep the channel busy by transmitting random bits, thereby causing interference to the communication that has already been commenced by a node, resulting in corruption of the packets or prevention for a node to access the channel. A *random jammer* instead moves to a sleep state for a random time interval *ts* and jams the network for another random time interval *tj*. Depending on the operating conditions, different levels of efficacy and power saving can be achieved by tuning the duration of time of sleep and jamming action. Another variant is *reactive jamming*. A reactive jammer listens to the activity in the channel. When a legitimate activity is detected, a reactive jammer instantaneously emits a random signal that collides with the useful signal on the channel, which leads to the corruption of the transmitted data packets.

By comparing these different types of jamming actions, it is evident that constant and reactive jammers are effective in causing the packet delivery ratio to plummet to almost 0, especially if they are in the proximity of victim nodes. However, constant jammers may exhaust their energy quickly, and, thus, reactive jammers are preferred from the perspective of jammer efficiency. However, due to long propagation delay, which is intrinsic in underwater scenarios, a reactive jammer could react too late, and, thus, its action can be ineffective. Consequently, it could be more efficient and simpler for the jammer to perform a random jamming action. Accordingly, in the rest of this paper, we assume that a jammer randomly jams a set of legitimate nodes in its proximity according to an average assigned duty cycle.

5 Communication and routing protocol

To counteract a possible jamming action performed by an intruder node, legitimate nodes belonging to the underwater network implement a communication and routing protocol, which will be detailed in the rest of this section. This protocol will exploit a machine-learning methodology to identify possible clues of ongoing jamming action. In Section 6, this Q-learning methodology will be detailed.

In the following section, we present the communication protocol employed by underwater sensor devices to transmit information to the gateway surface node to advertise the presence of possible intruder nodes. In particular, we will discuss how the availability of multiple paths from source to destination can reduce the impact of the jamming effect. Furthermore, considering that the propagation delay is high in seawater, the proposed communication protocol needs to execute a proactive action to

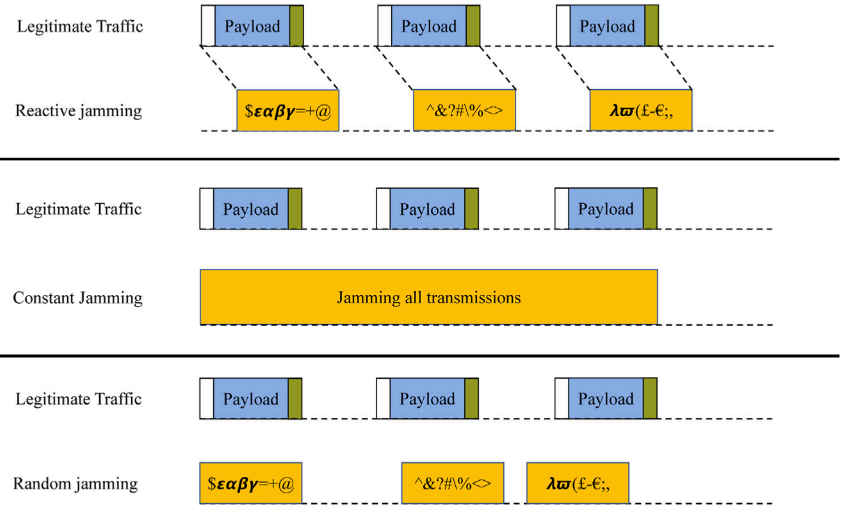

FIGURE 2
Possible mechanisms for jamming activation in a typical underwater network.

provide paths before the actual network operation begins. To this purpose, nodes implicitly or explicitly collect routing information to build the communication topology. In this way, paths can be established before they are used, thereby resulting in faster data delivery.

While an on-demand route discovery process would take too long to be executed due to the long propagation delay, to take into account the noise and jamming effects, the route establishment procedure is executed and updated every T_{up} seconds. It should be noted that tuning this timer can be useful in limiting the overhead associated with the route establishment procedure. If a neighbor node is not heard for more than T_{up} seconds, it is assumed that a topology change has occurred, possibly because a node is no longer available or has been jammed. In this context, the node is considered no longer available until the next update window.

5.1 Route establishment procedure

The route establishment procedure is initiated by the sink and executed to refresh the topological information every T_{up} seconds. The route establishment packet, RE_packet, sent by the sink (i.e., the gateway node) carries various pieces of information, including its ID, a sequence number, and a hop count field, which is increased every time each relay forwards this packet to its neighboring nodes. Each network node, upon receiving this packet from the sink, adds the sink node to its neighbor table, as well as the hop counter to the sink. Furthermore, the ID of the one-hop neighbor, which forwarded the packet, is added to this table. Upon receiving this packet for the first time, a node after having updated its table increases the hop counter and again relays the packet in broadcast. In the case of reception of a duplicate packet delivered by different one-hop neighbors, their identities are saved in the perspective of finding alternative routes to the sink. This may be needed in the case of bad channel conditions, temporary interference, and jamming or node failures. This allows the nodes to rapidly store alternative paths that can be made available for possible future use. Routing entries are periodically updated, either because they expire after a certain time T_o or because a new packet with a different sequence number can be received. It should be noted that to reduce the network signaling overhead, the exchange of information needed by the route establishment procedure can be implemented through piggybacking on data packets that are periodically sent to the sink[1].

5.2 Message forwarding procedure

In this section, we detail the procedure executed by underwater nodes to forward data packets into the network, toward the surface gateway node, while selecting the most appropriate forwarder[2].

Each node periodically sends in broadcast to its neighbors' information regarding its ID, the residual energy value, and the Q value parameter needed for the execution of the Q-learning protocol as described in the following sections.

Each packet is identified through an M_{ID} parameter set by the initial source that never changes during the delivery process, while all the other fields of the packet are updated by each forwarding node. Moreover, an indicator of the average residual energy of its one-hop neighbor node is also added, together with the result of the

[1] Specific definition of frequency of data exchanges and route signaling is out of the scope of this paper. However, we are assuming that the time scale of the exchange of data and signaling are similar. Note that in case these time scales are not comparable and, thus, more frequent exchange of signaling is needed as compared to data, the route establishment procedure will be explicitly invoked, and the corresponding system overhead will need to be accounted for.

[2] In our work we assume that the network is cooperative in the sense that all network nodes collaborate and cannot get away from forwarding packets if selected as next hop relay.

computation of the previous hop node and the selected next forwarder as estimated based on the Q value (see the following sections for details).

Before sending a data packet considering the previous successes and failures in the forwarding process and the topological information associated with the one-hop neighbors, each current forwarder will identify the best next relay and will add this information to the packet relayed.

Upon hearing this packet, the node that has been selected as the best next relay among the one-hop neighbors of the previous relay or source will extract some information from the sender, including the residual energy and the Q value associated with the learning procedure, as discussed in the following section. The selected next relay will update the associated entry in the local neighbor list owned by each node and will use it to estimate the Q value of its one-hop neighbors to proceed to the next selection step. All the other nodes, once not selected as the next hop relay, will discard the packet after having extracted from the packet the information needed to update the Q values and the status of the corresponding one-hop neighbors.

An implicit confirmation is implemented to identify whether or not a packet has been successfully delivered by analyzing the traffic issued by the selected next-hop relay node. In particular, a packet sent and heard by the previous forwarder will be considered an implicit acknowledgment. If, instead, the relayed packet is not heard, a maximum number of re-transmissions will be allowed before a packet is assumed to be lost.

In several scenarios, it is impossible or impractical to know the global network topology and its status. Therefore, our solution is based on the exchange of local information only. The downside of such a design decision is that a packet may arrive at a node that is unable to send the packet further toward the destination. This is a well-known void problem, and several solutions have been proposed in the literature, even in the specific context of underwater networks. In our solution, we propose to use the void-circumnavigating approach discussed in Coutinho et al. (2017), Mhemed et al. (2022).

6 Q-learning framework

This section details the machine-learning mathematical framework that we have used in this work to increase the system's robustness in defending against possible jamming attacks.

Reinforcement learning is a category of machine-learning approaches, where, by trial and error, through interactions with the dynamically changing *environment*, *actions* can be taken by agents to maximize a given *reward*. To describe the environment, a Markov decision process (MDP) can be employed. It consists of a set of states $\mathbf{S}$, a set of actions $\mathbf{A}$, a reward function $\mathbf{R}$, and a state transition matrix $\mathbf{P}$. Elements of the latter identify the probability of making a transition from state s_i to state s_j using action $a \in \mathbf{A}$. Elements in $\mathbf{R}$ are, instead, the related rewards for making a transition from state s_i to state s_j using action a. We observe that the actions taken not only have an impact on the sender's reward but also on all the future evolutions of the system.

In Cybenko et al. (1997), Q-learning, a variant of the RL mechanism based on the value of state–action pairs, has been presented. In this case, agents can learn to act optimally in Markov environments by assessing the consequences of their actions.

The policy π is a way of associating each state, $s \in S$, and possible action, $a \in A$, to the probability of executing the action when in state s. The value of taking an action a in state s under a policy π is defined as Q (s, a) and represents the expected return for taking an action a and using policy π.

Considering time evolution, the optimal policy at time t is denoted as $V^*(s_t)$ and is represented by the maximum overall possible actions $a \in \mathbf{A}$ of the value Q (s_t, a), i.e.,

$$V^*(s_t) = \max_a \{Q(s_t, a)\} \tag{1}$$

where

$$Q(s_t, a) = r_t + \gamma \sum_{s_{t+1} \in S} p^a_{s_t s_{t+1}} \max_a \{Q(s_{t+1}, a)\} \tag{2}$$

and at each iteration, Q (s_t, a) can be updated as

$$Q(s_t, a) \leftarrow (1 - \alpha) Q(s_t, a) + \alpha \left[r_t + \gamma \cdot \max_a \{Q(s_{t+1}, a)\} \right] \tag{3}$$

In the aforementioned Eqs. 2, 3, the term r_t is the reward. After executing an action a from state $\mathbf{s}$ at time t, r_t results in

$$r_t = \sum p^a_{s_t, s_{t+1}} R^a_{s_t, s_{t+1}} \tag{4}$$

where $p^a_{s_t, s_{t+1}} \in \mathbf{P}$ and $R^a_{s_t, s_{t+1}} \in \mathbf{R}$. In Eqs. 2, 3, it should be noted that α is the learning rate, which models the rate at which we estimate the Q-values, and $\gamma \in [0, 1]$ is a discount factor for future rewards, which considers how recent actions affect the current value compared to future ones. More specifically, γ specifies the importance given to future rewards. When γ is set to 0, the system considers the current reward only and tries to maximize the reward from a short-term perspective. When γ is set to 1, the system will try to achieve a long-term relevant reward. A balance between these two opposite trends implies that a good choice of γ is in the range $[0.5, 0.99]$.

The elements of the reward function $\mathbf{R}$ will be calculated as discussed in the following sections; the computation of the transition matrix $\mathbf{P}$ will instead be addressed in Section 7.

6.1 Calculation of the reward function

Despite the unreliability and time variability of underwater link conditions, as well as needing to maximize the network lifetime, a reward function can be used to stimulate data delivery among nodes. This function appropriately measures both the average energy consumption across all one-hop neighbor nodes and the energy consumption at each node. This facilitates the provision of fair energy consumption distribution in the network, thus preserving network connectivity. In more detail, the initial energy available at all network nodes can be spent to send and forward any type of packet, both data and management packets. Accordingly, residual energy keeps decreasing at each node n every time it is used as a relay. Two parameters can be employed to account for this energy consumption, namely, $c(n)$ and $d(n)$, which are included in the reward function. In particular,

- $c(n)$ is the cost function associated to the residual energy at a node n, i.e., $c(n) = 1 - \frac{E_{res}(n)}{E_{init}(n)}$
- $d(n)$ is the reward term due to energy consumption distribution across a whole group of one-hop sensor nodes, i.e., $d(n) = \frac{E_{res}(n) - E_{avg}(n)}{E_{init}(n)}$.

By setting all nodes as having the same initial energy, it follows $c(n) \in [0,1]$ and $d(n) \in [-1,1]$.

A function $R^{a_m}_{s_n,s_m}$, that represents the reward can be introduced to measure the costs in the one-hop transmission from a node n to a neighbor m. In this work, we identify the state s_n with the condition when a packet is held by node n, and we identify action a_m as the action to forward a packet to node m. Specifically, in the case of a successful forwarding, the reward function is

$$R^{a_m}_{s_n,s_m} = -g - \alpha_1[c(n) + c(m)] + \alpha_2[d(n) + d(m)]. \qquad (5)$$

Instead, in the case of transmission failure from node n to m, the reward function is

$$R^{a_m}_{s_n,s_n} = -g - \beta_1 c(n) + \beta_2 d(n). \qquad (6)$$

In the aforementioned equations, g identifies a constant cost incurred when a node forwards a packet, independently of the outcome of the packet transmission. Parameters α_1 and α_2 measure the cost function terms, thus figuring out a trade-off between a reduction in the number of hops to the destination and the selection of nodes with higher residual energy. Analogous considerations apply to parameters β_1 and β_2.

Equations 5, 6 can be used in Eq. 4 to estimate r_t. We observe that $R^{a}_{s_t,s_{t+1}}$ is always negative and, thus, all the $Q(s,a)$ values for the non-destination nodes are negative. In our scenario, we assume that the sink is the only final destination for all transmissions. The Q value of the destination node will be set to 0 because it has to be the largest among all the Q values. Considering the model described so far, we associate each packet forwarding attempt with energy and channel bandwidth consumption, as well as a resulting delivery delay. By appropriately choosing the measures assigned to the cost terms, it is possible to balance opposite targets: minimize the delay and the corresponding number of hops and balance the network energy consumption. This could sometimes increase the hop counter, but with a gain in terms of network lifetime, since nodes are alive for a longer time, and the network gets more chances to remain connected.

In the next section, we will provide underwater channel modeling through the calculation of the values of the state transition matrix **P** needed in the Q-learning model.

7 Markov channel model

In underwater scenarios, upon propagating the acoustic signal from a transmitter to a receiver, numerous replicas are found due to seabed and sea surface reflections, resulting in serious multipath fading. More specifically, signals traveling on different paths can result in both in-phase and out-of-phase components. Moreover, vertical temperature and pressure variations, as well as the salinity of water and pH, impact propagation losses, which are tightly frequency-dependent because of the frequency selectivity of the channel.

Ambient noise sources, e.g., due to environmental features in the proximity of the surface (e.g., wind and rain) or exogenous sources, e.g., ship activity of cargoes and thermal noise or turbulence, play a crucial role in path attenuation and losses. In the recent past, multiple underwater channel models have been presented, e.g., B. Tomasi et al. (2010), Casari and L. Finesso and G. Zappa and K. McCoy and M. Zorzi (2010), and Pignieri et al. (2008)[3]. Among these, for the sake of simplicity, we refer to Pignieri et al. (2008), where a discrete-time Markov chain (DTMC) model of an underwater acoustic channel was elaborated from real traces collected in the Mediterranean Sea. More specifically, a DTMC is a simple discrete-time memoryless stochastic process, where the current state of the channel only depends on the previous one and not on the history of the previous process states.

A transition probability matrix can be obtained, where the generic element $p_{i,j}$ represents the probability that the process is in state j at time t, given that at time $t-1$, it was in state i.

In the transition probability matrix, the sum of the elements of each row is always equal to 1, i.e.,

$$\sum_{j=0}^{n} p_{i,j} = 1 \quad i = 0, 1, 2 \ldots n \qquad (7)$$

Transition probabilities are used to characterize the efficiency of the links. To describe the underwater channel state by way of a DTMC, the stationarity needs to be proved. Accordingly, in Pignieri et al. (2008), a Reverse Arrangement Test (Bendat and Piersol, 1986) was used to calculate the distribution of the sojourn time, proving that the sojourn time is exponentially distributed. Then, through the Kolmogorov–Smirnov test (Montgomery, 2003), DTMC models of order K were derived with the aim of capturing different degradation levels in the underwater channel.

As employed in Pignieri et al. (2008), we consider a DTMC channel model of order K, where multiple channel states can be used. To reach a trade-off between complexity and fidelity in channel description, in our previous paper (Shivani et al., 2020), we compared different Markov channel models with various numbers of states and consequent increasing size and complexity. In particular, we considered a Markov model with several states ranging from 2 to 8. What emerged from this comparison is that, despite the complexity coming from the use of the Markov model with multiple states, the 2-state channel model is reliable and complete and can realistically describe the underwater channel in a chosen setting.

Accordingly, in our work, only *good* and *bad* channel states are considered, and the transition probability matrix [for a setting with a Doppler frequency of 3 Hz, a BER of approximately 6%, and assuming OFDM multiplexing (Pignieri et al., 2008)] can, thus, be given as

$$\mathbf{P} = \begin{pmatrix} 0.8718 & 0.1282 \\ 0.4659 & 0.5341 \end{pmatrix} \qquad (8)$$

[3] Different underwater channel models could be used to more or less accurately describe the channel. An alternative choice would only impact on the numerical values of the p^{a}_{s_{t},s_{t+1}} transition probabilities but without still affecting the validity of the design framework.

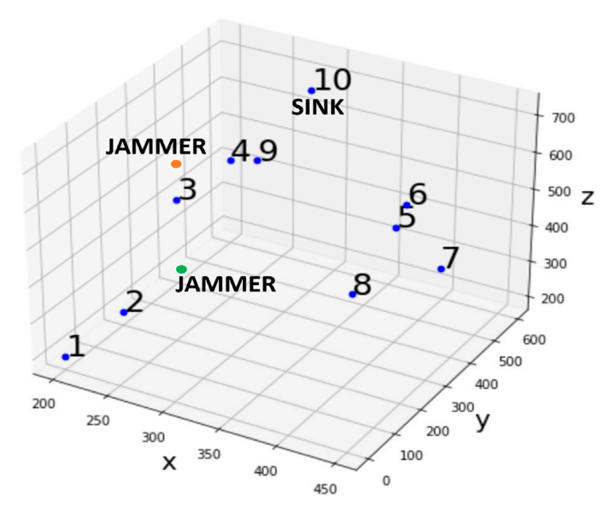

FIGURE 3
Jammer positions in the network (in the case of 10 nodes).

It should be noted that if the channel conditions on all possible outgoing links for a node are bad, it defers transmission until channel conditions improve and the channel can be identified as being in a good state.

Consequently, in the following numerical analysis, we restrict our modeling of the underwater channel to the specific case of a 2-state Markov model.

8 Performance analysis

To assess the effectiveness of our proposed protocol, in this section, we analyze the achievable system performance.

In our work, we considered a network of 10 underwater nodes deployed at different depths. All the nodes are assumed to be in connection with each other if the distance between them is less than 300 m, provided that the channel is in a good state and they are not impacted by any ongoing jamming action. In our scenario, we have assumed that the sink node is placed on the top of the network, i.e., closer to the surface, and acts as a gateway device to which all the transmitted packets will be delivered. The sink (which acts as a data receiver only) will, for instance, represent the edge of the underwater network and could be connected in some way to the remote terrestrial network. Packets are continuously generated at random by any network node. Furthermore, the packet generation time is assumed to be instantaneous, while packet length is fixed and equal to a one-time unit. The jamming action instead is variable and can last for multiple time units. Simulations were conducted considering that the jammer jams according to a duty cycle equal to either 50% or 90%. This means that provided that the jammer is located in the proximity of a transmitting node, both in the case of our approach and in the RLOR, the jammer can impair the transmission for a given percentage of the packet transmission window (i.e., 50% or 90%, respectively).

Simulations carried out consider the transmission of up to 1,000 packets in a simulation time of 3,000 time units. Each packet transmission is assumed to be executed in one time unit. It should be noted that each packet transmission corresponds to 100 training epochs/episodes for the Q-learning algorithm and implies a consequent update of the Q matrix. The mean reward is calculated for each episode and taken into account for every 100 packet transmissions to investigate the performance of the Q-learning approach. As it is well-known that the performance of the agent will be poor in the initial stage and will be improved with training, the mean episode reward will be lower during the initial stage of learning and will increase after the transmission of a certain number of packets.

We have assumed that for each network node, an initial energy of 1 KJ is available and that for each packet transmission, 1 J of energy is spent. Concerning the hyperparameters of the Q-learning model, we have assumed that the learning rate used in Eq. 3 is $\alpha = 0.5$, while the parameter g, which identifies a constant cost incurred when a node forwards a packet, independently of the outcome of the packet transmission, is equal to 1. Concerning the discount factor of the rewards, which considers how recent actions affect the current value compared to future ones, it is chosen as $\gamma = 0.5$. Furthermore, the parameters used in the calculation of the reward function were chosen as $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$.

As discussed previously, to realistically characterize the underwater channel, while also reducing the complexity of the modeling, a 2-state Markov model is employed, namely, the one in Eq. 8.

To estimate the possible impact of jamming performed by misbehaving nodes, we also consider a jammer node located at two different positions inside the network, as shown in Figure 3. For more in-depth information, we investigate the case when the jammer is located at the bottom of the network and another case where the jammer is located in a more central position. Both situations are depicted in Figure 3.

We investigate how different jamming positions and different duty cycle choices impact the overall achievable data delivery performance upon employing our Q-learning routing mechanism.

In the scenarios shown in Figures 2, 3, nodes i and j are in each other's coverage range, if their distance is less than 300 m. Link conditions vary with time, and, thus, if the channel state is good in the link between the two nodes and there is no ongoing jamming action, the transmission will be successful; otherwise, it will not be successful.

The channel state is updated before each packet transmission. The probability that the channel state is in a specific state i, Π_i can be calculated by considering the well-known Markov conditions, i.e.,

$$\begin{cases} \mathbf{\Pi} \cdot \mathbf{P} = \mathbf{\Pi} \\ \sum_i \Pi_i = 1 \end{cases} \tag{9}$$

All the simulation experiments were performed using Python in the Google Colab environment, as it provides all the relevant tools for supporting machine-learning experiments and provides the transparent use of GPU resources in the case of intensive computations. However, it is also possible to run our proposed machine-learning approach on devices that support Python programming language, although limited in terms of power and computational resources, such as in the case of microprocessors, e.g., Raspberry Pi.

For the sake of comparison, in this section, the effectiveness of the proposed protocol solution to counteract jamming attacks has

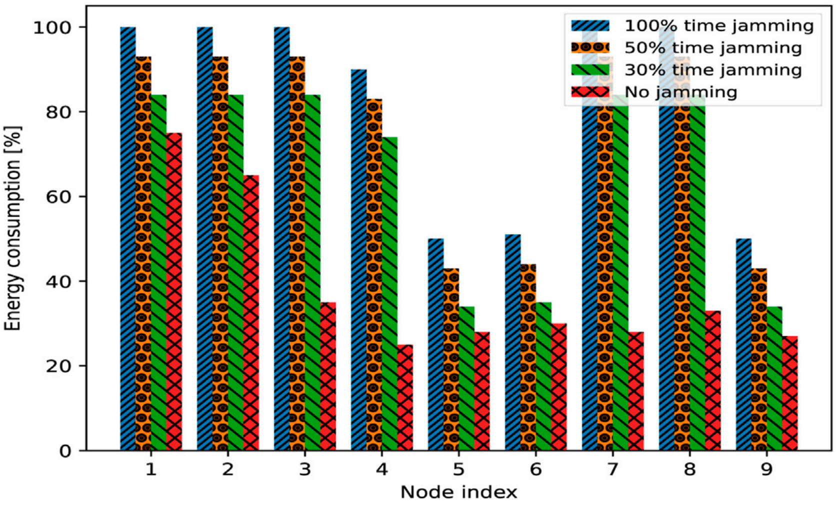

FIGURE 4
Comparison of energy consumption for each network node in the 10-node topology in the presence and absence of the jammer, using the approach proposed in this paper (jamming executed with a duty cycle of 90%).

been investigated and compared with a benchmark approach, i.e., the well-known RLOR algorithm proposed in Zhang et al. (2021), which is the most relevant to our work.

In particular, the RLOR protocol Zhang et al. (2021), similar to our approach, is a reinforcement learning-based routing solution developed for underwater acoustic sensor networks. The relay nodes are selected based on their status and on some topological information. In Zhang et al. (2021), the focus is on avoiding incurring void nodes, which do not allow for finally delivering the intended data to the receiver sink node.

The RLOR algorithm combines learning and opportunistic routing to identify the relay nodes for forwarding data. First, a set of possible nodes is selected based on the residual energy, node depth, and neighboring node count. Second, the best relay nodes are selected based on a Q-learning approach, where the reward function, unlike ours, considers that, at time t, if the current packet is held by node n_i, the reward function will depend on the number of neighbors above node n_i, their residual energy, the depth difference, and the probability of successful packet transmission. The latter depends on the signal-to-noise ratio and, thus, on the distance between the intermediate source and destination, as well as on the characteristics of the physical level modulation methodology being used. The RLOR also includes an opportunistic methodology, in which a node first forwards the packets to each node selected based on the algorithm, and each candidate node holds a copy of the packets. A timer is set for each node to hold the copies. Once the timer is over, that particular node will be selected as the relay node. The timer is set based on several parameters such as the maximum communication range of the node and the propagation speed of sound in water.

The RLOR was compared in Zhang et al. (2021) with other reinforcement learning-based protocols, and simulation results show that the RLOR outperforms the other protocols in terms of end-to-end delay, energy efficiency, and reliability, resulting in better performance. This motivates our choice to compare the RLOR with our proposed approach.

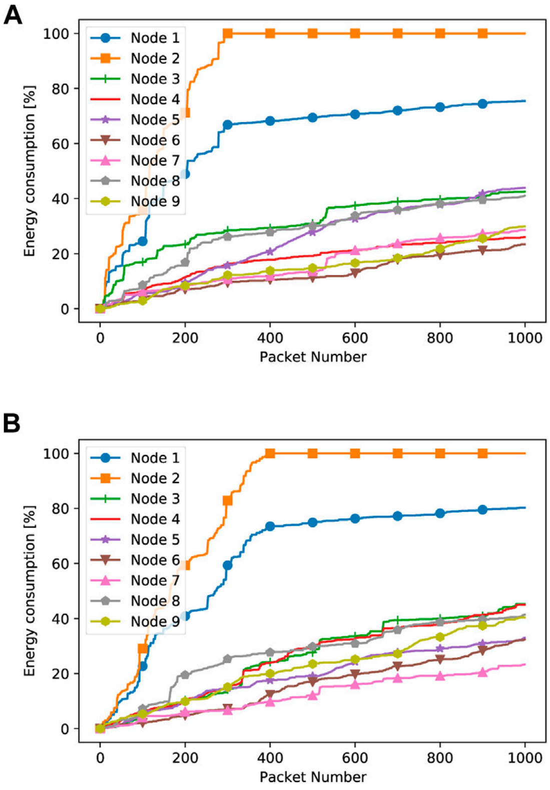

FIGURE 5
Energy consumption at network nodes when the Jammer is located in the bottom of the network with Duty Cycle 90% **(A)** and Duty Cycle 50% **(B)** (our approach).

8.1 Energy efficiency performance

In this subsection, we evaluate and compare the energy consumption in both our approach and the RLOR. In our

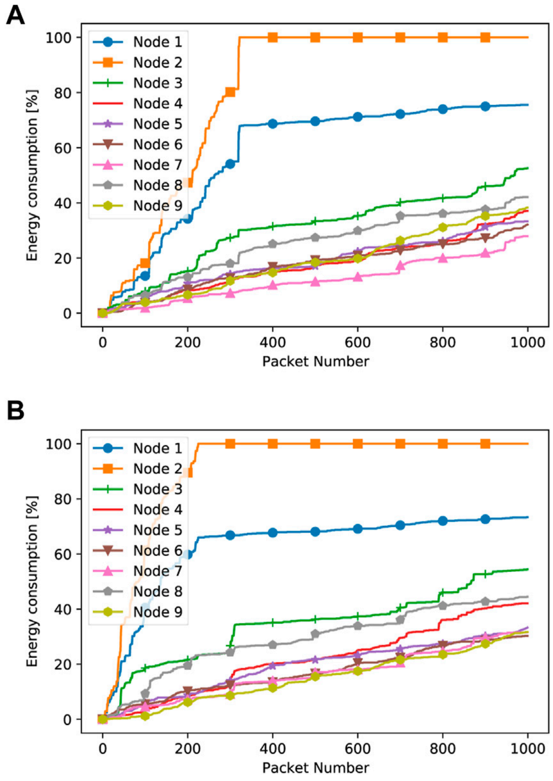

FIGURE 6
Energy consumption at network nodes when the jammer is located in the center of the network with Duty Cycle 90% **(A)** and Duty Cycle 50% **(B)** (our approach).

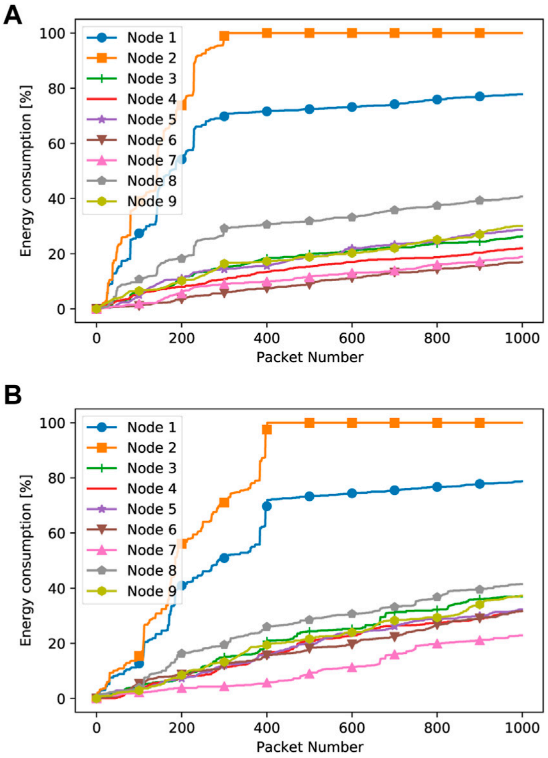

FIGURE 7
RLOR energy consumption at network nodes with the jammer at the bottom of the network with Duty Cycle 90% **(A)** and Duty Cycle 50% **(B)**.

experiments, we executed 10 tests, in which the values were averaged and reported in the following figures. Furthermore, it should be noted that based on the *t*-Student distribution properties (Ifram, 1970), our results provide a confidence interval of 90%.

As shown in Figure 4, we preliminarily compare the energy consumption achieved using our approach in the case that the jammer is active and when the jammer is not active. Concerning the case of the ongoing jamming action (for all cases, a duty cycle of 90% is assumed), we report, in the same figure, the case of a jammer that is active all the time, the case of a jammer that acts only for 50% of the time and then switches off, and the case of a jammer that acts only for 30% of the time and then switches off. As is evident in the figure, the energy consumption is significantly higher when the jammer is active compared to when it is not. Furthermore, a longer jamming action causes more serious energy depletion.

Figures 5, 6 show that the energy consumption at network nodes in our approach is a function of the number of packets transmitted inside the network and the duty cycle. It should be noted that these figures have been obtained for both the jammer positions (bottom and central jammer). It should also be noted that when a jamming action is ongoing, in the case of our approach, re-transmissions are avoided. Indeed, an alternative path is identified, even if this implies a change in the direction of propagation with a deviation from the shortest path direction.

By comparing the different plots, we observe that the energy consumption is higher for nodes 1 and 2, irrespective of the jammer positions. The reason is that nodes 1 and 2 are far from the other nodes and they undergo many unsuccessful transmission events due to the impact of the jammer's presence on the other nodes. In fact, in both cases, the jammer is located in the bottom part of the network, and therefore closer to nodes 1 and 2, or the center of the network; transmissions of 1 and 2 will either be directly jammed by the jammer with high probability or indirectly unsuccessful when nodes 3, 4, and 9 are attacked by the central jammer and not able to forward the packets coming from 1 and 2 themselves. The residual energy in node 1 is completely drained out after 300–400 packet transmissions, depending on the jammer setting. Node 3 also suffers from higher energy consumption than other nodes due to its closer proximity to the jammer in both settings. At the remaining nodes, the energy consumption is instead similar, apart from the fact that a larger duty cycle at the jammer implies slightly more energy consumption at the network nodes. It should be noted that by moving to the scenario where the jammer is in a central position in the network, the energy consumption of network nodes slightly moves right. In the case of our approach, even in the situation where an aggressive jamming action is ongoing (i.e., longer duty cycle), our protocol reacts quite efficiently, and, thus, the energy consumption remains unchanged.

Similarly, in Figures 7, 8, we report the energy consumption at network nodes when the RLOR protocol is used for different

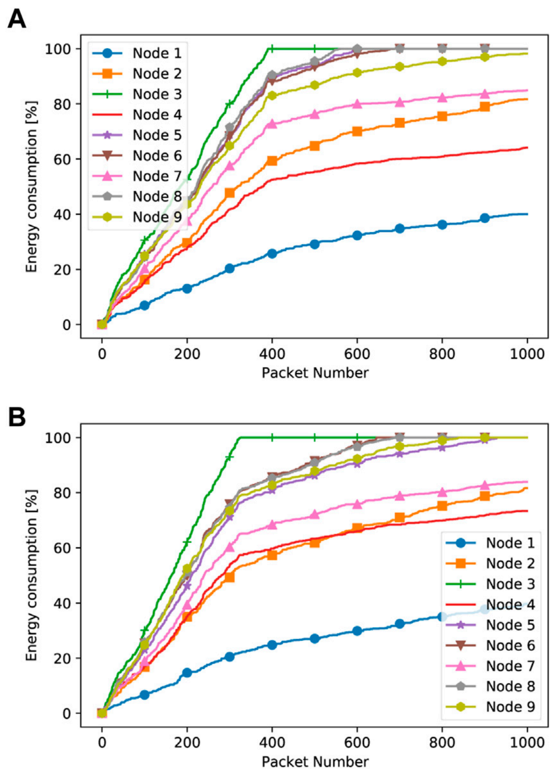

FIGURE 8
RLOR energy consumption at network nodes with the jammer in the center of the network with Duty Cycle 90% **(A)** and Duty Cycle 50% **(B)**.

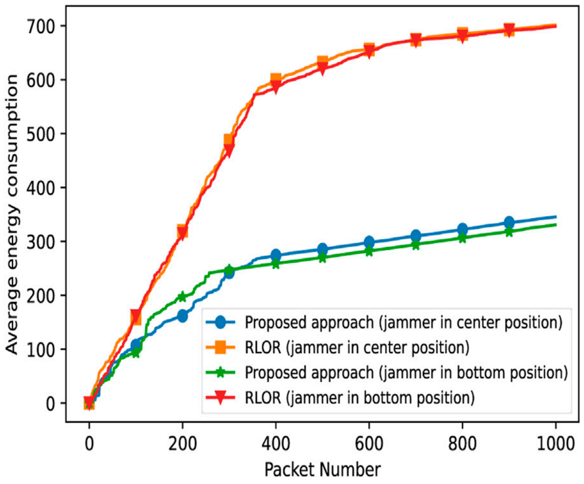

FIGURE 9
Comparison between our approach and the RLOR in terms of the average energy consumption, with the jammer in the center and at the bottom of the network and a duty cycle of 90%.

jammer positions. In the case of the central jammer, we note that nodes 1 and 2 consume less energy, whereas node 3 consumes more energy. Furthermore, in general, more energy consumption

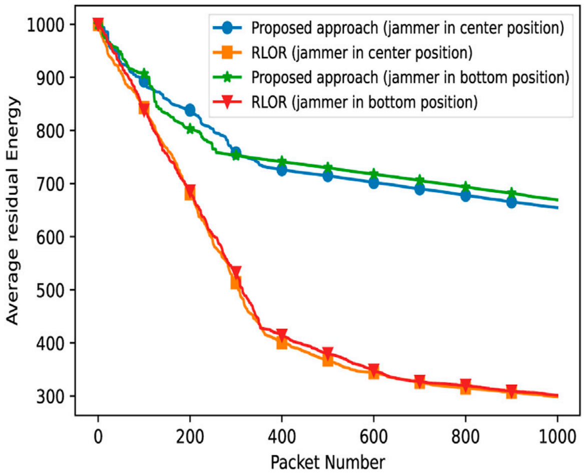

FIGURE 10
Comparison between our approach and the RLOR in terms of average residual energy, with the jammer in the center and at the bottom of the network and a duty cycle of 90%.

(about a 100% increase) is found than that in our case. The reason is that the proposed methodology, compared to state-of-the-art approaches such as the RLOR, tries to achieve a fair energy consumption balance among network nodes, with the aim of increasing the network lifetime. Instead, this is not the main target of the RLOR. It should also be noted that the impact of an increase in the jammer duty cycle is more relevant for the RLOR compared to our approach, especially in terms of the load on nodes 2 and 4.

In Figures 9, 10, we compare the average energy consumption and residual energy for both our approach and the RLOR, for different jammer positions. We observe that the average energy consumption in the RLOR case increases at a faster rate than in our approach, irrespective of jammer positions.

In Figure 9, we note that the average energy consumption changes its slope and tends to stabilize after the transmission of approximately 400 packets. This is a clue to the convergence of our proposed Q-learning algorithm after the transmission of a given number of packets (i.e., 400 packets in this case) as a consequence of the improvement in the mean episode reward after the initial learning phase. This is a common behavior that can be somehow observed in Figure 5 to Figure 8, although sometimes for different values of the number of packets.

Similarly, the residual energy reduces faster in the RLOR than in our approach. This indicates that our proposed methodology is efficient (efficiency is almost doubled) in terms of energy consumption, independent of the jammer position and duty cycle.

Finally, for the sake of the completeness of our work, we also consider multiple topologies to study the energy consumption associated with our proposed approach. More specifically, we consider topologies consisting of 20 and 30 nodes, in addition to the basic topology with 10 nodes. We illustrate in Figure 11 the average energy consumption, average residual energy, and Jain's fairness index for the three considered topologies. The fairness index has been calculated using (Vandalore et al., 2000):

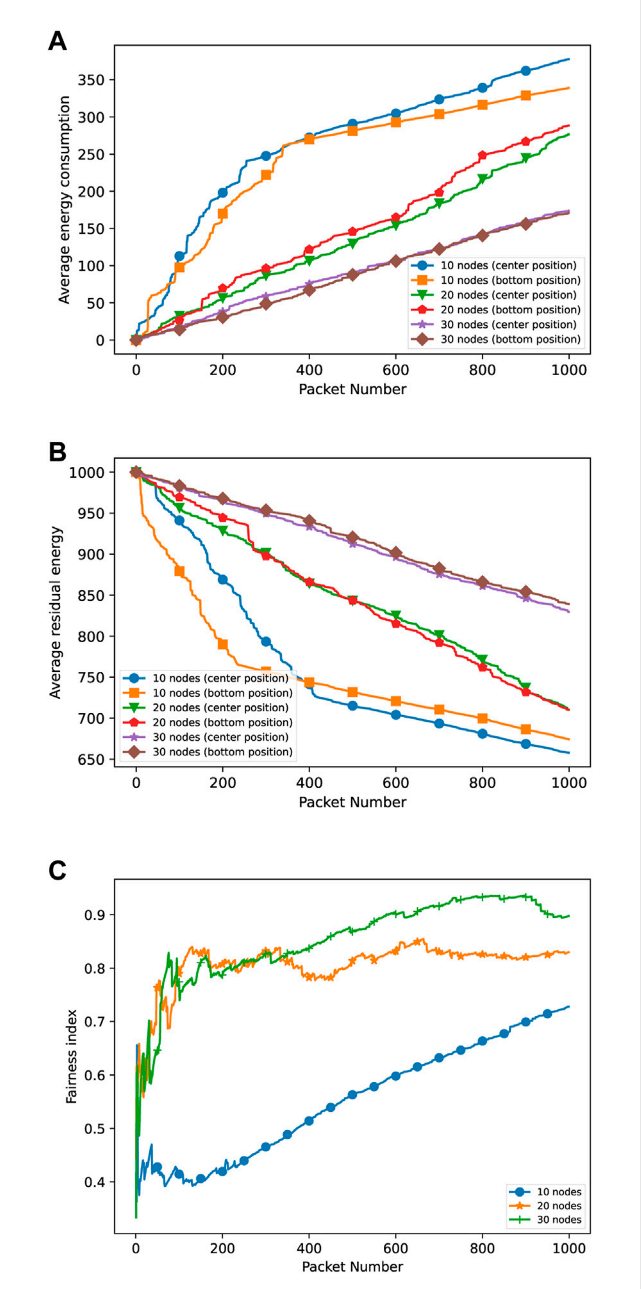

FIGURE 11
Average energy consumption **(A)** and average residual energy **(B)** in our approach for multiple topologies (10, 20, and 30 nodes) and different jammer positions. Fairness index for multiple topologies **(C)**.

$$\mathcal{F}(x_1,\dots x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n} x_i^2}, \tag{10}$$

where x_i denotes the average energy consumption at the ith node, $i \in 1\dots n$.

As shown, the energy consumption is higher when a topology with a lower number of nodes is considered. On the other hand, with larger topologies, the average energy consumption improves. Consequently, as expected, the fairness is higher for larger topologies. Furthermore, we noticed that the Fairness index is poor at the beginning and gets better after some packet transmissions. The reason for this behavior is related to the proposed routing mechanism. In fact, depending on the ongoing jamming action, as well as on the availability of multiple paths, energy consumption results from the balancing of residual energy considerations at different nodes. In the case of a smaller topology (i.e., 10 nodes only), the routing of packets during an ongoing jamming action can be problematic since fewer nodes are available for forwarding packets toward the final destination. This also implies higher consumed energy at all network nodes as directly or indirectly suffering the jamming action. Instead, in the case of larger topologies, e.g., 20 or 30 nodes, the machine-learning-based routing scheme will show its potential in terms of exploitation of multiple alternative paths with a consequent decrease in average energy consumption at network nodes and better fairness. It should be noted that this robustness is observed, independently of the position where the jammer device is located. Furthermore, it should be noted that when a limited number of packets is considered (less than 300), in the case of 20 or 30 nodes, similar behavior is exhibited since few packet transmissions are not sufficient to exhaust node batteries, and thus, the choice of relayer is less critical. On the other hand, upon increasing the number of packets, the solution can adjust and respond well to the presence of multiple nodes and, thus,

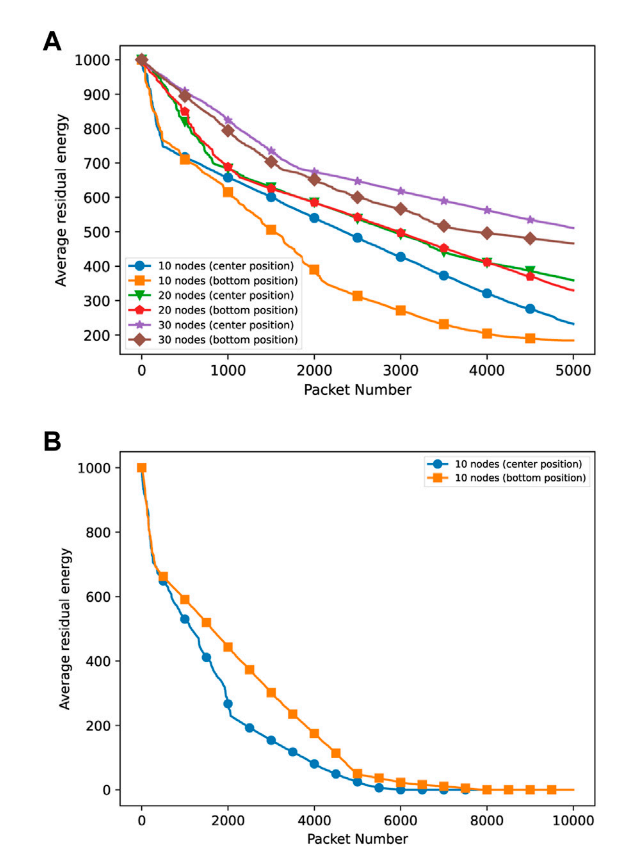

FIGURE 12
Comparison between the average residual energy of the network with 10, 20 and 30 nodes with Duty Cycle 90% for 5000 packets **(A)** and Average residual energy of the network with 10 nodes and Duty Cycle 90% for 10000 packets **(B)**.

TABLE 1 Latency and packet delivery rate for our proposed approach and the RLOR.

Approach	$\alpha 1$	$\alpha 2$	Latency (Center)	Delivery rate (Center)	Latency (Bottom)	Delivery rate (Bottom)
Duty cycle of 90%; number of nodes: 10						
Our approach	0.5	1	2.834	0.6	2.981	0.6
	0.5	0.75	3.08	0.796	3.594	0.820
	0.5	0.5	3.014	0.646	3.539	0.839
	0.5	0.25	3.062	0.671	2.869	0.591
	0.5	0.05	2.829	0.579	3.13	0.806
RLOR	-	-	4.67	0.432	4.822	0.430
Duty cycle of 50%; number of nodes: 10						
Our approach	0.5	0.5	2.883	0.605	2.883	0.727
RLOR	-	-	4.706	0.431	4.76	0.415
Duty cycle of 90%; number of nodes: 20						
Our approach	0.5	0.5	4.888	0.972	4.994	0.99
RLOR	-	-	12.797	0.266	12.687	0.290
Duty cycle of 50%; number of nodes: 20						
Our approach	0.5	0.5	4.818	0.961	4.967	1.0
RLOR	-	-	12.848	0.295	12.737	0.303
Duty cycle of 90%; number of nodes: 30						
Our approach	0.5	0.5	3.686	1.0	4.379	1.0
RLOR	-	-	16.174	0.420	16.202	0.444
Duty cycle of 50%; number of nodes: 30						
Our approach	0.5	0.5	4.242	1.0	4.06	1.0
RLOR	-	-	16.33	0.410	16.285	0.422

multiple possible relayers. Dual considerations apply in the case of the average residual energy.

In Figure 12, we report the average residual energy with a duty cycle of 90% and different topologies for the jammer and a different number of network nodes. It should be noted that upon increasing the number of packets, the average residual energy keeps decreasing. However, it should also be noted that, as discussed previously, in the case of a larger number of nodes, the average residual energy remains higher. For the most critical scenario of 10 nodes, we also explored the network lifetime evolution, showing that for the case of the bottom jammer, the average residual energy is larger than in the case of the central jammer (see Figure 12B).

8.2 Average latency and delivery rate

In this section, we report results about the average latency and the packet delivery rate obtained using both our approach and the RLOR.

In Table 1, we show the average latency and packet delivery rate in our approach for all the possible jammer positions and different values of α_1 and α_2. In Table 1, we also report the average latency and packet delivery rate for the RLOR case. It should be noted that the RLOR does not consider parameters α_1 and α_2. As is evident in this table, the average latency is significantly larger in the RLOR case than in our approach. Furthermore, the packet delivery rate achieved using our solution is larger, always achieving values larger than 58%–60%, independently of the jammer position. The higher rate of energy depletion in the RLOR case causes the inactivation of certain nodes in the network, resulting in packet drops and a lower delivery rate.

Furthermore, it is interesting to note that, in our approach, the delivery rate is quite stable and the nodes are not significantly impacted by jammer position and/or α values. This assesses the relevant protocol reliability and robustness provided by our proposed solution.

For the sake of completeness, in the same table, we have also considered larger topologies, in which there are 20 and 30 nodes. By observing that the algorithm is quite stable and the α_1 parameter does not play a relevant role, we have chosen to show the latency and delivery rate results in the case of $\alpha_1 = \alpha_2 = 0.5$, for 10, 20, and 30 nodes and duty cycle values equal to 50% and 90%. It should be noted that in the case of a larger number of nodes, i.e., 20 or

30 nodes, our approach always outperforms the RLOR scheme, although, as expected, in general, the latency is a bit higher than that in the case of 10 nodes because of the longer detour resulting from the use of multiple hops. However, it should be noted that the topology with 30 nodes performs better than that with 20 nodes in terms of latency because the detour can be partially reduced thanks to the denser topology. The stability of the approach is also guaranteed, despite the different duty cycles. Concerning the delivery rate, the values obtained in our approach are significantly higher than those of the RLOR, and a delivery rate of 100% is achieved in the case of 30 nodes. This is due to the dense network topology, which efficiently counteracts the jamming action. In the case of the RLOR for 20 and 30 nodes, the latency is much larger than in the case of 10 nodes, which reflects the fact that the RLOR does not cope efficiently with jamming. Our approach, instead, works effectively even in scenarios with a larger number of nodes, providing a higher delivery rate and lower latency.

9 Conclusion

The use of national border surveillance to identify possible unauthorized intrusions has, in the recent past, been proven critical in the context of terrestrial or marine attacks. In this paper, we have introduced a machine-learning-based routing methodology to cope with jamming attacks aimed at denial of service in underwater scenarios. The proposed methodology, relying on the use of Q-learning to choose which neighbor node to relay data to, under an ongoing jamming attack, has relevant applications in security-preserving contexts. The proposed methodology was shown to increase reliability in data delivery while preserving robustness and low complexity. Extensive simulation and performance analysis, also comparing the proposed approach to other state-of-the-art solutions, and considering different positions for the jammer and a duty cycle mechanism, showed that the proposed solution is effective and efficient in reducing useless energy consumption, while at the same time, balancing energy expenditure across network nodes and preserving the data delivery rate and limited latency.

Author's Note

A previous version of this paper appeared in Shivani et al. (2020).

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

JM, AP, and AS contributed to the definition of the protocol details and the implementation of the code needed for the purpose of performance evaluation and comparison. NP and LG helped in the development of the idea and framework, contributed to the protocol definition, and helped in the characterization of the experimental setting and performance metrics. All authors contributed to the article and approved the submitted version.

Funding

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, the partnership on "Telecommunications of the Future" (PE0000001—program "RESTART"), and by UNICT Projects "PIACERI OMNIA and CHANCE PIACERI." AS carried the work out as a visiting student at the University of Catania, Italy.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ahmad, I., Rahman, T., Zeb, A., Khan, I., Ullah, I., Hamam, H., et al. (2021). Analysis of security attacks and taxonomy in underwater wireless sensor networks. *Wirel. Commun. Mob. Comput.* 2021, 1–15. doi:10.1155/2021/1444024

Akyildiz, I. F., Pompili, D., and Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Netw.* 3, 257–279. doi:10.1016/j.adhoc.2005.01.004

Aman, W., Al-Kuwari, S., Muzzammil, M., Rahman, M. M. U., and Kumar, A. (2023). Security of underwater and air-water wireless communication: State-of-the-art, challenges and outlook. *Ad Hoc Netw.* 142, 103114. doi:10.1016/j.adhoc.2023.103114

Bagali, S., and Sundaraguru, R. (2019). Efficient channel access model for detecting reactive jamming for underwater wireless sensor network. International Conference on Wireless Communications Signal Processing and Networking (WiSPNET). 21-23 March 2019, India, IEEE, 196. doi:10.1109/WiSPNET45539.2019.9032861

Bendat, J., and Piersol, A. (1986). *Random data: Analysis and measurement procedures.* (NY,USA: Wiley.

Caiti, A., Calabrò, V., Dini, G., Lo Duca, A., and Munafò, A. (2012). Secure cooperation of autonomous mobile sensors using an underwater acoustic network. *Sensors* 12, 1967–1989. doi:10.3390/s120201967

Cong, Y., Yang, G., Wei, Z., and Zhou, W. (2010). Security in underwater sensor network. *Int. Conf. Commun. Mob. Comput.* 1, 162–168.

Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., and Loureiro, A. A. F. (2017). Performance modeling and analysis of void-handling methodologies in underwater wireless sensor networks. *Comput. Netw.* 126, 1–14. doi:10.1016/j.comnet.2017.06.027

Cybenko, G., Gray, R., and Moizumi, K. (1997). *Q-learning: A tutorial and extensions.* Boston, MA: Springer US. 24–33. doi:10.1007/978-1-4615-6099-9_3

Di Valerio, V., Presti, F. L., Petrioli, C., Picari, L., Spaccini, D., and Basagni, S. (2019). Carma: Channel-aware reinforcement learning-based multi-path adaptive routing for underwater wireless sensor networks. *IEEE J. Sel. Areas Commun.* 37, 2634–2647. doi:10.1109/jsac.2019.2933968

Dini, G., and Lo Duca, A. (2012). A secure communication suite for underwater acoustic sensor networks. *Sensors* 12, 15133–15158. doi:10.3390/s121115133

Erpek, T., Sagduyu, Y. E., and Shi, Y. (2019). Deep learning for launching and mitigating wireless jamming attacks. *IEEE Trans. Cognitive Commun. Netw.* 5, 2–14. doi:10.1109/TCCN.2018.2884910

Evologics (2023). Evologics acoustic modems. Available at: https://evologics.de/acoustic-modems.

Goetz, M., Azad, S., Casari, P., Nissen, I., and Zorzi, M. (2011). "Jamming-resistant multi-path routing for reliable intruder detection in underwater networks," in *Proceedings of the 6th international workshop on underwater networks* (New York, NY, USA: Association for Computing Machinery). WUWNet '11. doi:10.1145/2076569.2076579

Hu, T., and Fei, Y. (2010). Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. *IEEE Trans. Mob. Comput.* 9, 796–809. doi:10.1109/TMC.2010.28

Ifram, A. F. (1970). On the characteristic function of f and t distributions. *Indian J. Statistics, Ser. A* 32.

Kalita, S., and Sahu, P. (2015). "An anti-jamming underwater communication transceiver model using uncoordinated direct sequence spread spectrum technique," in *2015 2nd international conference on electronics and communication systems (ICECS)* (IEEE), 972. –976.

Kulhandjian, H., Melodia, T., and Koutsonikolas, D. (2014). "Securing underwater acoustic communications through analog network coding," in 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 266–274. doi:10.1109/SAHCN.2014.6990362

Liu, Y., Jing, J., and Yang, J. (2008). Secure underwater acoustic communication based on a robust key generation scheme. 9th International Conference on Signal Processing 24th December 2023, China, IEEE, 1838–1841.

Mhemed, R., Phillips, W., Comeau, F., and Aslam, N. (2022). Void avoiding opportunistic routing protocols for underwater wireless sensor networks: a survey. *Sensors* 22 (23), 9525. doi:10.3390/s22239525

Mohsan, S. A. H., Li, Y., Sadiq, M., Liang, J., and Khan, M. A. (2023). Recent advances, future trends, applications and challenges of internet of underwater things (iout): A comprehensive review. *J. Mar. Sci. Eng.* 11, 124. doi:10.3390/jmse11010124

Montgomery, C. (2003). *Applied statistics and probability for engineers.* NY,USA: Wiley.

Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C., and Pantziou, G. (2009). A survey on jamming attacks and countermeasures in wsns. *IEEE Commun. Surv. Tutorials* 11, 42–56. doi:10.1109/SURV.2009.090404

Pignieri, F., De Rango, F., Veltri, F., and Marano, S. (2008). "Markovian approach to model underwater acoustic channel: Techniques comparison," in MILCOM 2008 - 2008 IEEE Military Communications Conference, China, 16-19 Nov. 2008 (IEEE). 1–7. doi:10.1109/MILCOM.2008.4753161

Samir, M., Kowalski, M., Zhou, S., and Shi, Z. (2014). An experimental study of effective jamming in underwater acoustic links. IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems 28-30 October 2014, USA, (IEEE, 737–742).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Netw.* 61, 85–117. doi:10.1016/j.neunet.2014.09.003

Shi, E., and Perrig, A. (2004). Designing secure sensor networks. *IEEE Wirel. Commun.* 11, 38–43. doi:10.1109/MWC.2004.1368895

Shi, Y., and Sagduyu, Y. E. (2017). "Evasion and causative attacks with adversarial deep learning," in MILCOM 2017 - 2017 IEEE Military Communications Conference, USA, 16-19 Nov. 2008 (MILCOM), 243. –248. doi:10.1109/MILCOM.2017.8170807

Shivani, S., Surudhi, A., Prabagarane, N., and Galluccio, L. (2020). "A Q-learning approach for the support of reliable transmission in the internet of underwater things," in 2020 16th International Conference on Wireless and Mobile Computing, Germany, 12-14 Oct. 2020 (Networking and Communications (WiMob). 1–6. doi:10.1109/WiMob50308.2020.9253368

Signori, A., Chiariotti, F., Campagnaro, F., Petroccia, R., Pelekanakis, K., Paglierani, P., et al. (2022). A geometry-based game theoretical model of blind and reactive underwater jamming. *IEEE Trans. Wirel. Commun.* 21, 3737–3751. doi:10.1109/TWC.2021.3123454

Signori, A., Chiariotti, F., Campagnaro, F., and Zorzi, M. (2020). A game-theoretic and experimental analysis of energy-depleting underwater jamming attacks. *IEEE Internet Things J.* 7, 9793–9804. doi:10.1109/JIOT.2020.2982613

Signori, A., Pielli, C., Chiariotti, F., Giordani, M., Campagnaro, F., Laurenti, N., et al. (2021). "Jamming the underwater: A game-theoretic analysis of energy-depleting jamming attacks," in WUWNet '19: Proceedings of the 14th International Conference on Underwater Networks & Systems, USA, October 2019 (IEEE). doi:10.1145/3366486.3366546

Su, Y., Liu, Y., Fan, R., Li, L., Fan, H., and Zhang, S. (2022). A cooperative jamming scheme based on node authentication for underwater acoustic sensor networks. *J. Mar. Sci. Appl.* 21, 197–209. doi:10.1007/s11804-022-00277-8

TeledyneMarine (2023). Teledyne benthos acoustic modems. Available at: http://www.teledynemarine.com/benthos/.

Tomasi, B., Casari, P., Finesso, L., Zappa, G., McCoy, K., and Zorzi, M. (2010). "On modeling JANUS packet errors over a shallow water acoustic channel using Markov and hidden Markov models," in *2010-MILCOM 2010 military communications conference* (IEEE), 2406–2411.USA.

Vadori, V., Scalabrin, M., Guglielmi, A. V., and Badia, L. (2015). Jamming in underwater sensor networks as a bayesian zero-sum game with position uncertainty. *IEEE Glob. Commun. Conf. (GLOBECOM).* 1, 6. doi:10.1109/GLOCOM.2015.7417412

Vandalore, B., Fahmy, S., Jain, R., Goyal, R., and Goyal, M. (2000). General weighted fairness and its support in explicit rate switch algorithms. *Comput. Commun.* 23 (2), 149–161. doi:10.1016/S0140-3664(99)00157-7

Wang, Z., Zhen, F., Zhang, S., Liu, M., and Zhang, Q. (2017). Jamming-resilient algorithm for underwater cognitive acoustic networks. *Int. J. Distributed Sens. Netw.* 13, 155014771772630. doi:10.1177/1550147717726309

Xiao, L., Donghua, J., Wan, X., Su, W., and Tang, Y. (2018). Anti-jamming underwater transmission with mobility and learning. *IEEE Commun. Lett.* 22, 542–545. doi:10.1109/LCOMM.2018.2792015

Xiao, L., Jiang, D., Chen, Y., Su, W., and Tang, Y. (2020). Reinforcement-learning-based relay mobility and power allocation for underwater sensor networks against jamming. *IEEE J. Ocean. Eng.* 45, 1148–1156. doi:10.1109/JOE.2019.2910938

Xiao, L., Li, Q., Chen, T., Cheng, E., and Dai, H. (2014). Jamming games in underwater sensor networks with reinforcement learning. *IEEE Glob. Commun. Conf. (GLOBECOM)* 1, 6. doi:10.1109/GLOCOM.2015.7417192

Xiao, P., Kowalski, M., McCulley, D., and Zuba, M. (2015). "An experimental study of jamming attacks in underwater acoustic communication," in Proceedings of the 10th International Conference on Underwater Networks & Systems, USA, 22 October 2015 (IEEE).1–5.

Xiong, M., Zhuo, J., Dong, Y., and Jing, X. (2020). A layout strategy for distributed barrage jamming against underwater acoustic sensor networks. *J. Mar. Sci. Eng.* 8, 252. doi:10.3390/jmse8040252

Ye, Y., Peng, Z., Arun Raj Kumar, P., Vasudavan, A. R., and Hong, X. (2020). *Active jamming for eavesdropping prevention in underwater wireless networks*. New York, NY, USA: Association for Computing Machinery. doi:10.1145/3366486.3366511

Zhang, Y., Zhang, Z., Chen, L., and Wang, X. (2021). Reinforcement learning-based opportunistic routing protocol for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* 70, 2756–2770. doi:10.1109/TVT.2021.3058282

Zuba, M., Shi, Z., Peng, Z., Cui, J.-H., and Zhou, S. (2015). Vulnerabilities of underwater acoustic networks to denial-of-service jamming attacks. *Secur. Commun. Netw.* 8, 2635–2645. doi:10.1002/sec.507