Journal of
Marine Science and Engineering

MDPI

Article

Asynchronous Pattern-Designed Channel Access Protocol in Underwater Acoustic Wireless Sensor Networks

Jie Ren [1,2,3], Yanbo Wu [1,3,4,*] and Min Zhu [1,3,4]

[1] Ocean Acoustic Technology Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; renjie@mail.ioa.ac.cn (J.R.); zhumin@mail.ioa.ac.cn (M.Z.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Technology Research Center of Ocean Acoustic Equipment, Beijing 100190, China
[4] State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: wuyanbo@mail.ioa.ac.cn

Abstract: Due to the significant propagation delay in underwater sensor networks, conflict retransmission in channel access protocols comes at a high cost. This poses a challenge in scenarios where multiple sensor nodes generate data frames with strong temporal correlations, such as in disaster warning applications. Traditional channel allocation and timeout-based retransmission mechanisms lead to considerable access delays, making it difficult to meet the requirements. To tackle this issue, we propose the asynchronous pattern-designed random access (APDRA) protocol. This protocol enhances the access probability by designing retransmission time intervals for data frames based on pattern design. Additionally, we introduce a successive interference cancellation (SIC) mechanism at the receiver for decoding. This mechanism facilitates the transformation of the conventional method of discarding conflicted data frames into iterative decoding, thereby enhancing transmission efficiency. Via the utilization of simulations, we compare the APDRA protocol conventional underwater medium access control (MAC) protocols and existing retransmission mechanisms. The results demonstrate that the APDRA protocol has the ability to improve both the transmission success ratio (TSR) and reduces the access delay to some extent.

Keywords: underwater acoustic sensor networks; MAC protocol; random access; successive interference cancellation

Citation: Ren, J.; Wu, Y.; Zhu, M. Asynchronous Pattern-Designed Channel Access Protocol in Underwater Acoustic Wireless Sensor Networks. *J. Mar. Sci. Eng.* **2023**, *11*, 1899. https://doi.org/10.3390/jmse11101899

Academic Editor: Daniel Rouseff

Received: 25 August 2023
Revised: 22 September 2023
Accepted: 28 September 2023
Published: 29 September 2023

1. Introduction

With the continuous improvement of human capabilities in exploring the ocean and developing marine resources, point-to-point underwater acoustic communication is no longer sufficient to meet the demands of information transmission [1–3]. A large amount of diverse information needs to be transmitted among the surface stations, underwater fixed nodes, and autonomous underwater vehicles (AUV) [4]. It is inevitable that underwater acoustic communication technology will transit from point-to-point links to networking systems.

In recent years, underwater acoustic sensor networks have been widely applied in various areas, such as marine data collection, underwater environmental monitoring, disaster prediction and alarm systems, and surveillance for military operations. However, compared to terrestrial wireless communication networks, underwater acoustic communication networks face significant challenges. These challenges arise from the long propagation delay caused by the slow speed of underwater sound, as well as the complex and dynamic characteristics of the underwater acoustic channel [5,6].

In the protocol stacks for underwater acoustic communication networking, the Media Access Control (MAC) layer is mainly responsible for managing the channel access of the network nodes [7]. The underwater MAC protocols mainly include the non-contention-based time division multiple access (TDMA) protocol [8], as well as the random-access protocols additive links online Hawaii area (ALOHA) [9,10] and the handshake-based MACAW (multiple access with collision avoidance for wireless) [11]. Although the TDMA protocol can effectively avoid transmission conflicts [12], it is not suitable for delay-sensitive scenarios or systems without an efficient clock synchronization mechanism. The ALOHA protocol allows for rapid access only under conditions of relatively low payload and non-concurrent data transmission from multiple source nodes. As the load rate increases, the probability of collisions also increases, which can result in transmission failures. In the MACAW protocol, the introduction of the acknowledgment and retransmission mechanism [13] can ensure successful delivery rates; however, the overall access delay significantly increases. In many applications of underwater sensor networks, such as disaster warning and prevention, the generation times of the multiple sources are strongly correlated. This correlation makes it difficult for MAC protocols like ALOHA and MACAW to meet the low-delay requirement.

The slotted ALOHA (SA) [14] protocol is an improvement over the pure ALOHA protocol. It divides time into equally spaced discrete slots, and nodes must wait for the next time slot to start transmitting data. This reduces the possibility of data collisions and improves channel utilization. The improved SA protocol allows the sender to transmit multiple identical frames during a transmission process. The receiver can then utilize the conflicting data information to reduce the waiting time for data retransmission and improve the achievable capacity of the channel by the multi-user decoding technique. In this paper, we propose an asynchronous MAC protocol called the asynchronous pattern-designed random access (APDRA) protocol. This protocol is based on the SA protocol and involves the repetition of the same data packet twice at fixed intervals. The main contributions of this paper are as follows:

1. We optimized the repetitive pattern of data transmission. After sending data packets according to random access, different intervals are assigned to different nodes based on the APDRA for data frame retransmission. This approach aims to improve the success rate of communication and save the underwater acoustic channel propagation delay required for channel conflict-retransmission waiting time, thereby enhancing access efficiency.
2. The length of the waiting time is solely dependent on the local node clock, and the communication between nodes in the network is asynchronous. Therefore, there is no need to maintain clock synchronization across all nodes, leading to resource saving in terms of clock synchronization.
3. The proposed protocol can ensure that at least one data frame sent by a node can be correctly decoded by the receiving node, especially in scenarios with a high number of concurrent nodes, such as disaster warnings. If successive interference cancellation (SIC) is introduced, a transmission success rate of 100% can be achieved in cases where the number of sensor nodes is limited, and the physical layer communication is reliable.
4. We made a comparison between APDRA and commonly used MAC protocols for underwater acoustic communication, demonstrating the effectiveness and applicable conditions of APDRA.

The rest of the paper is organized as follows. In Section 2, we review previous research and studies that are relevant to this paper. In Section 3, we propose the APDRA protocol and provide the protocol transmission process. In Section 4, we evaluate the performance of the APDRA protocol using comparative simulations. In Section 5, we summarize our work and discuss future research directions.

2. Related Work

In 1983, G. Choudhury and S. Rappaport proposed diversity ALOHA (DA) [15] as an improvement to the SA protocol. The DA protocol allows users to transmit multiple copies of the same data frame using time diversity. Research has shown that multiple transmissions can provide better delay performance, especially under low traffic load conditions. In 2007, Enrico Casini et al. further improved the DA protocol while studying random access protocols. They proposed a new diversity access scheme called contention resolution diversity slotted ALOHA (CRDSA) [16], which combines diversity transmission of burst data with the successive interference cancellation technique. Results have shown that under the condition of an equal packet loss rate of 0.02, the throughput of the CRDSA protocol is 17 times higher than that of the SA protocol. CRDSA technology can provide low latency for small-sized sparse packet transmission. In 2011, Gianluigi Liva improved the CRDSA protocol by modifying the number of random transmissions from 2 to d, where d is greater than or equal to 2. The specific value of d is determined by each node using a probability distribution function. This retransmission mechanism is named irregular repetition slotted ALOHA (IRSA) [17]. The theoretical throughput of this scheme can reach 0.97, achieving approximately 45% gain compared to the CRDSA protocol. In the same year, Gianluigi Liva et al. proposed a new scheme called coding-slot ALOHA (CSA) [18], which is based on IRSA. Users split burst data frames into segments, encode them locally, and transmit them. The receiver employs the SIC technique and decodes the local code to recover data from data collisions, thereby further improving throughput.

Unlike the channel access protocols mentioned above, the clock synchronization across all the nodes to mark the slot boundaries is not required in the asynchronous contention resolution diversity ALOHA (ACRDA) [19] protocol. Instead, it locally defines slots and frames at each node, referred to as virtual slots and virtual frames. The virtual frame can start as soon as the packet is ready to be transmitted, thanks to the RA's asynchronous nature. All the nodes operate asynchronously with independent time offsets, thereby saving a significant amount of transmitter synchronization overhead. When a sensor node serves as the source node, the access mode of ACRDA shares similarities with that of CRDSA. It requires encoding the replica position information of the same data packet at the same location. However, the position representation of real slots in CRDSA differs from ACRDA, where the position information refers to the number of virtual slot offsets. At the receiving end, SIC technology is still employed to recover collided data packets. Globally, ACRDA does not require slot timing synchronization among transmitters. This significantly reduces system complexity, signaling traffic, and modulator complexity, thus enhancing network scalability and providing better transmission latency performance.

The contention resolution ALOHA (CRA) [20] protocol allows nodes to randomly select a time slot for transmission based on a certain probability within a given time window. It also randomly selects the number of retransmissions. The protocol uses SIC decoding and forward error correction (FEC) to recover the interfered data frame copies. This protocol performs particularly well in scenarios with high signal-to-interference plus noise ratio (SINR). Building upon the protocol, the enhanced contention resolution ALOHA (ECRA) [21] combines the non-interfered portions (or the minimally interfered portions) of all data copies from each user into a new one. This combined packet exhibits similarities to the original frame or has minor interference. The ECRA protocol then applies the SIC and FEC processes of the CRA protocol, thereby improving the efficiency of the CRA protocol. It has been shown that ECRA can achieve significant improvements in both throughput and packet error rate (PER) compared to CRA.

3. The Proposed Protocol

In this section, we propose the asynchronous pattern-designed random access (AP-DRA) protocol, which is primarily designed for situations with multiple source nodes, and one receiver node. As depicted in Figure 1, the communication buoy functions as a surface node and serves as the central node in this network. After collecting the information from the sensor nodes, the communication buoy can process the data and transmit it to a communication satellite using wireless communication. Sensor nodes in the network are mobile, and the distances between them and the underwater central node are randomly distributed. The total number of sensor nodes is not predefined or fixed, and new sensor nodes can be added to the network. However, it is recommended that the total number of sensor nodes be less than or equal to 7. This recommendation is based on the protocol transmission pattern, which is explained in more detail in Sections 3.2 and 3.4.

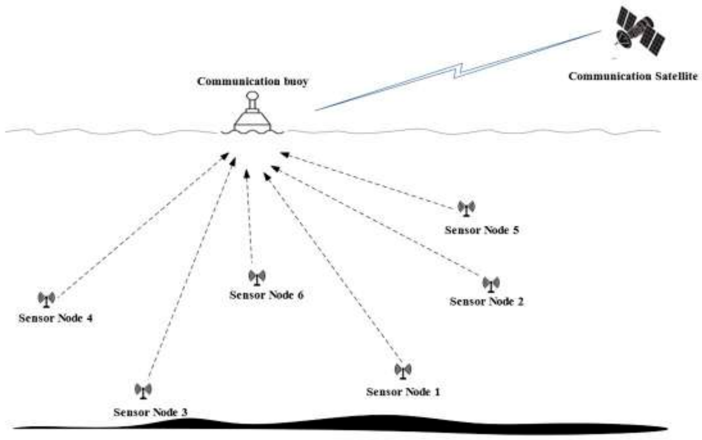


Figure 1. Network Scenario Diagram.

3.1. Overview of APDRA MAC Protocol

The transmission process of the APDRA protocol can be divided into two parts: (a) The sensor nodes have burst data frames that need to be transmitted, and then they transmit the data to the central node according to the designed protocol. (b) After receiving the data, the central node may face an increased probability of collisions because the sensor nodes transmit the data frames twice at regular intervals according to the APDRA protocol. To enhance the decoding accuracy at the receiver, the successive cancellation method SIC is utilized. The flowchart illustrating this process is shown in Figure 2.

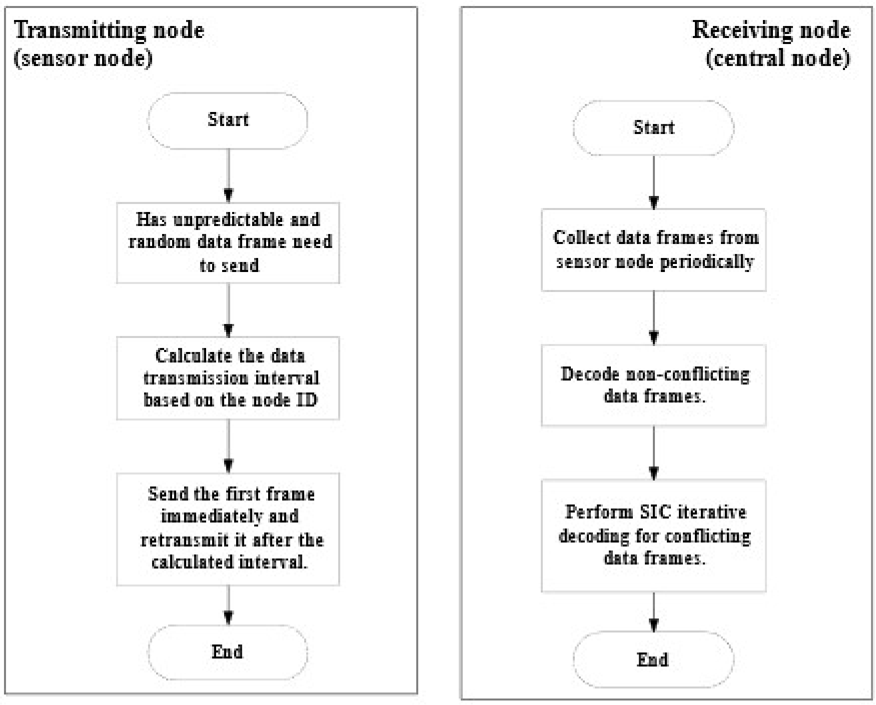


Figure 2. APDRA Protocol Flowchart.

According to the designed protocol, the sensor nodes in the network are asynchronous, meaning they do not have synchronized clocks. Each node independently defines its own time slots and frame boundaries based on its local time reference. As a result, there may be time offsets between different nodes. The following two sections will provide detailed explanations of the frame transmission mechanism for the sending nodes and the iterative decoding mechanism for the receiving nodes.

3.2. Design of Data Frame Retransmission Mechanism

In Section 2, we provide an overview of relevant protocols, and Table 1 describes the retransmission mechanisms commonly used in MAC protocols. Currently, many MAC protocols with retransmission mechanisms are designed under the assumption of synchrony. These protocols define a time window as the available transmission interval for data frames. They aim to enhance retransmission attempts for the same data frame, such as CRDSA, or achieve irregular repetition via degree distribution, like IRSA. Specifically, the CRDSA protocol can be regarded as a special case of the IRSA protocol when there are exactly two retransmission attempts. On the other hand, there are fewer protocols that specifically address asynchronous scenarios, and ACRDA is a notable example. In ACRDA, the length of the time window is specific to each local node, and frames and time slots are defined based on the receiver timeline, which is referred to as the virtual frame (VF). ACRDA can be considered an enhanced version of the CRDSA protocol, specifically adapted for asynchronous conditions.

Table 1. Current designs of the retransmission mechanisms in protocols.

Protocol	Communication Mode	Data Frame Retransmission Mechanism
CRDSA	Synchronous	In a time window, randomly select two time slots to send.
IRSA	Synchronous	In a time window, design d ($d \geq 2$), the number of frames to be sent. Use degree distribution to randomly select n time slots for transmission.
ACRDA	Asynchronous	Similar to CRDSA, but the time slots are only defined within the local node and are not applicable to the entire network.

Unlike ACRDA, where the time window length is fixed for each node, we propose a design in which the time window length is variable and determined by the ID of the sensor node. This approach aims to minimize the likelihood of collisions. When a sensor node has data to transmit, it immediately initiates the first transmission of the data frame. Asynchronous communication is utilized, so the node only needs to rely on its local clock and does not require global time synchronization or the need to wait for time slots to arrive.

Sensor nodes must wait for a specific time period before retransmitting the data frame. The length of the time window varies depending on the ID of the sensor node. The smallest unit of time slot is determined by the length of the data frame. The formula for calculating the length of the time slots for data frame retransmissions for sensor nodes with IDs $1, 2, \cdots, n$ is presented in Equation (1).

$$\begin{cases} I_1 = 1, \\ I_2 = (I_1 + 1) + 1 = 3, \\ I_3 = (I_1 + 1) + (I_2 + 1) + 1 = 7, \\ \cdots \\ I_n = (I_1 + 1) + (I_2 + 1) + \cdots + (I_{n-1} + 1) + 1 = 2^n - 1. \end{cases} \quad (1)$$

In Equation (1), if we set the time of the first data frame transmission as 1, we can I_1 , I_2 , $\cdots$, I_n represent the transmitting times of the second data frame from sensor nodes 1 to n. Therefore, the expression for the total length of the time window associated with node IDs can be derived, as shown in Equation (2). It is evident that for two adjacent nodes in terms of their ID, the time interval for the node with a higher ID is equal to the length of the time window of the preceding node. Under asynchronous conditions, where sensor nodes are mobile, and the central node may not have real-time knowledge of the positions of all nodes, this method can effectively minimize the probability of receiver collisions from the perspective of the transmitter.

$$Window_n = I_n + 1 = 2^n - 1 + 1 = 2^n \tag{2}$$

It is important to note that the proposed retransmission interval mechanism may lead to varying lengths of transmission time windows for different nodes, resulting in uneven opportunities among them. The ordering of sensor node IDs is directly related to the length of the time window, with lower ID nodes having a higher probability of successful transmission. To address this issue, different approaches can be considered depending on the application scenario where data frames are sent from sensor nodes to a central node with a consistent purpose, such as disaster warning. In this scenario, where all nodes transmit the same data content and information from nodes farther away is more crucial, ID numbers can be assigned in ascending order based on the distance from the central node, starting from the farthest node and moving towards the closest. For scenarios where each node transmits different data content but with equal priority, ID numbers can be randomly assigned and periodically rotated using modular arithmetic with the total number of sensor nodes. This ensures fairness and equal distribution of transmission opportunities among nodes. In conclusion, the selection of an ID ordering method should be determined by the specific requirements and characteristics of the application scenario.

Under the condition that the sensor node transmits data frames to the central node for the same purpose, it can be considered that the data content transmitted by each node is the same. We propose Proposition 1 and provide an explanation for it.

Proposition 1. *In the asynchronous communication mode, the APDRA protocol transmission mechanism ensures that the central node can receive data from at least two nodes without any collisions, regardless of when the nodes initiate data transmission.*

Explanation of Proposition 1. The nature of sensor networks, where sensor nodes can move, and their distances from the central node are uncertain, introduces challenges in synchronization and timing. When sensor nodes receive broadcast commands from the central node and respond with data frames, or when they autonomously decide to transmit data, it is necessary to normalize the unknown propagation delay and the unpredictable timing of data with respect to the timeline of the receiving end. To address this challenge, the normalization process involves aligning the timing of data generation and propagation delay with a reference timeline based on the perspective of the central node. This ensures that the data received at the central node can be correctly interpreted and processed, regardless of the randomness in the timing of data generation and propagation delay.

In Figure 3, the diagram illustrates the transmitter and receiver of the APDRA protocol. Let's consider the scenario with four sensor nodes as an example. The time axis of the transmitting nodes (IDs 1 to 4) represents their respective local timelines. The starting point is when each node generates its first data frame. According to the protocol designed, nodes with different IDs wait for a fixed time interval that is related to their ID before sending repeated data frames. The central node serves as the receiver and uses its own timeline as a reference. In the worst-case scenario, where there are n sensor nodes, $(2n - 2)$ data frames will collide. Figure 3 shows an illustration of one of the worst-case scenarios at the receiver end. It is worth noting that due to the mobility of sensor nodes in the APDRA protocol, it is somewhat challenging for the probing nodes to report their current position coordinates in real time to the buoy. The random time offset of each node in Figure 3 is a normalized deviation that considers the asynchronism between nodes, the occurrence of data bursts at different times, and the impact of different distances between nodes and the buoy.

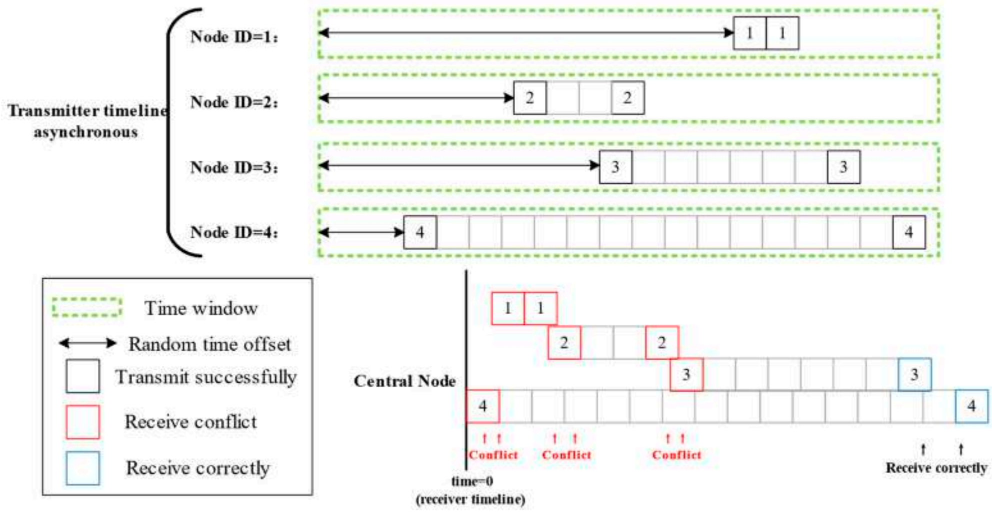


Figure 3. Receiver Status Diagram (Example of Maximum Collision Scenario with Four Nodes).

We believe that the primary reason for incorrect reception at the receiving nodes is the occurrence of data frame collisions. The severity of these collisions can be categorized into two cases. Case 1: Multiple data frames collide simultaneously at the same time. Case 2: All nodes' data frames collide with each other pairwise. Since each node has two opportunities to send the same data frame, and the interval between data frame transmissions is determined by the IDs and is never equal, if multiple data frames collide at the same time, it is guaranteed that the other data frame sent by the same node does not, ensuring correct reception. However, if a node's transmissions collide with data frames from two different nodes, the performance of the receiving device is compromised.

For "In a network with n ($n \geq 2$) sensor nodes using the APDRA protocol, the receiver is guaranteed to receive 2 data frames from different nodes without collision," we will provide a proof using mathematical induction.

(1) When $n = 2$, we can deduce that the worst-case scenario for reception is the collision between the first and last frames of the two nodes, or between their first frames, or between their retransmission frames. This is because the time window length for frames sent by different nodes is different. In any case, it can be ensured that each of the two nodes has received at least one frame without collision;

(2) Assuming it holds true for $n = k$, one of the worst-case scenarios for the reception at the central node is the collision of k nodes' frames, resulting in $(k - 1)$ collisions. In other words, there are $(2k - 2)$ data frames affected, and 2 data frames from different nodes are received without any collisions.

(3) When $n = k + 1$, considering the worst-case scenario discussed in (2), the addition of the $(k + 1)$-th node introduces additional collision conditions. According to Equation (1), the time window length for the $(k + 1)$-th node is the sum of the time window lengths of the first k nodes. Combining this with (1), it can be concluded that regardless of whether the data frame sent by the $(k + 1)$-th node collides with the first or last frames of the previous k nodes or with their retransmission frames; it guarantees that the number of collision frames for the $(k + 1)$th node is less than or equal to 1. Therefore, a total of $(2k - 2) + 1 + 1$ data frames are affected, ensuring the collision-free reception of at least two data frames. Due to the occurrence of collisions in the network, it is impossible for these two data frames to originate from the same node.

Thus, it is proven.

Proposition 1 explains that the APDRA protocol serves the consistent purpose of rapidly and reliably transmitting data from various sensor nodes to the central node in scenarios such as natural disaster warnings or military alerts. In these situations, it can be assumed that the sensor nodes send consistent data content. The APDRA protocol ensures that regardless of when the sensor nodes detect environmental information and report it proactively, at least two nodes can transmit data frames without any collisions. While traditional TDMA protocols can achieve this as well, they may sacrifice access delay as a trade-off. In contrast, the APDRA protocol aims to minimize access time as much as possible, as will be concluded in the fourth section.

Although the data retransmission mechanism of the APDRA protocol ensures successful access for at least two nodes, there is a high probability of data frame collision scenarios involving daily sensor detection and of self and surrounding environmental data, among other similar applications. To further enhance the success rate of data transmission, the APDRA protocol incorporates the SIC technique at the receiving end to improve efficiency. A thorough elucidation of this matter will be presented in Section 3.3.

3.3. Successive Interference Cancellation Processing at the Receiver

In traditional MAC protocols, when multiple source nodes send data to the same receiving node and collision occurs, the receiving node considers all transmissions from the source nodes as failed. As a result, the receiving node discards the collided data without responding. The source nodes then wait for a certain period, and if they do not receive an ACK frame, they initiate the retransmission of the data frames. This not only leads to a significant increase in access delay but also wastes channel resources, especially in underwater acoustic channels where propagation delay is high. To overcome this challenge, the application of the multiple user detection (MUD) technique proves to be effective for multi-packet reception (MPR). In the context of sensor networks, employing complex physical layer decoding methods becomes counterproductive for improving transmission efficiency and accuracy. Therefore, the APDRA protocol adopts the SIC technique as the detection technology at the receiving node. This technique allows the receiving node to detect and decode multiple overlapping packets simultaneously.

The condition for a data packet to be correctly received can be explained using the Shannon formula. In the context of multi-packet reception, the Shannon formula can be redefined as Equation (3), where $\mathbf{S}_i$ represents the signal power of node i and $\sum\limits_{i \neq j} \mathbf{S}_i$ represents the combined signal power from all other sensor nodes except node i.

$$\mathrm{C} = \mathrm{W}\log_2\left(1 + \frac{\mathbf{S}_i}{\sum_{j \neq i} \mathbf{S}_j + \mathbf{N}_0}\right) \tag{3}$$

This allows us to calculate the signal-to-interference plus noise ratio for node i, as expressed in Equation (4).

$$\mathrm{SINR}_i = \frac{\mathbf{S}_i}{\sum_{j \neq i} \mathbf{S}_j + \mathbf{N}_0} \tag{4}$$

For node i to receive the data correctly, the SINR must exceed a given threshold. The iterative decoding process begins by selecting a data frame from a node that did not experience collisions while considering the data from other nodes as interference. After the initial decoding attempt, the receiving node conducts a cyclic redundancy check (CRC) on the data frame. If the CRC check passes, the superimposed data at the receiving node is subtracted from the reconstructed frame, completing the first round of SIC checks as described in Equation (5).

$$\widetilde{y}_1 = y - h_1 \times \widetilde{x}_1 \tag{5}$$

The variable y represents the multi-node superimposed data received by the receiving node, $\widetilde{x}_1$ represents the data frame reconstructed from a specific node, h_1 represents the underwater acoustic channel that the node has passed through, and $\widetilde{y}_1$ represents the superimposed data frame after eliminating interference from that node. By following this approach, we can obtain the data received by the receiving node after the jth iteration of interference elimination, as shown in Equation (6).

$$\widetilde{y}_j = \widetilde{y}_{j-1} - h_{j-1} \times \widetilde{x}_{j-1} \quad (6)$$

In this Equation, $\widetilde{y}_j$ and $\widetilde{y}_{j-1}$ represent the remaining data at the receiving node after the jth and $(j-1)$-th iterations of serial interference cancellation and decoding, h_{j-1} represents the underwater acoustic channel and $\widetilde{x}_{j-1}$ represents the reconstructed data frame obtained from the jth iteration. For long-range underwater acoustic communication systems, we believe that the underwater acoustic channel can be modeled as a Rayleigh fading plus Gaussian white noise model [22]. In this model, h_{j-1} can be represented by Equation (7).

$$h_{j-1} = \sum_{l=0}^{n-1} h_l(j-1)\delta(\tau) \quad 1 \leq j \leq n+1 \quad (7)$$

In this Equation, j represents the jth subcarrier, and n is the total number of subcarriers. $h_l(j-1)$ represents the tap gain of the lth multipath, which can be modeled using the Rayleigh model. $\delta(\tau)$ denotes the impulse response.

For underwater acoustic communication channels, when there is a data conflict between sensor nodes i and j at the central node, the frequency domain representation $\boldsymbol{x}_{\text{total}}$ of the total transmitted frames from nodes i and j can be expressed as Equation (8) for the central node.

$$\boldsymbol{x}_{\text{total}} = \alpha_i \boldsymbol{x}_i + \alpha_j \boldsymbol{x}_j \quad (8)$$

where α_i and $\alpha_j \in [0, 1]$ represent the different power coefficients allocated to nodes i and j in the network. Assuming that the length of the signal's cyclic prefix is greater than the maximum multipath delay in the underwater acoustic channel, $\hat{\mathbf{H}}$ represents the estimated frequency response of the underwater acoustic channel at the central node, and $\mathbf{N}$ represents the frequency domain additive noise. Thus, the frequency domain response of the received signal at the central node can be represented by Equation (9).

$$\boldsymbol{y} = \hat{\mathbf{H}}\left(\alpha_i \boldsymbol{x}_i + \alpha_j \boldsymbol{x}_j\right) + \mathbf{N} \quad (9)$$

For node i, node j is considered as part of the interference. $\hat{\mathbf{H}}_i$ represents the estimated frequency response of the channel at node i, and the interference term can be written as $\boldsymbol{\xi}_1 = \alpha_j \hat{\mathbf{H}}_i \boldsymbol{x}_j + \mathbf{N}$. The frequency-domain representation of the received signal for node i $\boldsymbol{y_i}$ is shown in Equation (10).

$$\boldsymbol{y_i} = \alpha_i \hat{\mathbf{H}}_i \boldsymbol{x}_i + \boldsymbol{\xi}_1 \quad (10)$$

Using the APDRA protocol, if node i has an unconflicted data frame and a retransmission frame that overlaps with node j, we can obtain an estimation value $\hat{\boldsymbol{x}}_i$ for the collision signal of node i by utilizing the unconflicted data frame from node i. After performing channel decoding and encoding on $\hat{\boldsymbol{x}}_i$, the reconstructed signal $\overline{\boldsymbol{x}}_i$ is obtained. By removing the reconstructed signal from the equalized received signal, the signal estimation value of node j ($\hat{\boldsymbol{x}}_j$) can be obtained, as shown in Equation (11).

$$\hat{\boldsymbol{x}}_j = \frac{\left(\hat{\mathbf{H}}_j^{\mathrm{H}} \hat{\mathbf{H}}_j + \sigma_j^2 \mathbf{I}\right)^{-1} \hat{\mathbf{H}}_j^{\mathrm{T}} \boldsymbol{y} - \alpha_i \overline{\boldsymbol{x}}_i}{\alpha_j} \quad (11)$$

In the Equation, $\hat{\mathbf{H}}_j$ represents the channel frequency response estimation of node j, and σ_j^2 represents the noise variance of node j.

In [23], the decoding results of multiple sensor nodes in underwater acoustic communication are provided. After the second iteration, the decoding performance stabilizes. By the fifth iteration, the bit error rate (BER) can be reduced below 10^{-3}. Therefore, for the APDRA protocol, we can consider SIC iterative decoding to be consistently effective.

The order of data frame detection in SIC is crucial because it determines the sequence in which the data frames are decoded and subsequently used for data reconstruction and interference cancellation. If errors occur during the decoding process, they can result in inaccurate decoding of the subsequent data frames and even propagate errors. This holds true for both synchronous communication protocols, such as CRDSA and IRSA, and asynchronous communication protocols, like ACRDA.

However, in contrast to the MAC protocol mentioned above, Proposition 1 has demonstrated that the APDRA protocol ensures the correct reception of data frames by at least two nodes. By using received data frames from these nodes as different starting points for interference cancellation iterations, serial interference cancellation can be accomplished using various node ordering methods. Figure 4 illustrates this process, where the decoding starts by considering the scenario depicted in Figure 3. By choosing node 4 and node 3 as the initial points for decoding and implementing interference cancellation, the potential errors resulting from serial decoding can be reduced. The time required for SIC iterative decoding is related to the design of the demodulator. Taking CRDSA as an example, the delay of demodulation processing is less than one frame [16]. From a hardware implementation perspective, it can be considered that this type of demodulator is equally effective for the APDRA protocol. Therefore, it is recommended to wait for the delay of a data frame for the communication beacon to perform iterative decoding. This should be done after the complete time window for data transmission from a sensor node ends. It helps to avoid buoy overload.

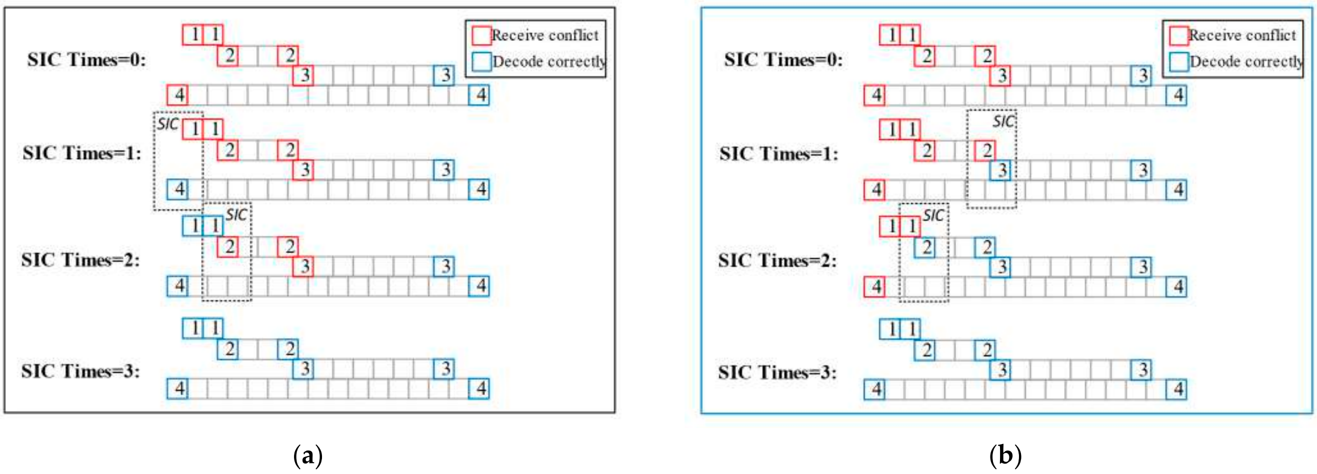


Figure 4. Iterative interference elimination process. (**a**) Decoding starts with the retransmitted frame from Node 4 as the decoding starting point; (**b**) Decoding starts with the retransmitted frame from Node 3 as the decoding starting point.

Based on the analysis of SIC, the CRDSA, IRSA, and ACRDA protocols have a significant drawback: they may lead to deadlock situations. In these deadlocks, the receiving is unable to cannot determine the transmitted contents of the two nodes involved in a complete collision of data frames. However, this issue can be addressed by the APDRA (Asymmetric Power Division Random Access) protocol. We present Proposition 2 and its explanation as follows.

Proposition 2. *In the case of a limited number of nodes, the APDRA protocol achieves a transmission success rate of 100% by iterative decoding. The required number of iterations does not exceed (n − 2) times, where n represents the number of sensor nodes.*

Explanation of Proposition 2. Due to the limitations in the transmitting intervals of the APDRA protocol, it is not possible for the transmission timings of the first data frame and the retransmission frames from each sensor to align perfectly. Consequently, the most complex decoding scenario occurs when n nodes have their data frames collide. In this scenario, the first and last frames collide from (n – 2) nodes, while the other two frames conflict with the remaining nodes. Since the maximum number of conflicting data frames at any given time is 2 (due to the first and last collisions), the nodes with decodable frames can be used as the starting point for interference cancellation. Via a maximum of (n – 2) iterations of interference cancellation, all data frames can eventually be decoded successfully. Therefore, in the absence of any other interfering factors, it can be argued that the APDRA protocol achieves a transmission success rate of 100%.

But Proposition 2 is a theoretical inference of the APDRA protocol. In practice, the success rate may be influenced by various factors, which could potentially result in a decline in performance.

3.4. Summary of APDRA Protocol

Based on the analysis mentioned above, the APDRA protocol exhibits certain advantages in terms of transmission accuracy and iterative efficiency. However, it also suffers from clear disadvantages. When compared to other retransmission protocols, one notable drawback is the exponential increase in the length of the time window. This is caused by the fixed data frame retransmission interval assigned to different nodes. By considering the minimum unit of data frame transmission delay as one, when the number of sensor nodes reaches six, the normalized window length expands to 64. Similarly, with seven nodes, the window length increases to 128. Furthermore, Equation (12) provides an alternative formula for calculating the window length.

$$Window_n = D_\mathrm{t} + D_\mathrm{p} + D_\mathrm{t} = 2D_\mathrm{t} + D_\mathrm{p} \tag{12}$$

D_t and D_p respectively refer to the transmission delay and propagation delay of the data frame.

Based on Equation (2), the length of the time window is influenced by the number of nodes. As the number of nodes increases, the length of the time window at the receiving end also increases substantially. Consequently, a problem arises where the sensor nodes at later positions in the ordering haven't waited long enough for the retransmission time of the data frame, while the sensor nodes at earlier positions have already completed data transmission. This creates a significant imbalance in data exchange among nodes, rendering the protocol meaningless. Assuming that $D_\mathrm{t} = 1$ s, the time window length for sensor node ID 7 in the network is 128 s. Excluding the time taken for two data transmissions, which equals $D_\mathrm{p} = 126$ s. Assuming the propagation speed of the underwater acoustic channel is $\mathrm{c} = 1500$ m/s; data retransmission is only effective when the distance between the node and the central node exceeds $Distance = D_\mathrm{p} \cdot \mathrm{c} = 189$ km. Otherwise, switching to protocols like TDMA may be more practical. Similarly, it can be inferred that the sensor node with ID 6 should be at least 93 km away from the central node. Otherwise, the benefits of the data retransmission mechanism over timeout retransmission and wait-for-retransmission MAC protocols will be lost.

However, the effectiveness of the APDRA protocol becomes more apparent in scenarios with a small number of nodes (seven or fewer) and shorter data frame lengths. In these cases, the APDRA protocol demonstrates better performance.

The pseudocode for the APDRA protocol is outlined in Algorithm 1. Algorithm 1 is designed to simulate the entire network rather than simulating the actions of individual nodes. It can be seen as a transmission mechanism for sensor nodes and SIC decoding for receiving nodes.

Algorithm 1. APDRA Protocol Algorithm

Input: Normalized window length W_l, data frame generation probability of sending node $Prob_g$, and maximum successive interference cancellation iteration count Max_{SIC}.

Output: total number of successfully sent nodes Num_s and total access delay $Delay_{access}$.

$Num_s = 0,\ Delay_{access} = 0$;

According to Equation (2), calculate the number of sensor nodes n using W_l

for i = 1 to n **do//**obtain the frame transmission time

//Not every sensor node needs to send data within every time window.

randomly generate a probability p between 0 and 1;

//The probability of data generation follows a uniform distribution.

if $p < Prob_g$ **then**

//The central node normalizes the receiving time T_{twins}, considering the data

//generation time, the impact of asynchronism and transmission delay.

generate T_{twins} for node i according to Equations (1) and (12);

else

silence, no data was sent this time.

end if

end for

let *NodeCol* be a new $2n \times n$ matrix

for j = 1 to n **do**

if the abs ((T_{twins} of node j) $-$ (T_{twins} of other node)) ≤ 1 **then**

//The $(2j-1)$-th row of matrix *NodeCol* records the ID numbers of the other

//nodes that have data frame collisions with node j for the first frame, while

//the $(2j)$-th row records the numbers of the other nodes that have conflicts

//with the retransmitted frame of node j.

update conflict matrix *NodeCol*;

end if

end for

the number of iterations for SIC $Times_{SIC}$ = 0;

while ($Times_{SIC} < Max_{SIC}$) **do**

for k = 1 to n **do**

if (the $(2k-1)$- th row of $NodeCol = 0$) || (the $(2k)$ -th row of *NodeCol* = 0) **then**

find another frame of *NodeCol* for node k;

update *NodeCol*;

$Times_{SIC}$ + = 1;

end if

end for

end while

for m = 1 to n **do**

if *NodeCol* of m = 0 **then**

Num_s + = 1;//The number of successful decoding nodes has increased.

Update $Delay_{access}$;

end if

end for

return $Num_s,\ Delay_{access}$.

4. Simulation Results Analysis

The simulation platform used in this paper mainly includes Python 3.6 and the discrete event simulation platform SimPy. Similar effects can also be achieved with MATLAB and SimEvents, which is a module of Simulink. Due to the constraint that the number of sensor nodes for the APDRA protocol cannot be too large, SimPy is used to simulate the process of all sensor nodes in the network generating data once and transmitting two identical data frames to the central node. Experimental data is collected using the Monte Carlo method. Assuming a data transmission rate of 200 bps, the underwater sound speed of 1500 m/s, and data sent by each sensor node not exceeding 15 bytes, the maximum distance between each sensor node and the central node is 120 km.

4.1. Comparison with Classical Underwater Acoustic MAC Protocols

In the field of underwater sensor network research, two commonly used classical MAC layer protocols are Slotted ALOHA (SA) and TDMA. SA, as a random-access protocol, has the advantage of high channel utilization. On the other hand, TDMA employs time slots for channel allocation, ensuring collision-free transmission. While both SA and TDMA require channel partitioning into fixed time slots, TDMA allows only one node to send data in each time slot, whereas SA permits nodes to select the next available time slot closest to the current moment for frame transmission when data is ready to be sent. In the event of conflicts or unsuccessful transmission, transmission is considered unsuccessful, and the respective data frame needs to be retransmitted. TDMA can be viewed as a special case of SA, where data frames for transmission are generated at fixed time intervals. In the following Sections 4.1.1 and 4.1.2, a comparative evaluation of the APDRA protocol will be presented in comparison to the SA and TDMA protocols, supported using simulation results.

4.1.1. Comparison of Transmission Success Ratio between APDRA and SA

The APDRA and SA protocols, both belonging to the random-access protocols, may experience collisions. Figure 5 illustrates a comparison of the transmission success ratio among these protocols, considering different window lengths.

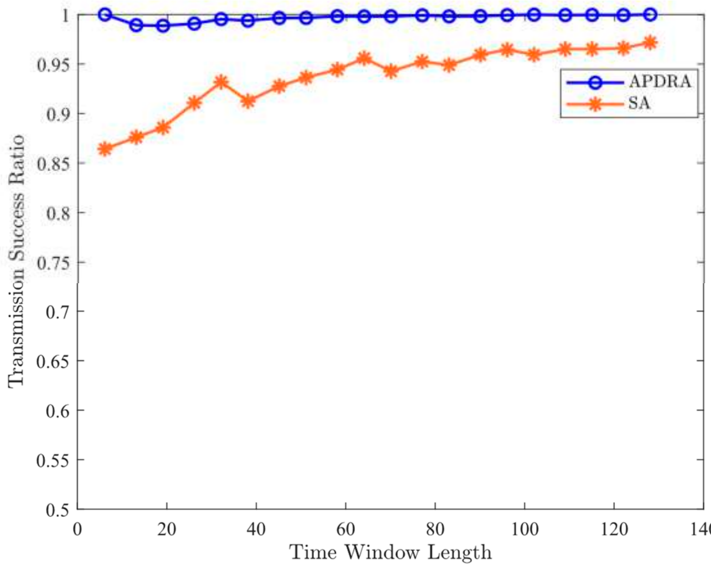


Figure 5. Comparison of transmission success ratio with SA protocol.

Based on Equation (2), we can calculate the maximum number of supported sensor nodes considering the length of the normalized time window. Setting the window length to 128, a maximum of 7 sensor nodes can be supported. The probability of a sensor node generating a data packet at a certain moment is 0.8. The iteration count for successive interference cancellation decoding is 1, and the Monte Carlo simulations are performed 2000 times.

The transmission success ratio (TSR) is calculated using Equation (13), while another important performance metric that evaluates the quality of communication is the packet loss ratio (PLR), calculated according to Equation (14).

$$\mathrm{TSR} = \left(\frac{Num_{\text{success\_node}}}{Num_{\text{total\_node}}} \right) \times 100\% \tag{13}$$

$$\mathrm{PLR} = \left(\frac{Num_{\text{input\_frame}} - Num_{\text{output\_frame}}}{Num_{\text{input\_frame}}} \right) \times 100\% \tag{14}$$

$Num_{\text{success\_node}}$ represents the number of nodes with successful data frame transmission while $Num_{\text{total\_node}}$ denotes the number of nodes that have sent data frames. $Num_{\text{input\_frame}}$ signifies the total number of data frames transmitted using sensor nodes and $Num_{\text{output\_frame}}$ represents the number of data frames received by the central node. Since the APDRA protocol involves the repetition of sending two identical data frames by sensor nodes, we consider the transmission success ratio to be a more meaningful metric than the packet loss ratio. Based on Figure 5, it is evident that in the case of SIC decoding once, the APDRA protocol has achieved a higher transmission accuracy compared to SA.

4.1.2. Comparison of Access Delay between APDRA and TDMA

The TDMA protocol is widely recognized as a collision-free protocol, which means that, assuming no other influencing factors, the transmission success rate of the TDMA protocol can be considered 100%. However, in practical scenarios where sensor nodes have data to transmit but it is not their designated time slot, they must wait until their local node communication slot arrives. The waiting time associated with the TDMA protocol results in a loss of access time compared to the APDRA random access protocol.

We believe that the maximum number of transmitting nodes supported by the APDRA protocol is seven. Although the TDMA protocol allows the existence of multiple transmitting nodes due to its synchronization and channel partitioning features, for the sake of a fair comparison, our simulation assumes equal frame lengths and seven transmitting nodes. Thus, the length of the time window can be determined as 128, based on Equation (2). Figure 6 illustrates the simulated results for total access delay under the given conditions. Since the distance between the sensor nodes and the central node is uncertain, the generation of data frames by the sensor nodes is random. Therefore, the total access delay $D_{\text{end}-\text{end}}$ is normalized and calculated using Equation (15).

$$D_{\text{end}-\text{end}} = D_{\text{proc}} + D_{\text{t}} + D_{\text{p}} \tag{15}$$

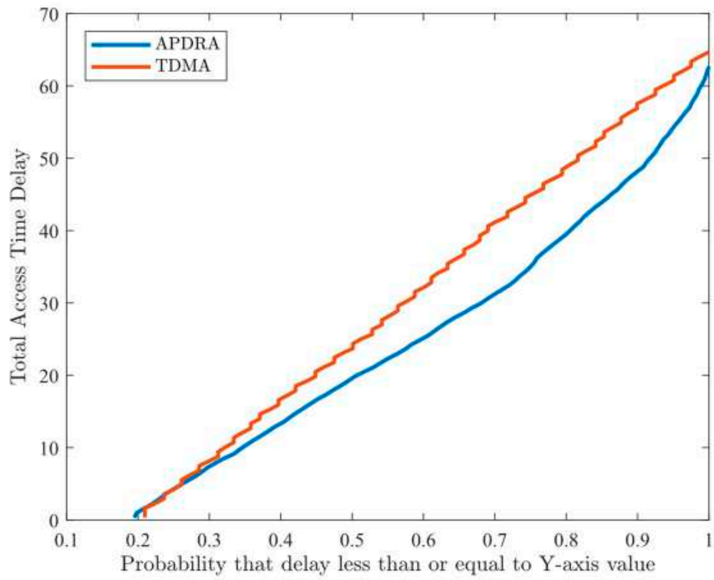


Figure 6. Comparison of Access Time Delay.

D_{proc} represents processing delay. Due to the collision-free nature of the TDMA protocol, to ensure fairness in the comparison simulation, we set the SIC time for the APDRA protocol to 0 in order to ensure fairness in the comparison simulation. It can be observed that when comparing the APDRA protocol to the TDMA protocol, the APDRA protocol has a higher probability of achieving the same access delay time. Additionally, the access delay of the APDRA protocol is smaller, making it more suitable for scenarios that necessitate fast data transmission.

4.2. Comparison with Retransmission Mechanism Protocols

The following discussion focuses on the synchronous protocol CRDSA and the asynchronous protocol ACRDA in the retransmission mechanism protocol. It compares the transmission success ratio, the number of iterations, and the end-to-end access delay.

4.2.1. Transmission Success Ratio

The parameter settings for the simulation are shown in Table 2.

Table 2. Simulation settings among retransmission protocols.

Parameter	Value
Maximum Time Window Length	128
Monte Carlo Experiment Iterations	2000
Probability of Generating Data Frame	0.8
Number of Sensor Nodes	No more than 7, varying with Time Window Length

The simulation results in Figure 7a,b compare the transmission success ratio of the APDRA and CRDSA protocols, as well as the ACRDA protocol, at different numbers of SIC iterations. For the synchronous CRDSA protocol, the transmission success ratio increases with the increase in time window length and the number of SIC iterations. For the asynchronous ACRDA protocol, due to the relaxed requirements on transmission timing, data transmission exhibits greater randomness, resulting in a significant fluctuation in the transmission success ratio. However, the APDRA protocol can maintain a high transmission rate close to 1, even with a small number of iteration decoding attempts.

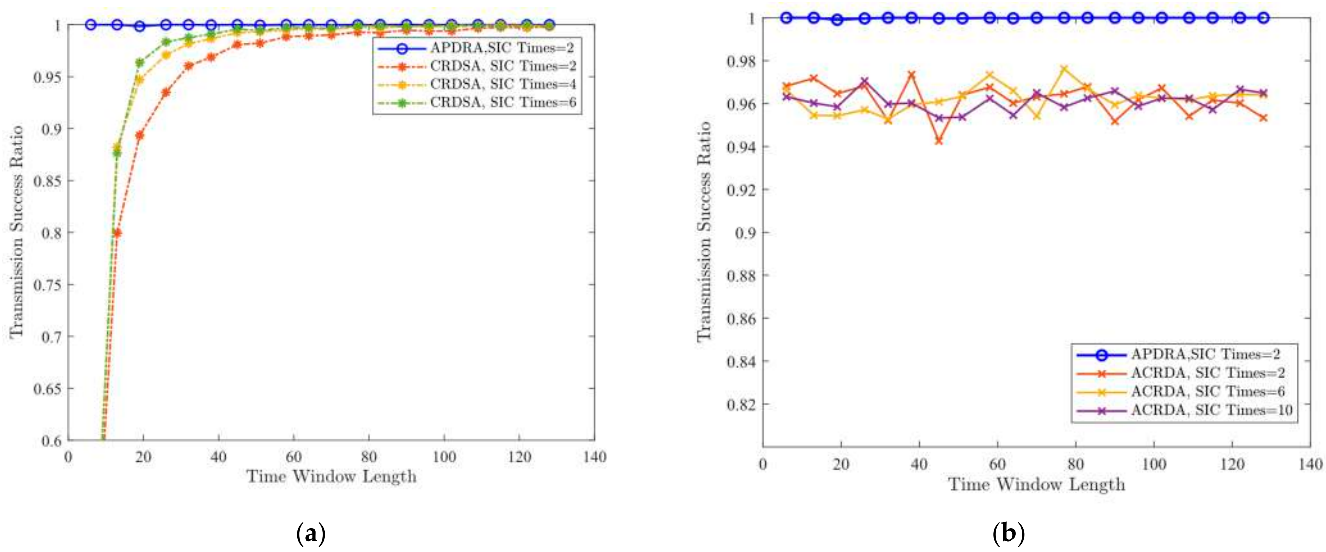


Figure 7. Comparisons of Transmission Success Rate for Different Iteration Numbers. (**a**) Comparison between APDRA and CRDSA Protocols; (**b**) Comparison between APDRA and ACRDA Protocols.

4.2.2. Number of Iterations for SIC

From Figure 7, it can be observed that the number of SIC iterations has a significant impact on the protocol channel quality for protocols like CRDSA and ACRDA, where data frames and retransmission frames are randomly generated. However, the impact on our proposed APDRA protocol is relatively small. The spacing between the transmitted frames in the APDRA protocol is carefully designed to minimize collisions as much as possible. If a collision does occur, the decoding can still be achieved with a small number of iterations using another frame sent by the same node. This is demonstrated by Proposition 2. By simulating the TSR under different iteration numbers ranging from 1 to 20 with the same simulation parameter settings, as shown in Table 2, we generated Figure 8.

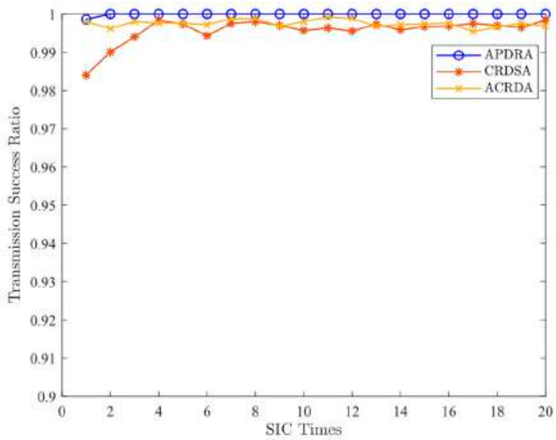


Figure 8. The impact of different SIC decoding iterations on the TSR.

In real-world scenarios, the time required for iterative decoding is longer, and the post-processing time for received data frames is also longer. Moreover, since interference cancellation is performed successively, the more iterations that are performed, the higher the likelihood of data distortion. Although other protocols can achieve high decoding accuracy, the APDRA protocol stands out for its ability to successfully decode data frames with fewer iterations. It rarely encounters situations where decoding fails, leading to a more accurate and efficient communication process.

4.2.3. End-to-End Delay

The ACRDA protocol can be seen as an extension of the CRDSA protocol in an asynchronous communication mode. Figure 9 illustrates the comparison of the access latency between the ACRDA and APDRA protocols. Because the data frames in the ACRDA protocol are randomly generated within a time window, while the data frames in the APDRA protocol are intentionally designed, there is a certain degree of reduction in access delay when the first frame sent by a sensor node according to the APDRA protocol is correctly received. We provide simulation results of the access delay for both the ACRDA and APRA protocols, with simulation parameters set as shown in Table 3.

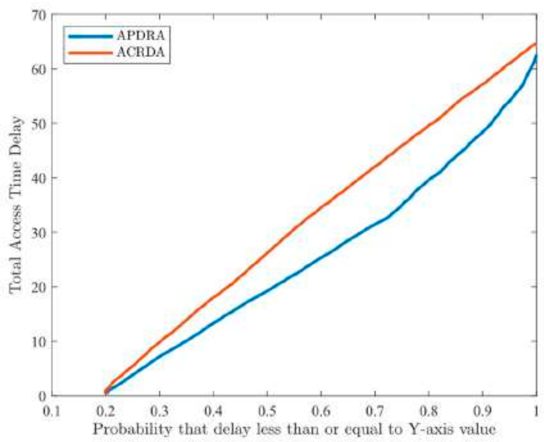


Figure 9. Comparison of total access delay with ACRDA protocol.

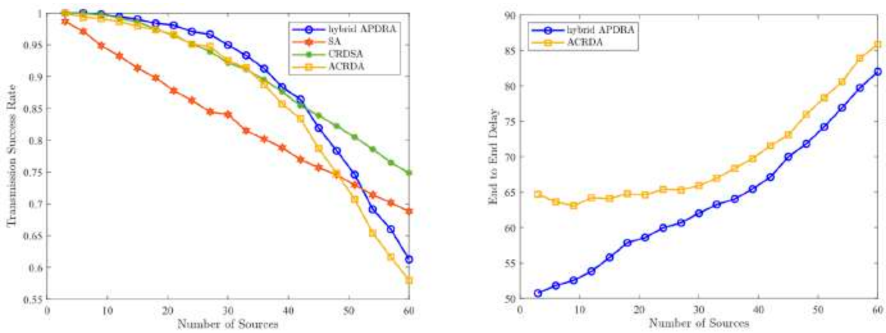
Although the APDRA protocol proposes methods to enforce the sending time of the first data frame and randomize the sending time of the second data frame in order to reduce transmission delay, a detailed design has not been provided. As depicted in Figure 9, when comparing the APDRA protocol to the ACDRA protocol, it is observed that the APDRA protocol achieves the same access delay probability while exhibiting a smaller average access delay time. The maximum reduction in access delay time is 19.1%.

Table 3. Simulation settings for APDRA and TDMA protocols.

Parameter	Value
Time Window Length	200
Monte Carlo Experiment Iterations	10,000
Probability of Generating Data Frame	1
Number of Sensor Nodes	7
Number of Iterations for SIC	10

4.3. Deficiency Analysis of the APDRA Protocol

As previously stated, the APDRA protocol exhibits a decline in performance as the number of sensor nodes in the network increases. When the quantity of sensor nodes within the network surpasses seven, the length of the necessary time window is 128. This constraint suggests that, based on the simulation model design, the distance between the node and the central node should be at least 113.4 km. This requirement is necessary to guarantee the significance of the first data frame and retransmit frame of the sensor node with ID 7. Without meeting this condition, the first frame may arrive while the retransmitted frame is still waiting to be sent, rendering the protocol meaningless in practical applications. Therefore, the APDRA protocol is designed in such a way that when the number of sensor nodes surpasses the maximum capacity of the time window (m), the retransmit wait time frames for sensor nodes with IDs greater than m will no longer adhere to an exponential multiple of the time window. Instead, these frames will be randomly transmitted after the completion of the first frame and before the time window concludes. For these nodes, the APDRA protocol exhibits similarities to the ACRDA protocol. The current protocol should not be considered solely as a pure APDRA protocol but rather as a hybrid protocol that combines practical considerations of both APDRA and ACRDA. We conducted an analysis of the variations in transmission success rate and end-to-end delay in relation to the number of nodes, as depicted in Figure 10. The simulation parameter settings are presented in Table 4.



(**a**) (**b**)

Figure 10. Comparison of the Hybrid APDRA Protocol with Other Protocols in Scenarios Involving a Significant Number of Nodes. (**a**) Transmission Success Rate; (**b**) End-to-End Delay.

As shown in Figure 10a, the TSR of the hybrid APDRA protocol is compared with the synchronous protocol SA, CRDSA, and the asynchronous protocol ACRDA. With the increase in sensor nodes, the TSR advantage of the hybrid APDRA protocol gradually diminishes. When the number of nodes is more than 42, the TSR is lower than that of the synchronous retransmission protocol CRDSA but slightly better than the asynchronous retransmission protocol ACRDA and the traditional protocol SA. Figure 10b presents a comparison of the end-to-end delay between the hybrid APDRA protocol and the ACRDA protocol. It is evident that the performance of the sensor nodes, exceeding the quantity m in the hybrid APDRA protocol, exhibits similarities to that of the hybrid ACRDA protocol. However, the sensor nodes within the quantity m ensure slightly superior performance for the hybrid APDRA protocol compared to ACRDA.

Table 4. Simulation settings among SA, CRDSA, ACRDA and APDRA protocols.

Parameter	Value
Time Window Length	128
Monte Carlo Experiment Iterations	2000
Probability of Generating Data Frame	0.8
Number of Sensor Nodes	0–60
Number of Iterations for SIC	10

5. Conclusions

Due to the long propagation delay characteristic of the acoustic communication channel, the use of contention protocols for channel access in underwater sensor networks would lead to a substantial increase in data retransmission time. On the other hand, using channel allocation protocols avoids access conflicts but sacrifices efficiency in channel access. Therefore, we propose the APDRA protocol, which provides each data frame with two transmission opportunities and designs different retransmission intervals for different nodes in order to minimize the probability of data collisions. Furthermore, by combining the receiver-side SIC method, we can further enhance the quality and efficiency of data transmission. After discussion, we have concluded that in the absence of SIC, a network with n nodes transmitting data frames asynchronously may result in collisions at the receiver. However, disregarding other influential factors, there will always be at least two data frames that can be received correctly. If SIC decoding is implemented, the transmission accuracy of the APDRA protocol can reach 100%, with an iteration decoding count not exceeding $(n-2)$ times. Simulation results demonstrate that the APDRA protocol achieves better communication quality compared to the CRDSA protocol and the asynchronous ACRDA protocol. Additionally, the APDRA protocol has certain advantages over classical acoustic communication protocols.

However, the disadvantages of the APDRA protocol are also evident. Since the retransmission interval of data frames exponentially increases compared to the data frame transmission delay, the APDRA has limitations on the scalability of underwater networks. It performs better for sensor node numbers that are less than or equal to 7. In future work, we will consider pattern-based design principles to further optimize the protocol's retransmission mechanism. To further enhance network efficiency, we will investigate the possibility of sending various data frames and their corresponding RS codes within the designated time window. This approach aims to increase overall throughput while ensuring accurate decoding.

Author Contributions: Conceptualization, J.R.; methodology, Y.W.; software, J.R.; validation, Y.W. and M.Z.; formal analysis, J.R.; investigation, J.R.; data curation, J.R.; writing—original draft preparation, J.R.; writing—review and editing, J.R.; visualization, J.R.; supervision, Y.W. and M.Z.; funding acquisition, Y.W. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 61971472, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA22030101, and the National Key Research and Development Program of China under Grants 2021YFC2800200 and 2021YFC2801200.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, S.; Nazareth, P.; Chandavarkar, B.R. Hop-count Based Routing Protocols for Underwater Acoustic Sensor Networks: A Survey. In Proceedings of the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 3–5 October 2022; pp. 1–6. [CrossRef]
2. Chen, K.; Ma, M.; Cheng, E.; Yuan, F.; Su, W. A Survey on MAC Protocols for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1433–1447. [CrossRef]
3. Riksfjord, H.; Haug, O.T.; Hovem, J.M. Underwater Acoustic Networks—Survey on Communication Challenges with Transmission Simulations. In Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications, Athens, Greece, 18–23 June 2009; pp. 300–305. [CrossRef]
4. Lu, Q.; Shengming, J. A review of routing protocols of underwater acoustic sensor networks from application perspective. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016; pp. 1–6. [CrossRef]
5. Burrowes, G.E.; Khan, J.Y. Investigation of a short-range underwater acoustic communication channel for MAC protocol design. In Proceedings of the 2010 4th International Conference on Signal Processing and Communication Systems, Gold Coast, QLD, Australia, 13–15 December 2010; IEEE: New York City, NY, USA, 2010; pp. 1–8.
6. Makar, A. Simplified Method of Determination of the Sound Speed in Water on the Basis of Temperature Measurements and Salinity Prediction for Shallow Water Bathymetry. *Remote Sens.* **2022**, *14*, 636. [CrossRef]
7. Jiang, S. State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 96–131. [CrossRef]
8. Chen, Y.D.; Lien, C.Y.; Chuang, S.W.; Shih, K.-P. DSSS: A TDMA-based MAC protocol with dynamic slot scheduling strategy for underwater acoustic sensor networks. In Proceedings of the OCEANS 2011 IEEE-Spain, Santander, Spain, 6–9 June 2011; IEEE: New York City, NY, USA, 2011; pp. 1–6.
9. Vázquez-Gallego, F.; Tuset-Peiró, P.; Alonso, L.; Alonso-Zarate, J. Delay and Energy Consumption Analysis of Frame Slotted ALOHA variants for Massive Data Collection in Internet-of-Things Scenarios. *Appl. Sci.* **2020**, *10*, 327. [CrossRef]
10. Vieira, L.F.M.; Kong, J.; Lee, U.; Gerla, M. Analysis of Aloha protocols for underwater acoustic sensor networks. *Ext. Abstr. WUWNet* **2006**, *6*. Available online: https://www.semanticscholar.org/paper/Analysis-of-Aloha-Protocols-for-Underwater-Acoustic-Vieira-Kong/095862fcbbbde81ba0cbb51831276e8c0e70accb (accessed on 24 August 2023).
11. Bharghavan, V.; Deers, A.; Shenker, S.; Zhang, L. MACAW: A media access protocol for wireless LAN's. *ACM SIGCOMM Comput. Commun. Rev.* **1994**, *24*, 212–225. [CrossRef]
12. Yongqiong, Y.; Feng, Z.; Yuhui, F.; Junlu, L. Simulation of time synchronization protocol of underwater acoustic communication network with cluster topology. In Proceedings of the 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xiamen, China, 22–25 October 2017; pp. 1–4. [CrossRef]
13. Lundy, G.M.; Almquist, M.; Oruk, T. Specification, verification and simulation of a wireless LAN protocol: MACAW. In Proceedings of the IEEE Military Communications Conference. Proceedings. MILCOM 98 (Cat. No.98CH36201), Boston, MA, USA, 19–21 October 1998; Volume 2, pp. 565–569.
14. Buratti, C.; Zanella, A.; Verdone, R. Throughput analysis of Slotted Aloha multiple hop systems with Virtual Antenna Arrays. In Proceedings of the 2010 Future Network & Mobile Summit, Florence, Italy, 16–18 June 2010; pp. 1–8.
15. Choudhury, G.; Rappaport, S. Diversity ALOHA—A Random Access Scheme for Satellite Communications. *IEEE Trans. Commun.* **1983**, *31*, 450–457. [CrossRef]
16. Casini, E.; De Gaudenzi, R.; Herrero, O.D.R. Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 1408–1419. [CrossRef]
17. Liva, G. Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA. *IEEE Trans. Commun.* **2011**, *59*, 477–487. [CrossRef]
18. Paolini, E.; Liva, G.; Chiani, M. High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA. In Proceedings of the 2011 IEEE International Conference on Communications (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–6.
19. De Gaudenzi, R.; Herrero, O.D.R.; Acar, G.; Barrabes, E.G. Asynchronous Contention Resolution Diversity ALOHA: Making CRDSA Truly Asynchronous. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 6193–6206. [CrossRef]
20. Kissling, C. Performance Enhancements for Asynchronous Random Access Protocols over Satellite. In Proceedings of the 2011 IEEE International Conference on Communications (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–6.
21. Clazzer, F.; Kissling, C. Enhanced Contention Resolution ALOHA-ECRA. In Proceedings of the 2013 9th International ITG Conference on Systems, Communication and Coding, Munich, Germany, 21–24 January 2013; pp. 1–6.
22. Radosevic, A.; Ahmed, R.; Duman, T.; Proakis, J.; Stojanovic, M. Adaptive OFDM modulation for underwater acoustic communications: Design considerations and experimental results. *IEEE J. Ocean. Eng.* **2014**, *39*, 357–370. [CrossRef]
23. Han, X.; Yin, J.; Du, P.; Zhang, X. Research on Time Domain Equalization of Single Carrier Underwater Acoustic Communication in Shallow Water Environment. Ph.D. Thesis, Harbin Engineering University, Harbin, China, 15 November 2016.