

Article

CM-MAC: A Cluster-Based Hierarchical Scheduling Protocol for Reliable Neighborhood Multicasting in Mobile AUV Swarms

Meiyan Liu ¹  and Guangjie Han ^{2,3,*} ¹ College of Information Science and Engineering, Hohai University, Changzhou 213200, China; myliu@zju.edu.cn² Key Laboratory of Maritime Intelligent Network Information Technology, Ministry of Education, Hohai University, Changzhou 213200, China³ State Key Laboratory of Acoustics and Marine Information, Chinese Academy of Sciences, Beijing 100190, China

* Correspondence: hanguangjie@gmail.com

Abstract

Deploying Autonomous Underwater Vehicle (AUV) swarms is pivotal for oceanographic exploration. However, swarm formation control is constrained by harsh Underwater Acoustic Communication Networks (UACNs), especially in dynamic, non-fully connected topologies. To address high packet collision rates and excessive signaling overhead in existing Medium Access Control (MAC) protocols, this paper proposes a novel Cluster-based Mobile MAC (CM-MAC) protocol for hierarchical clustered AUV networks. Operating under a decentralized three-tier architecture, the protocol coordinates transmissions using local state information. We establish transmission constraints to prevent packet collisions among mobile nodes. Building upon the transmission constraints, genetic algorithms are applied to the transmission scheduling across all layers to optimize the sending sequence and timing, reducing overall latency. Simulation results show that the CM-MAC protocol significantly improves network throughput and reduces information-sharing update intervals when compared to traditional TDMA, pure Aloha, and random-access CM-MAC. This study presents a robust communication framework for coordinating large-scale AUV swarms in complex underwater environments.

Keywords: autonomous underwater vehicles; medium access control; underwater acoustic communication networks; cluster-based network; transmission scheduling



Academic Editor: Douglas O'Shaughnessy

Received: 7 June 2026

Revised: 7 July 2026

Accepted: 14 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In recent years, Autonomous Underwater Vehicle (AUV) swarms have emerged as a transformative technology for ocean exploration. They offer superior capabilities in cooperative target tracking, oceanographic mapping, and distributed sensing compared to single-vehicle systems [1–5]. Effective AUV swarm formation control critically depends on the timely and reliable exchange of state information (e.g., position, velocity, and attitude) among distributed nodes [6–12]. Unlike terrestrial wireless networks, underwater environments rely on acoustic waves for long-range communication [13,14]. However, underwater acoustic channels are characterized by limited bandwidth, high propagation latency, severe multipath effects, and rapid time-varying fading [15]. These physical limitations impose severe constraints on Medium Access Control (MAC) protocols, which define the transmission rules for sharing limited underwater acoustic channel resources [16].

To address these challenges, existing research has extensively explored various MAC mechanisms. These can be broadly categorized into contention-based and non-contention-based protocols. Contention-based MAC protocols do not require pre-allocation of channel resources. Instead, nodes dynamically share the same channel through contention. They are less constrained by the physical-layer limitations of Underwater Acoustic Communication Networks (UACNs) and can better adapt to topology changes. These protocols can be further divided into random-access and handshake-based types. In a random-access protocol, a sender initiates transmission without prior consent from the intended receiver [17–21]. However, due to the hidden terminal problem, random-access protocols suffer from low channel utilization caused by frequent packet collisions. In contrast, handshake-based protocols involve the receiver feeding back contention results to the senders after receiving multiple access requests [22–26]. Nevertheless, the exchange of control packets significantly prolongs waiting times [27], making these protocols unsuitable for high network traffic loads.

Non-contention-based MAC protocols are mostly based on the Time Division Multiple Access (TDMA) framework. They incorporate additional mechanisms, such as scheduling and reservation, to allocate transmission slots to nodes. This allows for more efficient use of limited time resources. Specifically, scheduling-based MAC protocols coordinate node transmissions via predefined rules or centralized controllers. This improves bandwidth utilization by enabling more concurrent transmissions and reducing collisions at the receiver [22]. However, most existing scheduling schemes rely on complete topological knowledge to generate conflict-free schedules [28–32]. Consequently, applying them to large-scale AUV swarms is non-trivial. In non-fully connected networks, the distance between some AUV nodes exceeds the maximum communication range. In multicast communication scenarios for information sharing among adjacent nodes, each node can only obtain the status information of a subset of nodes. This results in incomplete and asymmetric topology information. Therefore, global slot allocation is impractical due to the high signaling overhead and latency required to maintain global consistency. Furthermore, static slot allocation is ill-suited for highly dynamic AUV swarms, where nodes are constantly moving and the network topology changes rapidly.

To support large-scale AUV swarms and reduce global coordination complexity, clustered network architectures have been widely adopted for their scalability and energy efficiency [33–38]. Such architectures decompose large-scale networks into manageable local topologies, enabling decentralized scheduling. However, designing an efficient MAC for these architectures remains challenging. Existing research on non-contention-based MAC protocols for clustered UACNs typically falls into two categories, yet each has limitations when applied to AUV swarms. Table 1 shows the main features and potential improvement directions of typical clustered MAC protocols. Some studies assign orthogonal carrier frequencies or spreading codes to adjacent clusters to eliminate inter-cluster interference [39–41]. While effective, this approach significantly reduces the already scarce bandwidth available to each cluster, limiting network capacity for high-volume formation data. Other works utilize graph coloring or conflict graphs to maximize spatial reuse [42–45]. However, these algorithms often rely on global topology knowledge and complex calculations. In dynamic AUV swarms, the overhead of collecting global information and updating schedules is too high to support real-time control.

Therefore, designing a decentralized scheduling-based MAC protocol that supports high-concurrency multicast interactions for AUV formation control while mitigating inter-cluster interference remains an open challenge. The protocol must leverage the collision-avoidance advantages of scheduling without incurring the prohibitive overhead of global coordination. The specific problems to be solved and corresponding measurable objectives

include: (i) Deriving collision-free neighborhood multicasting transmission constraints applicable to AUVs. This ensures a zero packet collision rate at any receiving node when the distance between two transmitting nodes is known. (ii) Designing a genetic algorithm to jointly optimize the sending order and timing of nodes. This aims to reduce the average update interval for adjacent AUV information sharing and improve network throughput. (iii) Designing a hierarchical cluster-based transmission scheduling scheme. In this scheme, each level of cluster head utilizes local information to schedule the transmission times of nodes within the cluster, thereby reducing inter-cluster and intra-cluster packet collision rates.

Table 1. Comparison of Clustered MAC Protocols for AUV Swarms.

Protocol Category	Key Mechanism	Limitations	Potential Improvements
Orthogonal Allocation [39–41]	Assigns orthogonal frequencies or codes to adjacent clusters to eliminate interference.	Limitation: Significantly reduces the already scarce bandwidth available to each cluster.	Collision-free Multicasting Constraints: Enables spatial reuse without packet collision.
Graph-Based Scheduling [42–45]	Uses conflict graphs to maximize spatial reuse and avoid collisions.	Limitation: Relies on global topology knowledge; high computational complexity and signaling overhead.	Cluster-based Hierarchical Scheduling: Operates based on locally known state information.
Traditional TDMA	Pre-allocates fixed time slots to nodes to ensure collision-free transmission.	Limitation: Static allocation is ill-suited for mobile AUVs; global slot allocation incurs high latency.	GA-Based Optimization: Dynamically optimizes transmission sequences and timing.

Motivated by these challenges, this paper proposes a hierarchical scheduling-based MAC protocol tailored for clustered AUV swarm networks. The main contributions are summarized as follows:

1. **A Novel Hierarchical Scheduling-based MAC Protocol:** We propose a Cluster-based Mobile MAC (CM-MAC) protocol specifically tailored for hierarchical clustered AUV networks. Unlike traditional centralized approaches, CM-MAC operates on a decentralized three-tier architecture where cluster heads coordinate transmissions based on locally known state information. This design significantly reduces the prohibitive signaling overhead associated with global topology collection in dynamic underwater environments.
2. **Theoretical Analysis of Packet Collision Avoidance:** We establish rigorous transmission constraints to guarantee collision-free communication among mobile nodes despite dynamic topology changes. These constraints guarantee collision-free communication between two nodes based solely on their inter-node distance, eliminating the need for the locations of neighboring receivers.
3. **Scheduling Optimization via Genetic Algorithms:** We introduce a genetic algorithm-based scheduling mechanism across all network layers. By optimizing the transmission sequence and timing, the proposed method effectively reduces overall network latency and improves channel utilization.
4. **Performance Verification:** Extensive simulations demonstrate that CM-MAC outperforms traditional protocols, including TDMA, pure Aloha, and random-access CM-MAC. The results confirm that our protocol achieves significant improvements in network throughput and reduces information-sharing update intervals, providing a robust communication framework for large-scale AUV swarms.

2. Hierarchical Clustered Network Model

This paper describes a network scenario in which N AUV nodes migrate as a swarm to a specified target area. During this migration, the nodes share state information with adjacent nodes within their communication range. As the AUV cluster moves across a wide area, the distances between nodes gradually increase. Consequently, some nodes cannot communicate directly with each other due to the maximum communication distance limitation. This type of topology is referred to as a non-fully connected network. In this study, we only consider information exchange between adjacent AUV nodes within such a non-fully connected network. The primary communication requirement is to ensure that each node's state information is transmitted to all adjacent nodes as quickly and successfully as possible. This requirement is equivalent to neighborhood multicasting in a non-fully connected network. The proposed CM-MAC protocol governs the allocation of transmission timing for the nodes. To simplify the analysis, the bit-error ratio caused by channel quality is not considered. Furthermore, time synchronization and positioning errors are ignored.

As illustrated in Figure 1, this paper considers a three-tier hierarchical clustered network consisting of first-level cluster heads, second-level cluster heads, and surrounding nodes. The packet lengths transmitted by the first-level cluster head, the second-level cluster heads, and the surrounding nodes are denoted as L_1 , L_2 , and L , respectively. The first-level cluster head is represented as n_0 , while the second-level cluster heads are denoted as n_k , where $k \in \mathbb{Z}^+$. The set of second-level cluster heads managed by the first-level cluster head n_0 is defined as $S_0 = \{n_1, n_2, \dots, n_{|S_0|}\}$. Similarly, the set of surrounding nodes managed by the second-level cluster head n_k ($k \in \{1, \dots, |S_0|\}$) is defined as $S_k = \{m_1^{n_k}, m_2^{n_k}, \dots, m_{|S_k|}^{n_k}\}$. The main notations used in this paper are summarized in Table 2.

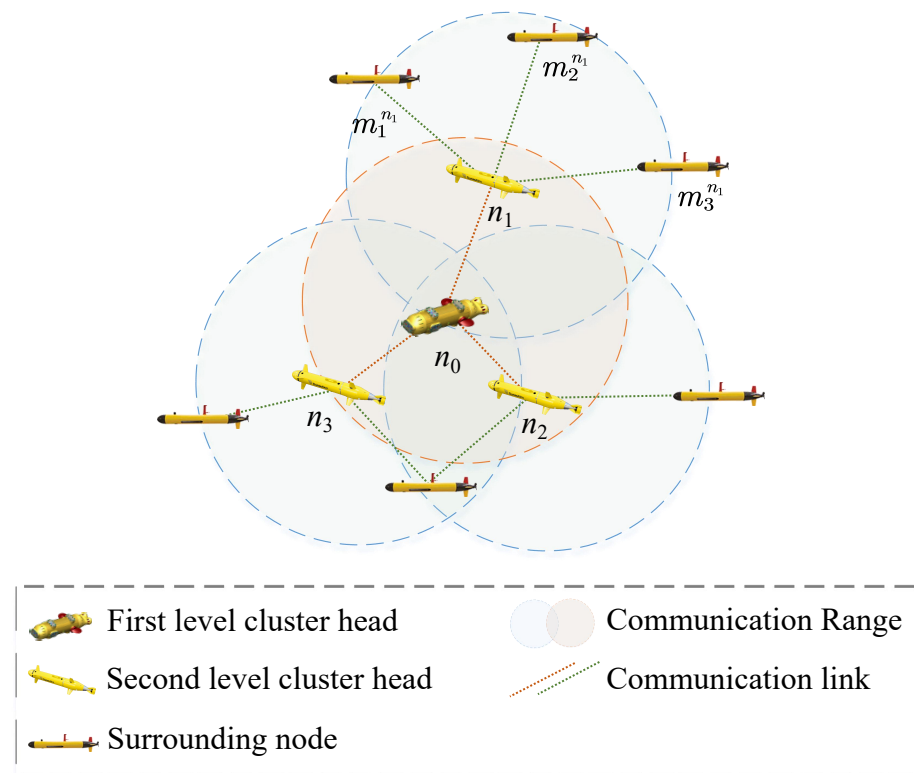


Figure 1. Three-tier Hierarchical clustered network structure.

Table 2. Notations.

Notation	Definition
L_1	Packet length transmitted by the first-level cluster head
L_2	Packet length transmitted by the second-level cluster head
L	Packet length transmitted by the surrounding node
n_0	First-level cluster head
n_k	Second-level cluster head
$\mathcal{S}_0 = \{n_1, n_2, \dots, n_{ \mathcal{S}_0 }\}$	The set of second-level cluster heads managed by n_0
$\mathcal{S}_k = \{m_1^{n_k}, m_2^{n_k}, \dots, m_{ \mathcal{S}_k }^{n_k}\}$	The set of surrounding nodes managed by n_k
ω	Transmission rate
c	Propagation speed
d	Transmission distance
$d_{i,q}^{t_i}$	Distance between node i and node q at time t_i
$v_{\max}$	Maximum AUV speed
d_0	Maximum communication distance
$L_{\max}/\omega$	Maximum packet transmission duration
t_i	Sending time of node i
M	Population size
G_m	Maximum population number
p_x	Probability of crossover
p_m	Probability of mutation
u_t	Inheritance ratio

3. Overview of the CM-MAC Protocol

We assume that the AUV swarm network transitions from a fully connected state to a non-fully connected state during its movement. During the fully connected stage, the transmission times of the nodes are determined by existing global scheduling protocols, such as the AB-MAC protocol [46]. When a control node makes a scheduling decision, it must calculate the distance to the scheduled node at the designated transmission time. This calculation is essential to verify whether the node remains within the control node's communication range. If any node exceeds this range at the specified time, the network transitions to a non-fully connected state. In this case, the control node broadcasts a state transition packet. This packet instructs the pre-designated first-level cluster head to manage the scheduling using the CM-MAC protocol. The main workflow of the CM-MAC protocol is detailed as follows.

3.1. First-Level Cluster Head Schedules Transmissions for Second-Level Cluster Heads

Upon receiving the state transition packet, the first-level cluster head calculates the distances to other nodes based on their current state information. It then identifies all single-hop adjacent nodes and schedules their transmission times. If an adjacent node falls outside the communication range at the designated time, the first-level cluster head cancels the time assignment for that node. After this process, any node with a successfully scheduled transmission time is designated as a second-level cluster head. The broadcast packet from the first-level cluster head contains its own state information, along with the transmission times and state information of all second-level cluster heads.

The critical challenge in this process is to select and arrange the second-level cluster heads. They must remain within communication range during their transmission times, while allowing as many nodes as possible to transmit as early as possible. Once the selection is complete, the final set of second-level cluster heads is determined. We outline the specific scheduling algorithm in Section 6.1.

3.2. Second-Level Cluster Head Schedules Transmissions for Surrounding Nodes in Its Sub-Cluster

A second-level cluster head gathers the state information and transmission times of all other second-level cluster heads by receiving broadcast packets from the first-level cluster head. Using this information, it identifies the surrounding nodes in its sub-cluster. These are defined as all nodes within its communication range that are not covered by any other second-level cluster head with an earlier transmission time. After identifying these surrounding nodes, the second-level cluster head schedules their transmission times accordingly. The broadcast packet sent by the second-level cluster head contains its own state information, as well as the scheduled transmission times and state information for all surrounding nodes in its sub-cluster. The specific scheduling algorithm is detailed in Section 6.2.

3.3. Initiation of the Next Scheduling Cycle

After receiving broadcast packets from all second-level cluster heads, the first-level cluster head gathers the transmission times of all surrounding nodes. It then calculates its own transmission time for the next round. This time is set to the latest transmission time plus the maximum propagation delay. Afterward, the network periodically executes the CM-MAC protocol as described above.

Unlike centralized protocols that require global network state information, the proposed protocol leverages a hierarchical clustering architecture to reduce signaling overhead. The Cluster head performs deterministic scheduling based solely on local information gathered during the routine MAC operations. This design eliminates the need for network-wide broadcasting or continuous real-time synchronization, making the protocol robust against the severe delays and fading characteristic of underwater acoustic channels.

4. Collision-Free Neighborhood Multicast Constraints for Mobile Nodes

Collision-free neighborhood multicast communication ensures that all adjacent neighbors successfully receive a node's transmitted packets. To achieve this, the packets sent by any transmitter must not collide with those from interfering nodes at neighboring receivers. In the following, we analyze the transmission constraints that an interfering node must satisfy to avoid disrupting the primary transmitter. This analysis is particularly relevant when the location of the adjacent receiver is unknown.

Assume that node j transmits a packet of length L_j at time t_j , while node i transmits a packet of length L_i at time t_i . Let ω represent the transmission rate, and c denote the speed of sound in water. To prevent collisions between the packets from nodes j and i at an adjacent receiver q , their transmission times must satisfy

$$t_i + \frac{d_{i,q}^{t_i}}{c} \geq t_j + \frac{d_{j,q}^{t_j}}{c} + \frac{L_j}{\omega}, \quad (1)$$

or

$$t_j + \frac{d_{j,q}^{t_j}}{c} \geq t_i + \frac{d_{i,q}^{t_i}}{c} + \frac{L_i}{\omega}. \quad (2)$$

These conditions can be rewritten as

$$t_i - t_j \geq \frac{d_{j,q}^{t_j} - d_{i,q}^{t_i}}{c} + \frac{L_j}{\omega}, \quad (3)$$

or

$$t_j - t_i \geq \frac{d_{i,q}^{t_i} - d_{j,q}^{t_j}}{c} + \frac{L_i}{\omega}, \quad (4)$$

where $d_{i,q}^{t_i}$ and $d_{j,q}^{t_j}$ represent the distances from nodes i and j to receiver q at their respective transmission times t_i and t_j . Here, we assume that the distance variation between the transmitter and the receiver is negligible during the packet transmission period (i.e., from the start of transmission until the packet is fully received). If the transmission time t_i of node i is known, but t_j of node j is unknown, the distance terms $d_{j,q}^{t_j} - d_{i,q}^{t_i}$ and $d_{i,q}^{t_i} - d_{j,q}^{t_j}$ can be expanded as $d_{j,q}^{t_j} - d_{i,q}^{t_i} + (d_{j,q}^{t_j} - d_{j,q}^{t_i})$ and $d_{i,q}^{t_i} - d_{j,q}^{t_j} + (d_{i,q}^{t_i} - d_{i,q}^{t_j})$, respectively. If AUVs in the network move with uniform linear motion, the distance variation is bounded by

$$|d_{j,q}^{t_j} - d_{j,q}^{t_i}| \leq 2v_{\max}|t_j - t_i|, \quad (5)$$

where $v_{\max}$ is the maximum velocity of an AUV, and $2v_{\max}$ represents the maximum relative velocity between two AUVs moving in opposite directions. According to Equations (3) and (4), a potential packet collision between node i and node j occurs only if the difference in their transmission times satisfies

$$|t_j - t_i| \leq \frac{d_0}{c} + \frac{L_{\max}}{\omega}, \quad (6)$$

where d_0 is the maximum communication distance, and $L_{\max}/\omega$ is the maximum packet transmission duration sent by nodes in the network. Substituting this bound into Equation (3) yields

$$t_i - t_j \geq \frac{d_{j,q}^{t_i} - d_{i,q}^{t_i}}{c} + \frac{L_j}{\omega} + \frac{2v_{\max}}{c} \left(\frac{d_0}{c} + \frac{L_{\max}}{\omega} \right). \quad (7)$$

Similarly, Equation (4) can be derived as

$$t_j - t_i \geq \frac{d_{i,q}^{t_i} - d_{j,q}^{t_i}}{c} + \frac{L_i}{\omega} + \frac{2v_{\max}}{c} \left(\frac{d_0}{c} + \frac{L_{\max}}{\omega} \right). \quad (8)$$

Let $t_g = \frac{2v_{\max}}{c} \left(\frac{d_0}{c} + \frac{L_{\max}}{\omega} \right)$. Based on Equations (7) and (8), nodes i and j are guaranteed to be collision-free at any receiver if the following relaxed conditions are met:

$$t_i - t_j \geq \frac{\max_{q \in \mathcal{N}} \{d_{j,q}^{t_i} - d_{i,q}^{t_i}\}}{c} + \frac{L_j}{\omega} + t_g, \quad (9)$$

$$t_j - t_i \geq \frac{\max_{q \in \mathcal{N}} \{d_{i,q}^{t_i} - d_{j,q}^{t_i}\}}{c} + \frac{L_i}{\omega} + t_g, \quad (10)$$

where $\mathcal{N}$ represents the set of all adjacent receivers. According to the triangle inequality, $\max_{q \in \mathcal{N}} \{d_{j,q}^{t_i} - d_{i,q}^{t_i}\} = \max_{q \in \mathcal{N}} \{d_{i,q}^{t_i} - d_{j,q}^{t_i}\} = d_{i,j}^{t_i}$, where $d_{i,j}^{t_i}$ is the distance between node i and node j at time t_i . Therefore, the collision avoidance transmission constraints for mobile node pairs can be summarized as

$$t_j - t_i \notin \left[-\frac{d_{i,j}^{t_i}}{c} - \frac{L_j}{\omega} - t_g, \frac{d_{i,j}^{t_i}}{c} + \frac{L_i}{\omega} + t_g \right]. \quad (11)$$

In this formulation, the distance parameters are estimated based on the known transmission time. The estimation error is compensated for by the protection time interval t_g .

5. Collision Avoidance Constraints in Hierarchical Clusters

In our hierarchical clustered network, packet collisions can be categorized into four distinct types: (1) collisions between second-level cluster heads, (2) collisions between second-level cluster heads and surrounding nodes in other sub-clusters, (3) collisions between surrounding nodes within the same sub-cluster, and (4) collisions between surround-

ing nodes across different sub-clusters. This section defines the transmission constraints designed to prevent these collisions.

5.1. Transmission Constraints Between Second-Level Cluster Heads

As illustrated in Figure 2, the first-level cluster head n_0 schedules the transmission times for the second-level cluster heads. During this process, it is essential to prevent mutual interference in their neighborhood multicast communications. Let t_0 denote the transmission time of the first-level cluster head. The sequence in which the second-level cluster heads determine their transmission times is represented by $\alpha = \{\alpha_1, \dots, \alpha_{|\mathcal{S}_0|}\}$. Here, α is a complete permutation of nodes n_k ($k = \{1, \dots, |\mathcal{S}_0|\}$). Note that α does not represent the final transmission order.

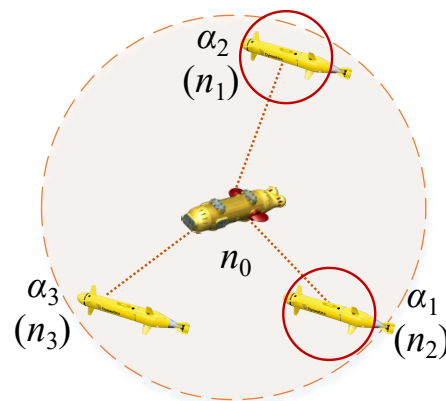


Figure 2. Transmission constraints between second-level cluster heads.

All second-level cluster heads must transmit only after receiving the broadcast packet from n_0 . This condition is expressed as

$$t_{\alpha_i} \geq t_0 + \frac{d_{n_0, \alpha_i}^{t_0}}{c} + \frac{L_1}{\omega}, \quad \forall i \in \{1, \dots, |\mathcal{S}_0|\}, \quad (12)$$

where $d_{n_0, \alpha_i}^{t_0}$ is the distance between node α_i and node n_0 at time t_0 . Additionally, L_1/ω indicates the transmission duration of the broadcast packet from the first-level cluster head.

In addition, to prevent packet collisions between node α_i ($i \in \{2, \dots, |\mathcal{S}_0|\}$) and any previously scheduled node $\alpha_{i'}$ ($i' \in \{1, \dots, |\mathcal{S}_0|\}$ and $i' < i$), the transmission time t_{α_i} must satisfy the following constraint once $t_{\alpha_{i'}}$ is determined:

$$t_{\alpha_i} - t_{\alpha_{i'}} \notin \left[-\frac{d_{\alpha_{i'}, \alpha_i}^{t_{\alpha_{i'}}}}{c} - \frac{L_2}{\omega} - t_g, \frac{d_{\alpha_{i'}, \alpha_i}^{t_{\alpha_{i'}}}}{c} + \frac{L_2}{\omega} + t_g \right]. \quad (13)$$

5.2. Transmission Constraints Between Surrounding Nodes and Second-Level Cluster Heads

As illustrated in Figure 3, when a second-level cluster head schedules the transmission times for surrounding nodes in its sub-cluster, it must ensure that the surrounding nodes in its sub-cluster and the second-level cluster heads from other sub-clusters do not interfere with each other's neighborhood multicast communication. Before initiating transmission, a second-level cluster head can obtain the transmission times and locations of other second-level cluster heads via broadcast packets from the first-level cluster head. Specifically, when second-level cluster head n_k ($k \in \{1, \dots, |\mathcal{S}_0|\}$) allocates transmission times for its surrounding nodes $m_p^{n_k}$ ($p = \{1, \dots, |\mathcal{S}_k|\}$), it already knows the location of node n_j ($j \in \{1, \dots, |\mathcal{S}_0|\}$ and $j \neq k$) and the transmission time t_{n_j} .

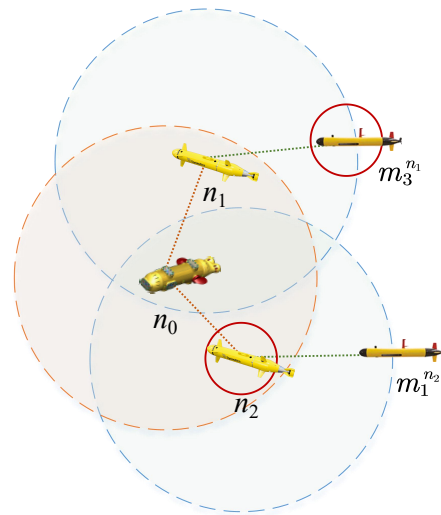


Figure 3. Transmission constraints between surrounding nodes and second-level cluster heads.

To avoid packet collision between node $m_p^{n_k}$ and node n_j , the transmission time $t_{m_p^{n_k}}$ must satisfy

$$t_{m_p^{n_k}} - t_{n_j} \notin \left[-\frac{d_{n_j, m_p^{n_k}}^{t_{n_j}}}{c} - \frac{L}{\omega} - t_g, \frac{d_{n_j, m_p^{n_k}}^{t_{n_j}}}{c} + \frac{L_2}{\omega} + t_g \right]. \quad (14)$$

5.3. Transmission Constraints Between Intra-Cluster Surrounding Nodes

As illustrated in Figure 4, when a second-level cluster head schedules the sending times for its surrounding nodes, it must prevent mutual interference in their neighborhood multicast communications. Let t_{n_k} denote the transmission time of the second-level cluster head n_k . The sequence in which the surrounding nodes determine their transmission times is represented by $\beta = \{\beta_1, \dots, \beta_{|\mathcal{S}_k|}\}$. Here, β is a complete permutation of the nodes $m_p^{n_k}$ ($p = \{1, \dots, |\mathcal{S}_k|\}$). Note that β does not represent the final transmission order.

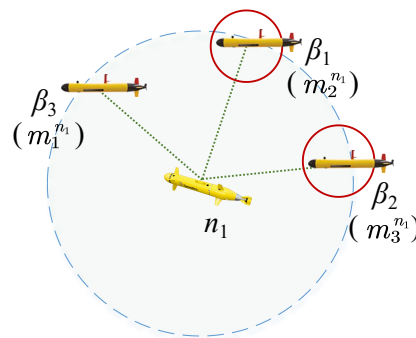


Figure 4. Transmission constraints between intra-cluster surrounding nodes.

All surrounding nodes within the sub-cluster must transmit only after receiving the broadcast packet from the second-level cluster head n_k . This condition is expressed as

$$t_{\beta_i} \geq t_{n_k} + d_{n_k, \beta_i}^{t_{n_k}}/c + L_2/\omega, \quad \forall i \in \{1, \dots, |\mathcal{S}_k|\}, \quad (15)$$

where $d_{n_k, \beta_i}^{t_{n_k}}$ is the distance between node β_i and node n_k while n_k is transmitting. The term L_2/ω denotes the transmission duration of the packet sent by the second-level cluster head.

To avoid packet collisions between node β_i ($i \in \{2, \dots, |\mathcal{S}_k|\}$) and any previously scheduled node $\beta_{i'}$ ($i' \in \{1, \dots, |\mathcal{S}_k|\}$ and $i' < i$), the transmission time t_{β_i} must satisfy the following constraint once $t_{\beta_{i'}}$ is determined:

$$t_{\beta_i} - t_{\beta_{i'}} \notin \left[-\frac{d_{\beta_{i'}, \beta_i}^{t_{\beta_{i'}}}}{c} - \frac{L}{\omega} - t_g, \frac{d_{\beta_{i'}, \beta_i}^{t_{\beta_{i'}}}}{c} + \frac{L}{\omega} + t_g \right]. \quad (16)$$

5.4. Transmission Constraints Between Inter-Cluster Surrounding Nodes

Equation (13) indicates that a second-level cluster head with a later transmission time can receive packets from neighboring second-level cluster heads before transmitting. The obtained information includes the locations and transmission times of the neighboring second-level cluster heads and their surrounding nodes.

As illustrated in Figure 5, consider two second-level cluster heads n_j and n_k ($j, k \in \{1, \dots, |\mathcal{S}_0|\}$ and $j \neq k$). Assume that the distance satisfies $d_{n_k, n_j}^{t_{n_j}} \leq d_0$ and $t_{n_j} < t_{n_k}$. In this case, when second-level cluster head n_k allocates a sending time for its surrounding node $m_p^{n_k}$ ($p \in \{1, \dots, |\mathcal{S}_k|\}$), it already knows the location and transmission time $t_{m_q^{n_j}}$ of the surrounding node $m_q^{n_j}$ ($q \in \{1, \dots, |\mathcal{S}_j|\}$) in the neighboring sub-cluster.

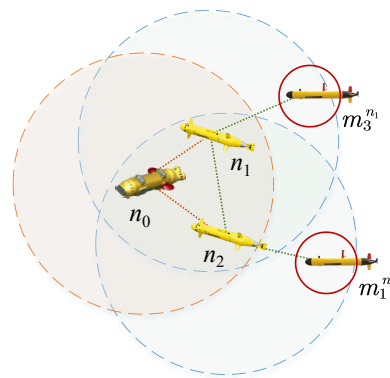


Figure 5. Transmission constraints between inter-cluster surrounding nodes.

To prevent packet collisions between node $m_p^{n_k}$ and node $m_q^{n_j}$, the sending time $t_{m_p^{n_k}}$ must satisfy

$$t_{m_p^{n_k}} - t_{m_q^{n_j}} \notin \left[-\frac{d_{m_q^{n_j}, m_p^{n_k}}^{t_{m_q^{n_j}}}}{c} - \frac{L}{\omega} - t_g, \frac{d_{m_q^{n_j}, m_p^{n_k}}^{t_{m_q^{n_j}}}}{c} + \frac{L}{\omega} + t_g \right]. \quad (17)$$

It should be noted that if two second-level cluster head nodes are outside each other's communication range, they cannot obtain information from each other's broadcast packets. Consequently, packet collisions among their surrounding nodes cannot be avoided.

6. Transmission Scheduling Optimization Algorithm

The principle of transmission scheduling is to ensure that a node with a newly assigned sending time does not cause a packet collision with any node that has an established transmission time. Two aspects of scheduling are mainly considered. One is to identify the set of second-level cluster heads and assign a sending time to each. Second, each second-level cluster head is responsible for allocating sending times to the surrounding nodes within its sub-cluster. This section outlines the specific scheduling algorithms.

6.1. Transmission Scheduling Optimization for Second-Level Cluster Heads

Assume that the first-level cluster head n_0 has m single-hop adjacent nodes at its transmission time t_0 . We denote the sequence of these nodes as M . The first-level cluster head is responsible for scheduling transmissions for the second-level cluster heads. This task involves selecting all second-level cluster heads from M and allocating transmission times to them to ensure collision-free neighborhood multicast communication.

To solve this combinatorial optimization problem, an improved genetic algorithm is employed. As illustrated in Algorithm 1, the specific algorithm for selecting second-level cluster heads and allocating sending times comprises five steps: chromosome coding, selection of excellent individuals, crossover and recombination, mutation, and an elite strategy.

Algorithm 1 GA-based Allocation Scheme

```

1: Input: local state information
2: Initialization:  $g = 1, P = 0$ 
3: Randomly generate  $M$  transmission sequences according to the rules of Chromosomal coding and put them into  $P$ 
4: repeat
5:    $g = g + 1, P_{\text{new}} = 0$ 
6:   for  $i = 1 : 2 : M - 1$  do
7:     Excellent individuals selection: Select two transmission sequences ( $s_1, s_2$ ) from  $P$ 
8:     Crossover and recombination: Cross  $s_1$  and  $s_2$  with a probability of  $p_x$ , and recombine them to get two legal new transmission sequences  $s_3, s_4$ 
9:     Mutation: Mutate on  $s_3$  and  $s_4$  with a probability of  $p_m$  respectively, and get two new transmission sequences  $s_5, s_6$ 
10:    Replace the  $i$ -th row of  $P_{\text{new}}$  with  $s_5$ 
11:    Replace the  $(i + 1)$ -th row of  $P_{\text{new}}$  with  $s_6$ 
12:   end for
13:   Elite strategy: Extract the first  $u_t\%$  excellent individuals of  $P$  and the first  $(1 - u_t\%)$  excellent individuals of  $P_{\text{new}}$  to form the next generation
14:   Update  $P$  to the next generation
15: until  $g = G_m$ 
16: Output: the transmission sequence with minimum latest sending time in the last generation  $P$ 

```

(1) Chromosomal coding: Different transmission sequences are represented as distinct chromosomes, i.e., individuals. A full permutation of the adjacent node sequence M describes all possible transmission sequences, expressed as $\alpha' = \{\alpha'_1, \dots, \alpha'_m\}$. No node index is repeated within a single chromosome. A population consists of M individuals.

(2) Excellent individuals selection: The fitness function of an individual is defined as

$$F(l) = \frac{1}{1 + f(l)}, \quad (18)$$

where $f(l)$ is the cost function value of the l -th individual, defined as the normalized scheduling span of all second-level cluster heads. Formally, given a set of $|S_0|$ second-level cluster heads with their respective assigned transmission times $\{t_{\alpha'_1}, \dots, t_{\alpha'_m}\}$, the normalized scheduling span is formulated as

$$f = \frac{\max_{1 \leq i \leq m} (t_{\alpha'_i}) - \min_{1 \leq i \leq m} (t_{\alpha'_i})}{N_{\text{within}} + \epsilon} \quad (i = 1, 2, \dots, m), \quad (19)$$

where N_{within} is the number of successfully assigned nodes and ϵ is set to a very small positive value to avoid the denominator being 0. Minimizing the normalized scheduling

span ensures that the second-level cluster heads complete their data transmission tasks as compactly as possible.

For an individual $\alpha' = \{\alpha'_1, \dots, \alpha'_m\}$, the sending time $t_{\alpha'_i}$ is calculated sequentially. Each calculated time must satisfy the transmission constraints given by Equations (12) and (13). After calculating $t_{\alpha'_i}$, the algorithm verifies whether the distance between node α'_i and the first-level cluster head n_0 is still less than the maximum communication distance. If $d_{n_0, \alpha'_i}^{t_{\alpha'_i}} \geq d_0$, node α'_i fails to meet the criteria for becoming a second-level cluster head. When calculating the fitness function value, N_{within} equals the number of valid second-level cluster heads within that individual. Algorithm 2 presents the procedure for evaluating the fitness function.

Algorithm 2 Fitness Function Calculation Process

Require: Chromosome C (Sequence of m nodes)

Ensure: Fitness Function Value F

```

1: Initialize Schedule Array  $T_{\text{send}}$ 
   /*Outer Loop: Iterate through all nodes*/
2: for  $j \leftarrow 1$  to  $m$  do
3:   Construct earliest transmission time  $t_{\text{earliest}}$  based on network topology
4:    $\mathcal{I}_j \leftarrow [-\infty, t_{\text{earliest}}]$ 
   /*Inner Loop: Check against previous nodes*/
5:   for  $k \leftarrow 1$  to  $j - 1$  do
6:     Calculate Conflict Interval  $\mathcal{I}_{j,k}$  between node  $j$  and  $k$ 
7:      $\mathcal{I}_j \leftarrow \mathcal{I}_j \cup \mathcal{I}_{j,k}$ 
8:   end for
9:   Sort conflict intervals in  $\mathcal{I}_j$  by their start time
10:   $T_{\text{send}}[j] \leftarrow$  Find earliest available transmission time in  $\mathcal{I}_j$  Verify whether node  $j$  is
      still within the transmission range of the first-level cluster node.
11: end for
12:  $N_{\text{within}} \leftarrow$  the number of nodes within transmission range.
13:  $f \leftarrow (\max(T_{\text{send}}) - \min(T_{\text{send}})) / (N_{\text{within}} + \epsilon)$ 
14:  $F \leftarrow 1 / (1 + f)$ 
15: return  $F$ 

```

In a population, the probability of selecting the l -th individual is $p(l) = F(l) / \sum_{k=1}^M F(k)$, where M is the total number of individuals in the population. The corresponding cumulative probability can be expressed as $p_c(l) = \sum_{k=1}^l p(k)$. Generate a random number $r \in [0, 1]$, if the l -th individual satisfies $p_c(l) \geq r$ and $p_c(l-1) < r$, this individual will be selected. Two individuals are chosen from the population each time for the next crossover and mutation operations.

(3) Crossover and recombination: Every two selected individuals will undergo crossover with a probability of p_x . First, two random integers $c_1, c_2 \in [1, N]$ are generated. Then, the sequence between c_1 and c_2 is swapped between the two individuals.

However, this operation may introduce duplicate node indices, violating the chromosome coding rules. To correct invalid codes, a recombination operation is performed on the crossed individuals, as outlined in Algorithm 3.

(4) Mutation: Two random integers $m_1, m_2 \in [1, N]$ are generated. The sequence between the m_1 and m_2 indices of a given individual is then reversed.

(5) Elite strategy: To retain exceptional individuals, those with the highest fitness values are directly inherited into the next generation without undergoing crossover or mutation. Specifically, the top $u_t\%$ of individuals from population P , along with the top $1 - u_t\%$ from population P_{new} , form the next generation. After that, update P to the next generation population.

Algorithm 3 Recombination procedure

```

1: Input: offspring transmission sequence 1, 2
   /*Correct transmission node IDs before  $c_1$ */
2: for  $i = 1 : c_1 - 1$  do
3:   repeat
4:      $h = \text{find}(\text{offspring } 1(c_1 : c_2) = \text{offspring } 1(i))$ 
5:      $y = \text{offspring } 2(c_1 - 1 + h)$ 
6:      $\text{offspring } 1(i) = y$ 
7:   until isempty(find(offspring 1( $c_1 : c_2$ )=offspring 1( $i$ )))
8:   repeat
9:      $h = \text{find}(\text{offspring } 2(c_1 : c_2) = \text{offspring } 2(i))$ 
10:     $y = \text{offspring } 1(c_1 - 1 + h)$ 
11:     $\text{offspring } 2(i) = y$ 
12:  until isempty(find(offspring 2( $c_1 : c_2$ )=offspring 2( $i$ )))
13: end for
   /*Correct transmission node IDs after  $c_2$ */
14: for  $i = c_2 + 1 : N$  do
15:   Perform the same procedure as line 3-12
16: end for
17: Output: offspring transmission sequence 1*, 2*

```

Operations (2) through (5) are repeated until the maximum number of generations G_m is reached. In the final generation, the individual with the highest fitness value represents the optimized transmission order. Note that during actual scheduling, only nodes within the communication range and their corresponding transmission times are retained based on the optimized output.

6.2. Transmission Scheduling Optimization for Surrounding Nodes

According to the CM-MAC protocol, the second-level cluster head determines the set of surrounding nodes in its sub-cluster before making scheduling decisions. Assume that the sub-cluster of node n_k contains n surrounding nodes. We denote the sequence of these nodes as N . The second-level cluster head performs transmission scheduling for these surrounding nodes. The objective is to allocate transmission times to all nodes in N to ensure collision-free neighborhood multicast communication.

The transmission scheduling process resembles that in Algorithm 1, but it differs in chromosome coding and the selection of excellent individuals. The specific differences in these two steps are detailed below.

(1) Chromosome coding: A full permutation of the surrounding node sequence N represents all possible transmission sequences, expressed as $\beta' = \{\beta'_1, \dots, \beta'_n\}$. No node index is repeated within a single chromosome. A population consists of M individuals.

(2) Selection of excellent individuals: For an individual $\beta' = \{\beta'_1, \dots, \beta'_n\}$, the sending time $t_{\beta'_i}$ will be calculated sequentially. Each calculated time must adhere to the transmission constraints outlined in Equations (14)–(17).

6.3. Computational Complexity and Scalability

The complexity of evaluating a single individual (chromosome) in the proposed Genetic Algorithm is primarily determined by the collision-free scheduling mechanism. For a cluster containing m nodes to be assigned, the algorithm requires $O(m)$ iterations to assign transmission times sequentially. In each iteration, it must evaluate potential collisions with all previously scheduled nodes, resulting in $O(m^2)$ pairwise distance and time-window calculations. Furthermore, sorting these conflict intervals requires $O(m^2 \log m)$ operations. Therefore, the computational complexity of a single fitness evaluation is $O(m^2 \log m)$. Given a population size of M and a maximum of G_m generations, the overall time complexity

of the proposed GA is $O(G_m \times M \times m^2 \log m)$. This polynomial complexity demonstrates that the algorithm remains computationally tractable and scalable as the AUV swarm size increases.

7. Simulation

To evaluate the effectiveness of transmission order optimization, we propose a random-access variant of the CM-MAC protocol, termed randCM-MAC, as a benchmark. Unlike the proposed scheme, randCM-MAC assigns transmission orders randomly rather than optimizing them. We compare our protocol against three typical schemes: the conventional TDMA protocol, the Aloha protocol, and randCM-MAC. In conventional TDMA, each node is assigned a time slot equal to the packet transmission duration plus the propagation delay corresponding to the maximum transmission range. Aloha serves as a representative contention-based MAC protocol, where nodes attempt transmission whenever the channel is idle. Both TDMA and Aloha are effectively utilized in mobile UACNs for neighborhood multicast communications. Table 3 summarizes the main characteristics of CM-MAC and the benchmark protocols.

Table 3. Comparison of Main Characteristics of CM-MAC and Benchmark Protocols.

Protocol	Key Mechanism	Limitations & Advantage
Aloha	Nodes attempt to transmit when the channel is sensed idle.	Limitation: High collision probability and idle channel waste.
Conventional TDMA	Pre-allocates fixed time slots (packet duration + max propagation delay) to each node.	Limitation: High latency and poor bandwidth utilization for mobile AUVs.
randCM-MAC	A variant of CM-MAC where the transmission order is assigned randomly instead of being optimized.	Limitation: Sub-optimal scheduling.
CM-MAC (Proposed)	Employs collision-free multicasting constraints, cluster-based hierarchical scheduling, and Genetic Algorithms.	Advantage: Reducing packet collisions and improving channel utilization based on local information.

7.1. Simulation Setup

All simulations are conducted using MATLAB R2020b. A square area is defined within a horizontal plane at a depth of 6 km below the sea surface. This area is divided into several square sub-regions, each with a side length of $d_0/\sqrt{5}$. Each sub-region is assigned to an AUV node as its target area. This specific side length ensures that each AUV maintains connectivity with others within its communication range while operating in its designated area. We assume all nodes dive from the same point on the sea surface to random locations within their assigned target sub-regions. The nodes follow predefined straight-line trajectories with uniform linear motion. Their speeds are controlled to keep all nodes within the same horizontal plane, with a maximum speed of 2.5 m/s. To simulate communication in a non-fully connected mobile UACN, the simulation starts when the network becomes non-fully connected and ends when any node moves beyond the two-hop communication range of the first-level cluster head.

In this simulation, we utilize the commercial spherical transducer specifications of D/17 from NEPTUNE SONAR (Neptune Sonar. Spherical Projectors. Available online: <https://www.neptune-sonar.co.uk/downloads/5c197ce2-7ea9-4dd5-b506-cbed9d540249/D-17.pdf> (accessed on 10 February 2025)). For this transducer, the transmission central frequency is 17 kHz, the bandwidth is about 3 kHz, the transmit sensitivity is 148 dB re 1 $\mu\text{Pa/V}$ @ 1 m, and the maximum continuous transmit voltage is 190 Vrms @ 100% (use

40 Vrms to reduce energy consumption). From these parameters, we can calculate that the transmitting source level is 180 dB, and the transmission power is 8.5 W. With a common spectral efficiency of 0.5 (LinkQuest. LinkQuest Underwater Acoustic Modems UWM10000 Specifications. Available online: <http://www.link-quest.com/html/uwm10000.htm> (accessed on 10 February 2025)), the data rate is 1500 bps. According to the Wenz curve [47], derived from measured environmental noise data, the environmental noise power spectral density at sea state 2 is about 43 dB/Hz. Assuming that receivers utilize a conventional Recursive Least Squares (RLS) algorithm, the SNR threshold is set at 17 dB for a bit-error ratio of 10^{-4} [48]. The maximum transmission range is set to 8 km. The lengths of packets sent by first-level cluster heads, second-level cluster heads, and surrounding nodes are 4000 bits, 4000 bits, and 2000 bits, respectively.

In the Genetic Algorithm (GA)-based approach, the population size (M), maximum population number (G_m), probability of crossover (p_x), probability of mutation (p_m), and inheritance ratio (u_i) are configured as follows: 200, 1000, 0.97, 0.05, and 5, respectively. The final results are obtained by averaging the outcomes of 20 repeated simulations, where the positions of all nodes' destinations are randomly reset in each simulation.

7.2. Performance Metrics

The performance metrics used to evaluate simulated protocols include network throughput, average update interval, average packet collision ratio, and category-specific packet collision ratios.

(1) Network Throughput: Network throughput is defined as the average number of packets successfully received by a node per second. Consider a network with N nodes. Let N_s be the total number of successfully received packets during the simulation time t_{run} . The network throughput S is expressed as

$$S = \frac{N_s}{t_{\text{run}}N}, \quad (20)$$

where t_{run} is measured in seconds, and S is expressed in packets/s.

(2) Average Update Interval: The average update interval for state information is defined as the time gap between two consecutive instances when a node receives packets from a specific neighbor. Let t_{ij}^k denote the time when node i receives a packet from node j for the k -th time. The average update interval $\bar{D}$ is formulated as

$$\bar{D} = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j \in N, j \neq i} \frac{\sum_{k=2}^{n_{ij}} (t_{ij}^k - t_{ij}^{k-1})}{n_{ij} - 1}, \quad (21)$$

where n_{ij} is the total number of packets successfully received by node i from node j , and N represents the set of all nodes. A shorter update interval is critical for the effectiveness of online swarm control algorithms.

(3) Average Packet Collision Ratio: A packet collision occurs when multiple packet signals overlap at the receiver. Let N_s represent the total number of successfully received packets, and N_f represent the total number of collided packets. The average packet collision ratio R is expressed as

$$R = \frac{N_s}{N_s + N_f}. \quad (22)$$

(4) Category-Specific Packet Collision Ratios: We further evaluate the collision ratios for different node categories. Let N_s^1 , N_s^2 , and N_s^0 denote the numbers of successfully received packets from first-level cluster heads, second-level cluster heads, and surrounding nodes, respectively. Correspondingly, let N_f^1 , N_f^2 , and N_f^0 denote the numbers of col-

lided packets for these categories. The packet collision ratios for the three categories are formulated as:

$$R_1 = \frac{N_s^1}{N_s^1 + N_f^1}, \quad (23)$$

$$R_2 = \frac{N_s^2}{N_s^2 + N_f^2}, \quad (24)$$

$$R_0 = \frac{N_s^0}{N_s^0 + N_f^0}. \quad (25)$$

7.3. Simulation Results and Discussion

Next, we analyze the performance of the CM-MAC protocol from two perspectives: the effectiveness of packet collision mitigation and the efficiency of the scheduling optimization algorithm. The CM-MAC protocol comprises three main mechanisms: hierarchical clustering, collision-avoidance constraints, and a GA-based optimization mechanism. Unlike CM-MAC, randCM-MAC employs a random node transmission order instead of GA-based optimization. This allows us to independently evaluate the contribution of the genetic algorithm. The effectiveness of the collision-avoidance constraints is assessed through the packet collision rates. Furthermore, the hierarchical clustering method serves as the framework for enforcing these constraints. Although its contribution is not evaluated separately, its feasibility is verified by comparing the collision rates among different node types.

7.3.1. Effectiveness of Packet Collision Mitigation

The CM-MAC protocol employs two strategies to reduce packet collision rates. First, it establishes transmission constraints to avoid packet collisions. Second, it uses a hierarchical clustered network architecture for partitioned scheduling, enabling effective use of local information.

Figure 6 shows the average packet collision ratio for all data packets in the network. Both the CM-MAC and randCM-MAC protocols exhibit significantly lower collision rates compared to the fully random-access ALOHA protocol. This substantial reduction demonstrates the effectiveness of the proposed transmission constraints designed to avoid collisions. However, the collision probability cannot be eliminated. The remaining collisions primarily occur between packets sent by peripheral nodes belonging to different sub-clusters, a phenomenon further investigated in Figure 7.

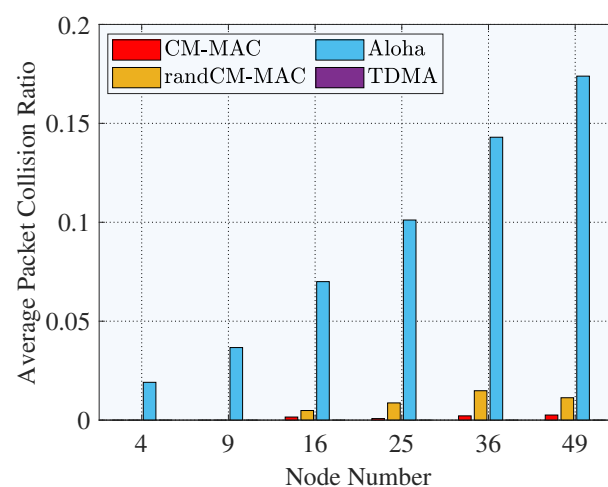


Figure 6. Average packet collision ratio with different numbers of nodes.

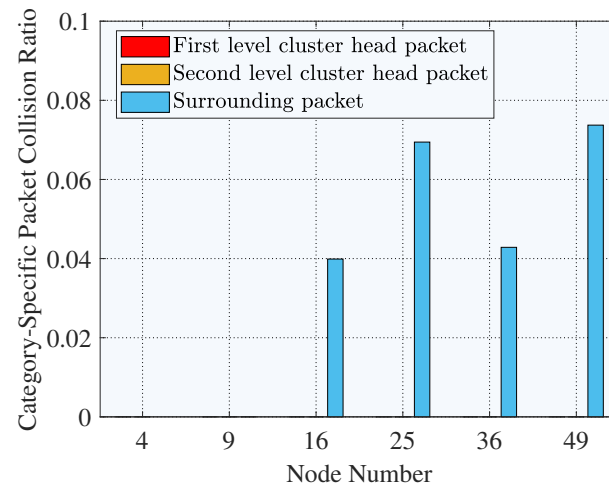


Figure 7. Category-Specific Packet Collision Ratio with different numbers of nodes.

Figure 7 illustrates the packet collision breakdown by node type within the hierarchical clustered network. The results show that packets from both first-level and second-level cluster heads achieve a zero collision rate, highlighting the effectiveness of the hierarchical scheduling in eliminating intra-cluster interference. In contrast, less than 10% of packets from surrounding nodes experience collisions. This discrepancy arises from the information asymmetry inherent in the scheduling process. A second-level cluster head schedules transmission times for its own sub-cluster surrounding nodes based on their location information and the schedules of known neighboring second-level cluster heads, effectively avoiding both intra-cluster and known inter-cluster collisions. However, the second-level cluster head lacks global network information. As a result, it cannot account for surrounding nodes in distant sub-clusters whose scheduling information is unavailable, which leads to the observed low-probability packet collisions.

In summary, the simulation results confirm that the proposed transmission constraints are highly effective in mitigating packet collisions. The CM-MAC protocol successfully leverages available local information to reduce interference. The remaining collisions do not result from a failure of the constraints but are an inevitable consequence of the lack of global network state information in a distributed environment.

7.3.2. Efficiency of the Scheduling Optimization Algorithm

The main distinction between the CM-MAC and randCM-MAC protocols lies in their transmission scheduling strategy. Specifically, CM-MAC employs a Genetic Algorithm to optimize the transmission order of nodes, whereas randCM-MAC uses a random transmission order.

The parameters (population size, mutation rate, crossover rate, and inheritance ratio) of our proposed Genetic Algorithm are determined based on standard practices in evolutionary computation and relevant literature. These values are chosen to maintain a balance between global exploration and local exploitation. To verify the robustness of these parameters, we conduct experiments by plotting the convergence curves under varying chromosome lengths. As shown in Figure 8, the algorithm consistently converges in terms of normalized scheduling span. This indicates that the selected parameters are effective and not overly sensitive to the chromosome length.

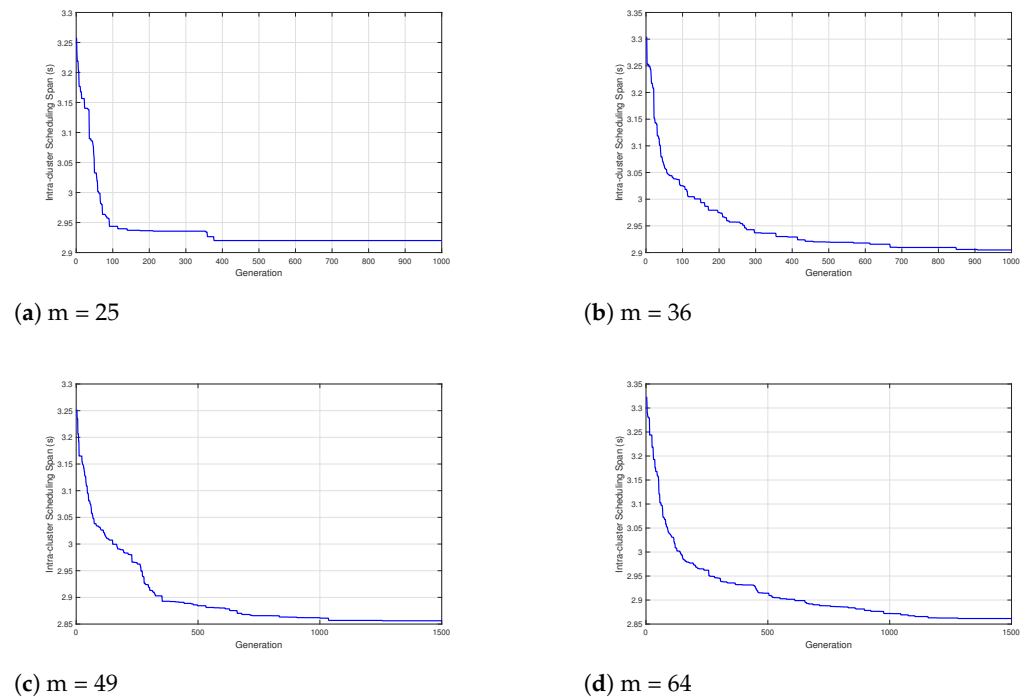


Figure 8. Convergence curves under varying chromosome lengths.

Figures 9 and 10 illustrate the performance of these protocols in terms of average update interval and network throughput, respectively. As shown in these figures, the ALOHA protocol significantly underperforms compared to the other three TDMA-based MAC protocols. This clearly demonstrates that in UACNs designed for AUV swarm information sharing, scheduling-based protocols offer considerable advantages over random-access protocols.

Among CM-MAC, randCM-MAC, and traditional TDMA protocols, CM-MAC delivers the best performance due to its GA-based optimization of transmission order. Notably, the performance advantage of CM-MAC becomes increasingly evident as the network scale expands. Additionally, although randCM-MAC employs a random order, it still outperforms the traditional TDMA protocol. This is because randCM-MAC schedules transmission times that satisfy transmission constraints, enabling concurrent transmissions among different nodes in the network.

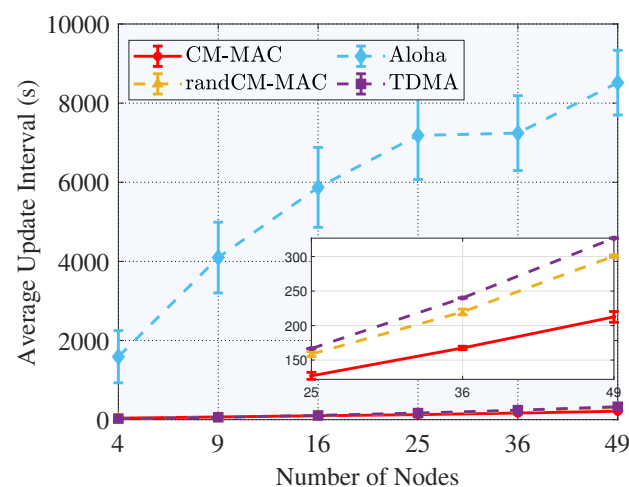


Figure 9. Average update interval with different numbers of nodes.

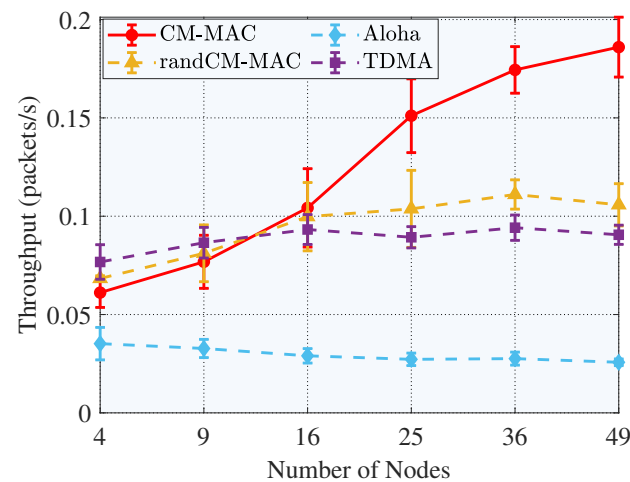


Figure 10. Throughput with different numbers of nodes.

8. Conclusions

This paper presents the CM-MAC protocol, a hierarchical scheduling-based solution designed to support reliable neighborhood multicasting in clustered AUV swarms. By establishing a two-tier cluster-based architecture, the protocol effectively decomposes the global coordination problem into localized scheduling sub-tasks. This approach reduces control signaling redundancy and adapts to dynamic network topologies. Theoretical analysis has identified the transmission constraints necessary to avoid packet collisions among mobile nodes, accounting for the unique challenges posed by underwater propagation delays. Simulation experiments have validated the performance of the CM-MAC protocol, demonstrating a significant reduction in packet collision ratios compared to contention-based Aloha protocols. Furthermore, the GA-based scheduling optimization showed superior throughput and shorter average update intervals when compared to random scheduling variants.

Despite the demonstrated improvements in throughput and latency, we acknowledge certain limitations in the current design. Future research will address these challenges from the following three perspectives.

Network Scalability and Information Constraints: The current protocol relies on a three-layer architecture to ensure collision-free scheduling based on local information. However, as the network scale expands, the lack of global state information makes it increasingly difficult to mitigate inter-cluster interference solely through deterministic constraints. To address this, future work will focus on integrating deterministic scheduling with data-driven optimization paradigms, such as Reinforcement Learning (RL). By leveraging the adaptive capabilities of RL, we aim to suppress inter-cluster interference and enhance network robustness in large-scale deployments without relying on rigorous prior information.

Energy Efficiency and Computational Overhead: While the hierarchical clustering and scheduling mechanisms ensure reliability, they introduce computational overhead that impacts the energy budget of battery-constrained AUVs. Future research will prioritize the development of lightweight, energy-aware scheduling algorithms. The goal is to balance the trade-off between scheduling precision and energy consumption, thereby extending the operational lifespan of the swarm.

Practical Deployment and Environmental Robustness: The simulations in this work assume idealized conditions, including perfect synchronization, precise positioning, and linear trajectories. In practical underwater deployments, however, the system faces challenges such as clock synchronization drifts, acoustic localization inaccuracies, ocean currents, and

ambient noise. These physical-layer imperfections can lead to time misalignment and require larger guard times, potentially reducing channel utilization. Future work will explicitly model these errors to evaluate performance degradation. Furthermore, we plan to validate the protocol's robustness through water tank experiments and open-sea trials to quantify its resilience against real-world hydrodynamic factors and hardware-induced latencies.

Author Contributions: Conceptualization, M.L.; methodology, M.L.; software, M.L.; validation, M.L.; writing—original draft preparation, M.L.; writing—review and editing, G.H.; funding acquisition, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, No. B250201201, and the China Postdoctoral Science Foundation under Grant No. 2025M780264. The APC was funded by the China Postdoctoral Science Foundation under Grant No. 2025M780264.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data links were provided because the data are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used Grammarly (v1.2.275.1918) for the purposes of checking English grammar, spelling, and writing style. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

References

1. Ni, R.; Wang, J.; Qin, D.; He, Z.; Li, Q.; Zhang, C. Review of autonomous underwater vehicle path planning. *Symmetry* **2026**, *18*, 476. [\[CrossRef\]](#)
2. Tan, J.; Sun, Y.; Cai, K.; Zhang, Y.; Zhang, L. Energy-efficient AUV path planning under multiple constraints using a synergistic differential evolution algorithm. *J. Mar. Sci. Appl.* **2025**, *21*, 153–169.
3. Zhang, J.; Sha, J.; Han, G.; Liu, J.; Qian, Y. A cooperative-control-based underwater target escorting mechanism with multiple autonomous underwater vehicles for underwater Internet of Things. *IEEE Internet Things J.* **2021**, *8*, 4403–4416. [\[CrossRef\]](#)
4. Yoon, S.; Qiao, C. Cooperative search and survey using autonomous underwater vehicles (AUVs). *IEEE Trans. Parallel Distrib. Syst.* **2011**, *22*, 364–379. [\[CrossRef\]](#)
5. Petillo, S.; Schmidt, H.; Balasuriya, A. Constructing a distributed AUV network for underwater plume-tracking operations. *Int. J. Distrib. Sens. Netw.* **2012**, *8*, 191235–191247.
6. He, L.; Xie, M.; Zhang, Y. A review of path following, trajectory tracking, and formation control for autonomous underwater vehicles. *Drones* **2025**, *9*, 286. [\[CrossRef\]](#)
7. Yan, T.; Xu, Z.; Yang, S.; Gadsden, A. Formation control of multiple autonomous underwater vehicles: A review. *Intell. Robot.* **2023**, *3*, 1–22. [\[CrossRef\]](#)
8. Er, M.J.; Gong, H.; Liu, Y.; Liu, T. Intelligent trajectory tracking and formation control of underactuated autonomous underwater vehicles: A critical review. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 543–555. [\[CrossRef\]](#)
9. Hadi, B.; Khosravi, A.; Sarhadi, P. A review of the path planning and formation control for multiple autonomous underwater vehicles. *J. Intell. Robot. Syst.* **2021**, *101*, 67. [\[CrossRef\]](#)
10. Yang, Y.; Xiao, Y.; Li, T. A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability. *IEEE Commun. Surv. Tut.* **2021**, *23*, 815–841. [\[CrossRef\]](#)
11. Yan, T.; Xu, Z.; Yang, S. Consensus formation tracking for multiple AUV systems using distributed bioinspired sliding mode control. *IEEE Trans. Intell. Veh.* **2023**, *8*, 1081–1092. [\[CrossRef\]](#)
12. Chen, S.; Ho, D. Consensus control for multiple AUVs under imperfect information caused by communication faults. *Inf. Sci.* **2016**, *370*, 565–577. [\[CrossRef\]](#)
13. Zhong, X.; Ji, F.; Chen, F.; Guan, Q.; Yu, H. A new acoustic channel interference model for 3-D underwater acoustic sensor networks and throughput analysis. *IEEE Internet Things J.* **2020**, *7*, 9930–9942. [\[CrossRef\]](#)
14. Zhuo, X.; Liu, M.; Wei, Y.; Yu, G.; Qu, F.; Sun, R. AUV-aided energy-efficient data collection in underwater acoustic sensor networks. *IEEE Internet Things J.* **2020**, *7*, 10010–10022. [\[CrossRef\]](#)

15. Stojanovic, M.; Preisig, J. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Commun. Mag.* **2009**, *47*, 84–89. [\[CrossRef\]](#)
16. 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. [\[CrossRef\]](#)
17. Han, Y.; Fei, Y. TARS: A traffic-adaptive receiver-synchronized MAC protocol for underwater sensor networks. *ACM Trans. Sens. Netw.* **2017**, *13*, 1–25. [\[CrossRef\]](#)
18. Marinakis, D.; Wu, K.; Ye, N.; Whitesides, S. Network optimization for lightweight stochastic scheduling in underwater sensor networks. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 2786–2795.
19. Li, C.; Xu, Y.; Xu, C.; An, Z.; Diao, B.; Li, X. DTMAC: A delay tolerant MAC protocol for underwater wireless sensor networks. *IEEE Sens. J.* **2016**, *16*, 4137–4146. [\[CrossRef\]](#)
20. Han, Y.; Fei, Y. DAP-MAC: A delay-aware probability-based MAC protocol for underwater acoustic sensor networks. *Ad Hoc Netw.* **2016**, *48*, 80–92. [\[CrossRef\]](#)
21. Mandal, P.; De, S.; Chakraborty, S.S. A receiver synchronized slotted Aloha for underwater wireless networks with imprecise propagation delay information. *Ad Hoc Netw.* **2013**, *11*, 1443–1455. [\[CrossRef\]](#)
22. Wei, X.; Guo, H.; Wang, X.; Wang, X.; Qiu, M. Reliable data collection techniques in underwater wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 404–431. [\[CrossRef\]](#)
23. Noh, Y.; Lee, U.; Han, S.; Wang, P.; Torres, D.; Kim, J.; Gerla, M. DOTS: A propagation delay-aware opportunistic MAC protocol for mobile underwater networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 766–782. [\[CrossRef\]](#)
24. Zhuo, X.; Qu, F.; Yang, H.; Wei, Y.; Wu, Y.; Li, J. Delay and queue aware adaptive scheduling-based MAC protocol for underwater acoustic sensor networks. *IEEE Access* **2019**, *7*, 56263–56275. [\[CrossRef\]](#)
25. Ng, H. H.; Soh, W.S.; Motani, M. Saturation throughput analysis of the slotted BiCMAC protocol for underwater acoustic networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 3948–3960. [\[CrossRef\]](#)
26. Zhuo, X.; Yang, H.; Liu, M.; Wei, Y.; Yu, G.; Qu, F. Data concurrent transmission MAC protocol for application oriented underwater acoustic sensor networks. *China Commun.* **2022**, *19*, 220–237. [\[CrossRef\]](#)
27. Diamant, R.; Shirazi, G.N.; Lampe, L. Robust spatial reuse scheduling in underwater acoustic communication networks. *IEEE J. Ocean. Eng.* **2014**, *39*, 32–46. [\[CrossRef\]](#)
28. Diamant, R.; Casari, P.; Campagnaro, F.; Zorzi, M. Leveraging the near-far effect for improved spatial-reuse scheduling in underwater acoustic networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1480–1493. [\[CrossRef\]](#)
29. ZZhang, R.; Cheng, X.; Cheng, X.; Yang, L. Interference-free graph based TDMA protocol for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4008–4019.
30. Bai, W.; Wang, H.; Shen, X.; Zhao, R. Link scheduling method for underwater acoustic sensor networks based on correlation matrix. *IEEE Sens. J.* **2016**, *16*, 4015–4022. [\[CrossRef\]](#)
31. Hsu, C.-C.; Kuo, M.-S.; Chou, C.-F.; Lin, K.C.-J. The elimination of spatial-temporal uncertainty in underwater sensor networks. *IEEE/ACM Trans. Netw.* **2013**, *21*, 1229–1242. [\[CrossRef\]](#)
32. Anjani, P.; Chitre, M. Propagation-delay-aware unslotted schedules with variable packet duration for underwater acoustic networks. *IEEE J. Ocean. Eng.* **2017**, *42*, 977–993. [\[CrossRef\]](#)
33. Ragavi, B.; Baranidharan, V.; Ramash Kumar, K. A novel hybridized cluster-based geographical opportunistic routing protocol for effective data routing in underwater wireless sensor networks. *J. Electr. Comput. Eng.* **2023**, *2023*, 5567483. [\[CrossRef\]](#)
34. Zhang, W.; Wang, J.; Han, G.; Zhang, X.; Feng, Y. A cluster sleep-wake scheduling algorithm based on 3D topology control in underwater sensor networks. *Sensors* **2019**, *19*, 156. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Islam, T.; Lee, Y. A cluster based localization scheme with partition handling for mobile underwater acoustic sensor networks. *Sensors* **2019**, *19*, 1039. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Zhang, W.; Han, G.; Liu, Y. A coverage vulnerability repair algorithm based on clustering in underwater wireless sensor networks. *Mob. Netw. Appl.* **2021**, *26*, 1107–1121.
37. Lin, C.; Han, G.; Wang, T.; Bi, Y.; Du, J.; Zhang, B. Fast node clustering based on an improved birch algorithm for data collection towards software-defined underwater acoustic sensor networks. *IEEE Sens. J.* **2021**, *21*, 25480–25488. [\[CrossRef\]](#)
38. Alkanhel, R.; Chaaf, A.; Samee, N.A.; Alohal, M.A.; Muthanna, M.S.A.; Poluektov, D.; Muthanna, A. DEDG: Cluster-based delay and energy-aware data gathering in 3D-UWSN with optimal movement of multi-AUV. *Drones* **2022**, *6*, 283. [\[CrossRef\]](#)
39. Guo, J.; Song, S.; Liu, J.; Chen, H.; Cui, J.-H.; Han, G. A hybrid NOMA-based MAC protocol for underwater acoustic networks. *IEEE/ACM Trans. Netw.* **2024**, *32*, 1187–1200. [\[CrossRef\]](#)
40. Li, J.; Zakharov, Y.V. Efficient use of space-time clustering for underwater acoustic communications. *IEEE J. Ocean. Eng.* **2018**, *43*, 173–183. [\[CrossRef\]](#)
41. Guo, J.; Song, S.; Liu, J.; Wan, L.; Yu, Y.; Han, G. An MC-CDMA-based MAC protocol for efficient concurrent communication in mobile underwater acoustic networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 12428–12443. [\[CrossRef\]](#)

42. Zhang, J.; Cai, M.; Han, G.; Qian, Y.; Shu, L. Cellular clustering-based interference-aware data transmission protocol for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3217–3230. [[CrossRef](#)]
43. Chen, Y.; Jin, Z.; Zeng, Q.; Yang, Q. A collision-avoided MAC protocol with time synchronization and power control for underwater sensor networks. *IEEE Sens. J.* **2022**, *22*, 19073–19087. [[CrossRef](#)]
44. Chen, Y.; Jin, Z.; Xing, G.; Zeng, Q.; Chen, Y.; Zhou, Z.; Yang, Q. An energy-efficient MAC protocol for three-dimensional underwater acoustic sensor networks with time synchronization and power control. *IEEE Access* **2023**, *11*, 20842–20860. [[CrossRef](#)]
45. Li, X.; Zhao, H.; Han, S.; Wang, B.; Liu, Z. Robust joint resource allocation algorithm for hierarchical underwater acoustic communication networks based on outage probability minimization. *Trans. Emerg. Tel. Tech.* **2021**, *32*, e4292. [[CrossRef](#)]
46. Liu, M.; Zhuo, X.; Yuan, Y.; Lu, Y.; Wei, Y.; Tu, X.; Qu, F. Adaptive scheduling MAC protocol in underwater acoustic broadcast communications for AUV formation. *IEEE Internet Things J.* **2023**, *10*, 6887–6901. [[CrossRef](#)]
47. Wenz, G.M. Acoustic ambient noise in the ocean: Spectra and sources. *J. Acoust. Soc. Am.* **1962**, *34*, 1936–1956. [[CrossRef](#)]
48. Tu, X.; Wei, Y.; Qu, F.; Song, A. Low computational complexity RLS-based decision-feedback equalization in underwater acoustic communications. *IEEE J. Ocean. Eng.* **2024**, *49*, 1067–1088. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.