

Distributed Channel Sensing MAC Protocol for Multi-UUV Underwater Acoustic Network

Tianyou Qiu^{ID}, Yiping Li^{ID}, and Xisheng Feng

Abstract—The media access control (MAC) protocol is a primary technology to achieve a collision-free transmission and improve the channel utilization for underwater acoustic networks (UANs). Most relative studies currently focus on solving underwater acoustic sensor networks (UASNs) with a limited mobility and fixed topology, where nearly all the data is transmitted from the network edge to the sink node. However, multiple underwater unmanned vehicle (multi-UUV) networks have complex and variable topologies and usually contain both unicast and multicast transmission. The MAC protocol designed for it should be flexible and achieve a low latency while avoiding signal collisions. This article proposes a distributed channel sensing MAC (DCSM) protocol based on the aforementioned requirement, which allows each node to sense changes in surrounding traffic by sharing the slot usage table. The proposed adaptive transmission scheduling algorithm makes each vehicle select the optimal transmission slot based on a predictive value function, which utilizes long propagation delays to achieve spatial-temporal reuse and is adaptive to the network load. The simulation results demonstrate that compared to the other four distributed MAC protocols, DCSM had significant advantages in the transmission success rate (15% higher) and throughput (35% higher) and achieved a short latency. The adaptability and flexibility of DCSM have also been verified through simulations.

Index Terms—Adaptive transmission scheduling, channel sensing, media access control (MAC) protocol, multiple underwater unmanned vehicle (multi-UUV), underwater acoustic network (UAN).

I. INTRODUCTION

UNMANNED underwater vehicle underwater unmanned vehicleless (UUVs) are a kind of advanced intelligent devices for humans to better understand and know the ocean. Furthermore, multiple underwater unmanned vehicle (multi-UUV) systems are robust, well-coordinated, highly efficient,

Manuscript received 17 October 2023; revised 16 December 2023; accepted 3 January 2024. Date of publication 12 January 2024; date of current version 25 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4602404, and in part by the National Natural Science Foundation of China under Grant 42176194. (*Corresponding author: Yiping Li*)

Tianyou Qiu is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, also with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, Anhui, China, and also with the Innovation Academy of Robotics and Intelligent Manufacturing Innovation, Chinese Academy of Sciences, Shenyang 110169, China (e-mail: qutianyou@sia.cn).

Yiping Li and Xisheng Feng are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the Innovation Academy of Robotics and Intelligent Manufacturing Innovation, Chinese Academy of Sciences, Shenyang 110169, China (e-mail: lyp@sia.cn; fxs@sia.cn).

Digital Object Identifier 10.1109/JIOT.2024.3352007

and can perform more complex and large-area tasks [1]. Several multi-UUV systems have been recently used in various fields, such as target tracking [2], geotechnical surveys [3], and dock patrols [4].

A multi-UUV system mainly operates in an underwater acoustic network (UAN), making it a part of the Internet of Things (IoT) and enabling remote monitoring from shore-based control centers. Because most underwater acoustic communication systems provide half-duplex transmission and have low-data rates, signal collisions are a major problem in UANs [5]. The media access control (MAC) protocol for UANs is the primary technology used for achieving collision-free transmission and improving channel utilization. Most relative studies currently focus on solving underwater acoustic sensor networks (UASNs) with limited mobility and a fixed topology, where nearly all the data is transmitted from the network edge to the sink node [6]. However, multi-UUV ad hoc networks simultaneously contain broadcast, unicast, and multicast transmission, and the network topology is complex and variable owing to the mobility of UUVs [7]. In addition, cooperation and remote monitoring require real-time communication; therefore, the MAC protocol should be substantially flexible, and achieve a low latency and high throughput while avoiding signal collisions.

Compared to radio in the air communication, underwater acoustic communication has a significantly more pronounced signal propagation delay, resulting in a spatial-temporal uncertainty of the MAC protocol [8]. Specifically, the arrival time of a signal not only depends on the transmission time, but also on the transmitter-receiver distance, which is called space-time coupling [9]. This characteristic causes mature terrestrial wireless MAC protocols to be ineffective or inefficient in UANs. Previous studies have found that using long the propagation delay ratio-nally can effectively improve channel utilization. Specifically, multiple transmissions run concurrently without signal collisions in the same space-time, which is called “concurrent transmission” [10].

Fig. 1 presents an example of concurrent transmission complicated by a single node, where the dotted circles indicate the propagating signals. Node A first sends a packet to node B and continues transmitting another packet to node C before the previous transmission is completed, that is, node A utilizes temporal reuse. Furthermore, concurrent transmission can be achieved through spatial reuse. In Fig. 2(a), where the overlapped part indicates “cross in flight” [11], nodes A and B concurrently transmit packets to node C without collisions,

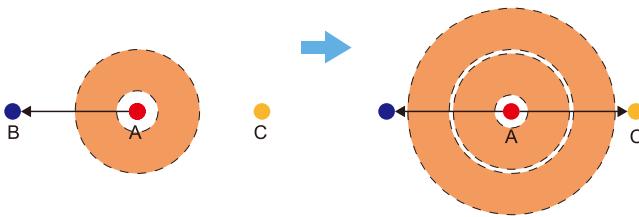


Fig. 1. Example of temporal reuse.

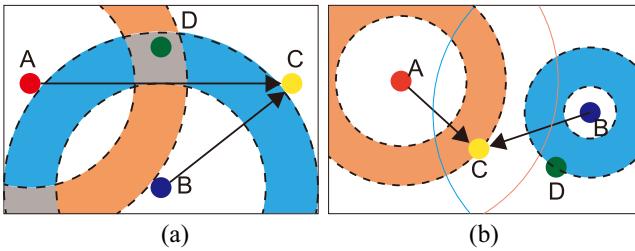


Fig. 2. Two examples of spatial reuse. (a) In single collision domain. (b) In multiple collision domain.

and the collision occurs at node D , which can be referred to as the “interference alignment” [12].

In addition, the limited transmission range causes many UANs to run in the multicollision domain, resulting in the problems of “hidden terminals” and “exposed terminals” [13]. In Fig. 2(b), where the solid circles indicate the transmission ranges, nodes A and B cannot hear one another and become hidden terminals for their transmission to node C . Additionally, if node C transmits to node A and node D transmits to node B , D will become the exposed terminal of C because D is outside the transmission range of A . Spatial reuse can also be utilized in multicollision domains. As shown in Fig. 2(b), transmissions from A and B to C run concurrently without collisions, despite A and B being hidden terminals.

Most studies regarding MAC protocols for UANs have recently focused on addressing temporal-spatial uncertainty to avoid collisions, while utilizing concurrent transmissions to improve channel utilization. With the location information of all the nodes, it is possible to avoid collisions and achieve a significantly high-channel utilization by scheduling the order and transmission time of all the nodes. In our previous study, we proposed an optimal broadcast scheduling algorithm that used a heuristic algorithm to solve this problem [14]; Liu et al. [15] also proposed an adaptive scheduling algorithm for broadcast communication in UUV formation. However, these were designed for fully connected networks.

For mobile UANs in multiple collision domains, a centralized scheduling method is unsuitable because it requires accurate network-wide information, incurs excessive overhead, and the information obtained by the central node may be outdated. Therefore, an adaptive and distributed MAC is preferred. Certain distributed protocols are variants of the carrier-sense multiple access (CSMA) that avoid collisions by monitoring channels. For instance, UAN-CW [16] uses a random backoff mechanism based on competition windows. To improve the channel utilization and the fairness of nodes

accessing the channel, Guo et al. [17] proposed an adaptive varying contention window MAC protocol based on propagation delay. However, a protocol-based only on overhearing cannot solve the “hidden/exposed terminal” problem nor effectively achieve spatial-temporal reuse. Therefore, many proposed MAC protocols use control packets to reserve the channel, sense propagation delays, or predict the transmission of the surrounding nodes. For example, in slotted FAMA [18], nodes exchange RTS/CTS before transmitting data packets to reserve the channel. However, the exchange of control packets before transmission increases the delay to a certain extent. With the development of artificial intelligence, several MAC protocols based on reinforcement and deep learning have been proposed to learn an optimal action through trial-and-error interactions in a dynamic environment, such as the UW-ALOHA-Q [19]. However, these protocols have long convergence times when applied to UANs.

Thus, we are motivated to design a MAC protocol that senses the surrounding channel without exchanging control packets and learns a collision-free schedule with a short convergence time. The key contributions of this study are summarized as follows.

- 1) A novel channel-sensing method that does not exchange control packets is proposed. With a short slot usage table (SUT) added to a data packet, each node can share its communication state with the surrounding nodes and sense the channel information (including propagation delay and neighboring reception/transmission slots).
- 2) An adaptive transmission-scheduling algorithm to cooperate with the channel-sensing method was developed. The proposed algorithm uses the sensed information to predict the results of future transmissions and chooses an optimal slot to send. This algorithm can increase the opportunities for concurrent transmissions while avoiding collisions. The algorithm can quickly converge with the sensed information and an implicit acknowledgment method.
- 3) The MAC protocol exhibits strong adaptability and flexibility. It can operate without time synchronization and can be applied to different topologies, such as cluster-based and mesh networks. This protocol supports both unicast and multicast transmission in one hop and supports two-way communication. Considering the variable traffic load of a multi-UUV network, the protocol can adjust the local load by sensing the traffic.

Section II presents the relative studies regarding adaptive MAC in UANs, which is followed by a description of the network model in Section III. Section IV introduces the distributed channel sensing method and Section V presents the adaptive scheduling algorithm. The simulation and conclusions are presented in Sections VI and VII, respectively.

II. RELATED WORK

The MAC protocols for solving the temporal-spatial uncertainty of UANs have been recently divided into contention-based and scheduling-based protocols. Several contention-based protocols are based on handshake

mechanisms. For example, T-lohi [20] employs a tone-based contention resolution mechanism that exploits space-time uncertainty to detect collisions and improve the throughput; In DOTS [10] and DCT [21], nodes establish a local transmission delay graph and are allowed to transmit concurrently without possible interference; To address spatial unfairness, Zheng et al. [22] proposed a handshake-based protocol with a contention window spatial fairness adjustment strategy, where nodes autonomously adjust their contention states based on the perceived information. To address the high-delay problem, Ma et al. [23] designed an improved handshake-based low delay (i-HALO) MAC with a dynamic adaptive backoff algorithm, which achieved the “one handshake, multiple transmission.” The advantage of this type of protocol is that it is substantially flexible and is suitable for various scenarios; however, the utilization of spatial-temporal reuse is insufficient, and the handshake adds an additional delay.

Scheduling-based protocols are usually variants of TDMA, in which each node transmits according to the schedule generated by a centralized or distributed method. Centralized scheduling-based protocols can utilize concurrent transmission through optimization algorithms to achieve high-channel utilization. For instance, Zhang et al. [24] proposed a dynamic interference-free graph and two clustering algorithms to achieve a high-spatial reuse efficiency; Liu et al. [25] formulated a slot scheduling problem in a frame into a combinatorial optimization problem and proposed a heuristic algorithm to minimize the end-to-end delay. However, both require an information collection phase to obtain information from all the nodes, which is inapplicable to large-scale multi-UUV networks. Distributed scheduling-based MAC has recently become more popular.

Certain distributed scheduling-based MAC protocols have been combined with a handshake mechanism. For example, to achieve good throughput and energy savings, Roy and Sarma [26] proposed a synchronous duty-cycled reservation-based MAC, in which a receiver collects the RTS from all the contending nodes and subsequently generates and broadcasts a transmission schedule for all the contending nodes. Shang and Du [27] proposed a concurrent scheduling MAC (CS-MAC) with a notably similar mechanism; the difference being that the receiver uses propagation delay from all contending nodes to achieve high-spatial reuse efficiency. However, they are usually only suitable for specific network architectures, such as clustered or layered networks. The learning-based scheduling MAC is relatively more flexible. To improve network resilience and adaptability, Park et al. [28] proposed UW-ALOHA-QM with Q -learning and a 7-uniform random back-off scheme. Geng and Zheng [29] also applied deep Q -learning to the MAC protocol of UANs and proposed async-DL-MAC that expands the action space and enables the agent to learn the optimal time delay as the transmission start time in each time slot. To reduce the cost of online training, Ye et al. [30] developed a deep reinforcement learning (DRL)-based MAC and proposed a new delayed-reward deep Q -network and a nimble training mechanism. Learning-based scheduling MACs do not require the collection of surrounding network information; However, their response speed to

network changes is slow owing to their long convergence time. Furthermore, several predictive scheduling protocols exist. For instance, UW-SEEDEX [31] enables nodes to predict the transmission of surrounding nodes by sharing “seeds” to determine a suitable transmit slot. The “update and communication intervals” alternate to cope with the network changes. However, the lengthy time slot in UW-SEEDEX increases the idle time of the channel.

By analyzing the advantages and disadvantages of the existing distributed MAC and considering the requirements of multi-UUV networks, this study proposes a novel distributed channel sensing MAC (DCSM) protocol. Unlike the protocols previously indicated, our protocol applies both learning and prediction. To achieve a short convergence time, we designed a channel sensing method that avoids meaningless trial and error. The sensing method is passive and has no additional delay.

III. NETWORK MODEL

Before introducing our proposed protocol, we first introduce the network model, including the characteristics of multi-UUV ad hoc networks and the acoustic signal collision model.

A. Multi-UUV Ad Hoc Network

Large-scale multi-UUV underwater ad hoc networks typically exhibit the following characteristics.

- 1) High-speed nodes, whose movements are unpredictable; thus, the network topology is variable.
- 2) The communication distance of a UUV is limited owing to the limited energy carried; therefore, the network is usually a multicollision domain.
- 3) The network can be cluster-based, in which each UUV transmits to its cluster header or a mesh network, where each UUV can be a relay node.
- 4) The number of nodes varies owing to the vehicles being in/out of the network during a task.
- 5) Unicast and multicast transmission coexist in two-way communication.

In such a network, in addition to the location, the transmission load and the one hop destination of a node may change. Therefore, the MAC protocol aims to achieve a strong adaptability while providing a sufficient communication performance in terms of collision avoidance, latency, and throughput. This study does not consider the energy efficiency issue, assuming that the transmission power all nodes is equal and fixed.

B. Acoustic Signal Collision Model

In a multicollision domain UAN, determining whether a collision has occurred based on if the signals overlap at the receiving node is inaccurate. The collision model measures whether packets can be successfully received based on the signal-to-interference plus-noise ratio (SINR) at the receiving node

$$\text{SINR}(l, f) = \frac{E_t}{A(l, f)(N(f)\Delta f + E_{\text{in}})} \quad (1)$$

where $A(l, f)$ is the path loss in the underwater acoustic channel, and depends on the propagation range l and signal frequency f . $N(f)$ is the power spectral density of the ambient noise, and Δf is the bandwidth of the receiver noise. The detailed calculation method of $A(l, f)$ and $N(f)$ can be found in [32]. The AN product limits the available bandwidth and communication range of underwater acoustic signals. E_t is the transmit power of the signal and E_{in} is the power of the interference signal from other transmissions. E_{in} is related to the attenuation of the interference signals and the modulation methods. The calculation of E_{in} in frequency-hopped frequency shift keying (FH-FSK) can be found in [33].

The probability of the signal error caused by the SINR is related to the modulation method. If each symbol carries 1 bit of information, such as in FH-FSK, the bite error rate (BER) can be estimated by the following:

$$\text{BER} = \frac{1}{2} \left(1 - \sqrt{\frac{\text{SINR}}{1 + \text{SINR}}} \right). \quad (2)$$

If the value of E_{in} from another transmission is excessively large, causing the BER to be too large to receive the packet, this transmission suffers a collision.

Therefore, multiple signals overlapping at the receiving node is only the necessary condition for a collision. However, owing to the actual unpredictability of the physical layer, it is still necessary to avoid multiple overlapping signals at the receiving node to prevent collisions.

IV. DISTRIBUTED CHANNEL SENSING METHOD

Because exchanging control packets before transmission results in additional latency and multi-UUV networks require real-time communication, we propose a novel channel sensing method based on SUTs. Each node uses three types of SUTs to sense the channels: 1) the shared SUT is added to the header of a packet to share the previous local communication states; 2) the neighbor SUT includes the states of the nodes that are sensed by the local node; and 3) the local SUT includes the states of the local node. A shared SUT is the latest part of the local SUT.

Table I lists the parameters and variables used to sense the channels. First, we describe how a node senses the channels by these three types of SUTs.

A. Slot Usage Table

The uncertainty of collisions in UANs is reflected by the following: 1) the arrival time of a signal is difficult to determine, as it is dependent on both transmission time and propagation distance and 2) the local node is unaware of the nodes with two-hop distances (hidden terminals) or whether there is a one-hop distance between two neighbors (exposed terminals). Our protocol eliminates the aforementioned uncertainty by sharing local SUTs.

The MAC header field added to a packet is shown in Fig. 3, including the basic information field and shared SUT. The basic information field includes the unique local address

TABLE I
PARAMETERS AND VARIABLE USED WHEN SENSING THE CHANNEL

T_c	Average time between sending at least one packet of an UUV
Len	Length of a shared SUT
τ	Duration of transmitting a standard packet
t_g	Guard time in a slot
Δt	Length of a time slot
s_i^S/s_i^L	i-th element in a shared/local SUT
q_i^x	i-th element in a neighbor SUT about node x
t_{off}^x	Offset of the slot start time between node x and local node
P_L/P_x	Position of the local node/node x
t_s/t_c	Start time of the current slot/ The current time
t_{ctn}^x	Time from node x receiving or overhearing a packet to the beginning of the next slot

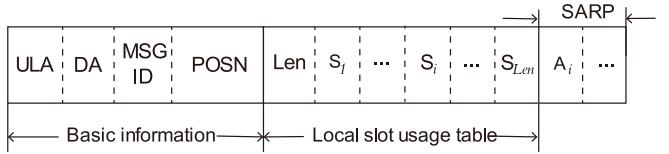


Fig. 3. MAC header in a data packet.

(ULA), destination address (DA), and message ID. The positional information (POSN) is necessary and can be delivered all the way to the top layer for navigation or data labeling.

Regarding the shared SUT, Len represents its length and the units $s_i^S \in S$ represent the communication state of the sender during the i th slot before sending this packet.

Not, the time slot used by the protocol is only composed of a “standard packet” duration τ and a guard time t_g , that is, slot length $\Delta t = \tau + t_g$. To avoid collisions, we only need to consider whether collisions occur during signal processing at the receivers instead of during propagation. A packet with a length of τ is called a standard packet in this article. If a longer packet must be sent, it is divided into multiple standard packets and transmitted at the beginning of different slots.

A s_i^S only requires 2 bits because there are only four elements in $S = \{IS, TX, OH, RX\}$.

- 1) **IS (0):** The sender did not receive or send a packet within this slot (idle state).
- 2) **TX (1):** The sender sends a packet at the beginning of this slot (transmit state).
- 3) **OH (2):** The sender completes a packet reception within this slot, and the destination of the packet does not include the sender (overheard state).
- 4) **RX (3):** The sender completes a packet reception within this slot, and the destination of the packet includes the sender (receive state).

Note, a node in the idle state does not indicate that its physical layer is idle because its transducer may have encountered a signal collision that causes reception failure. Because the propagation delay is usually not an integer multiple of the slot length, the completion of packet reception is usually not at the end of the slot; therefore, receiving a standard packet may span two slots.

$s_0^S = TX$ is an implicit information in the shared SUT, which indicates that the sender is in the TX state when transmitting the packet.

The source address of received packets (SARP) shown in Fig. 3 contains the source addresses of the packets received in the slots with the RX state in the shared SUT. SARP is used for a delayed acknowledgment mechanism, which will be introduced later. An address in SARP is one byte long.

The transmission frequency determines Len : if each node needs to send at least one packet at interval $T_C = C\Delta t$, then $\text{Len} \geq C$. The transmitting interval of UUVs is relatively short; therefore, the shared SUT is only a few or dozens of bytes long and the extra overhead is minimal.

The shared SUT is only a part of the local SUT maintained by each node. In addition, each node maintains a neighbor SUT by extracting the shared SUTs from the received or overheard packets. The local SUT unit s_i^L also belongs to S . However, the neighbor SUT unit q_i^x belongs to Q , which represents the communication state of neighbor x during the i th slot and has a different meaning. There are six elements in $Q = \{ER, SR, VC, EO, NO, UK\}$.

- 1) *ER*: The neighbor is expected to receive a packet from the local node within this slot (expected reception).
- 2) *SR*: The neighbor receives a packet from the local node within this slot (successful reception).
- 3) *VC*: The neighbor does not receive or send a packet within this slot (vacant).
- 4) *EO*: The neighbor receives a packet from another node or sent a packet within this slot (effective occupation).
- 5) *NO*: The neighbor overhears a packet or experiences a collision within this slot (noneffective occupation).
- 6) *UK*: The local node has no information to determine the state of the neighbor in this slot (unknown).

The state of *VC* differs from *IS*. When a node is in the *VC* state, its physical layer is idle. *ER* and *SR* can reflect whether transmission from the local node is successful, *VC* and *EO* are used to avoid collisions, and *NO* is used to achieve the interference alignment.

Because the nodes are not synchronized, the beginning of the same slot at different nodes differs. For example, there is a time offset between q_i^x and s_i^L , called the slot offset, which is referred to as t_{off}^x .

B. Updating the Local and Neighbor SUTs

The maximum length of the local and neighbor SUTs is $K \cdot C$. First, at the beginning of each time slot, the node must add new units in the local SUT and neighbor SUT by the first-in first-out method. The unit added to the local SUT is *IS*, and the unit added to the neighbor SUT is *UK*. Each time a node starts sending or completes receiving information, the two SUTs must be updated.

1) *Updating After Receiving/Overhearing a Packet*: When a node receives or overhears a packet from node x , it proceeds as follows.

a) *Update the units and slot offset about node x* :

Regardless of whether the local node is the destination of the packet, it first extracts the packet header information, including

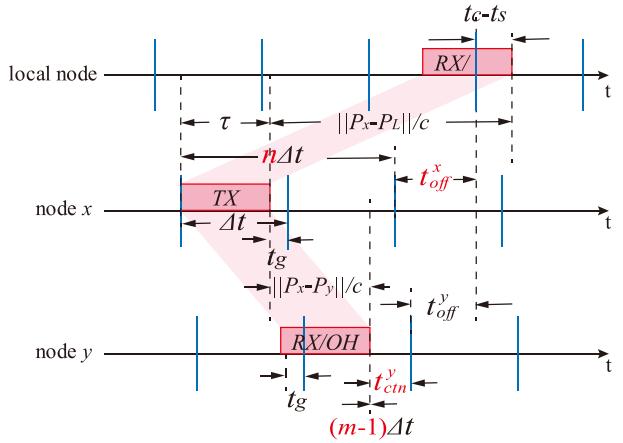


Fig. 4. Principle of calculating the slot count and slot offset.

the position P_x of x and the shared SUT ($s_i, i = 0, \dots, \text{Len}$). It then uses its current position P_L , the current time t_c , and the start time t_s of the current slot to calculate the slot offset t_{off}^x , as well as the slot count n from the transmission to the present by the following:

$$\begin{aligned} n\Delta t \leq n\Delta t + t_{\text{off}}^x &= t_s - t_c + \tau + \frac{\|P_x - P_L\|}{c} \leq (n+1)\Delta t \\ n \in \mathbb{Z}, 0 \leq t_{\text{off}}^x &< \Delta t \end{aligned} \quad (3)$$

where $c = 1500$ m/s is the speed of sound. The principle of (3) is illustrated in Fig. 4. If the current time is the beginning of a local slot, then the interval between the beginning of the slot at node x and the current time is t_{off}^x . Furthermore, n represents the number of slots passed on the timeline of node x from sending the packet to the local node to obtain the packet. In practice, an estimation error in the propagation delays is introduced by an inaccurate sound speed and positioning error; therefore, the guard time τ_g is used to avoid an erroneous n .

Subsequently, the local node records P_x and t_{off}^x and uses n and the shared SUT to update the neighbor SUT. Two points should be noted regarding t_{off}^x :

- 1) The calculated t_{off}^x is filtered using historical values for future use.
- 2) If the packet does not contain the location of x , it will not update t_{off}^x but will use the previously recorded P_x and t_{off}^x to solve for n in (3).

The local node uses Algorithm 1 to update the units from q_n^x to $q_{n+\text{Len}}^x$ in the neighbor SUT.

In Algorithm 1, the node uses lines 3–5 to assign elements to the unit when the unit is *UK*. Because *VC* is not equal to *IS*, line 3 indicates a preliminary speculation. Line 4 is owing to the definition of *EO*, line 5 is because *NO* includes the *OH* state. Lines 7–15 correct the unit when it is not *UK*. If a unit in the neighbor SUT is not *UK*, it has been updated by prediction before. In line 9, A_i (in the SARP) is the source address for unit s_i^S , and A_L is the address of the local node. Lines 7–11 indicated that, after the local node sent a packet to node x , it first predicted the state of x in a slot as *ER*; if the corresponding unit in the shared SUT from x is *RX* and the source address is the local address, then the previous transmission can be considered successful. This is a delayed

Algorithm 1 Update the Units in the Neighbor SUT by Shared SUT

Input: $x, n + i, s_i$
Output: q_{n+i}^x

```

1: if  $q_{n+i}^x = UK$ , then
2:   switch  $s_i$  do
3:     case IS, do  $q_{n+1}^x \leftarrow VC$ 
4:     case TX or RX, do  $q_{n+1}^x \leftarrow EO$ 
5:     case OH, do  $q_{n+1}^x \leftarrow NO$ 
6:   Otherwise, do  $q_{n+1}^x \leftarrow s_i$ 
7: else if  $q_{n+i}^x = ER$ , then
8:   if  $s_i = RX$ 
9:     if  $A_i = AL$ , then  $q_{n+1}^x \leftarrow SR$ 
10:    else,  $q_{n+1}^x \leftarrow EO$ 
11:   else,  $q_{n+1}^x \leftarrow NO$ 
12: else if  $q_{n+i}^x = EO$ , then
13:   if  $s_i \neq TX$  and  $s_i \neq RX$ , then  $q_{n+1}^x \leftarrow NO$ 
14: else if  $q_{n+i}^x = NO$ , then
15:   if  $s_i = RX$ , then  $q_{n+1}^x \leftarrow EO$ 

```

and implicit acknowledgment mechanism. Lines 12 and 13 are owing to the definition of *EO*. Line 15 is owing to *RX* indicating reception without collision; therefore, it can correct the unit of *NO* obtained by prediction.

Note, if the local node receives a packet from x for the first time, it adds the new neighbor to the neighbor SUT, initializes units with *UK*, and then uses Algorithm 1 for updating.

b) *Update the local SUT*: If the packet's destination includes the local node, set s_0^L in the local SUT to *RX*; otherwise, set it to *OH*. Note, if the current time is out of the guard time, that is, $t_c - t_s < \tau$, indicating that the packet's receiving started at the previous slot, set $s_1^L = s_0^L$.

c) *Update units of other nodes in the neighbor SUT*: If the local node infers that the neighbor y in the neighbor SUT is in the one-hop range of node x by recorded information, it calculates the slot number m of the reception by node y based on P_x, P_L, P_y , and t_{off}^y by the following:

$$(m-1)\Delta t \leq (m-1)\Delta t + t_{\text{ctn}}^y = \frac{\|P_x - P_L\| - \|P_x - P_y\|}{c} + t_s - t_c - t_{\text{off}}^y \leq m\Delta t$$

$$m \in Z, 0 < t_{\text{ctn}}^y \leq \Delta t. \quad (4)$$

The t_{ctn}^y obtained represents the time from the node's completion of packet reception to the beginning of the next slot. The principle of (4) is illustrated in Fig. 4. If the packet's destination includes y , set $q_m^y = EO$; otherwise, set $q_m^y = NO$. If $t_{\text{ctn}}^y > t_g$, which indicates that the reception of the packet spans two slots, set $q_{m+1}^y = q_m^y$. Note, because $\|P_x - P_L\| - \|P_x - P_y\|$ may be less than zero, m may be negative; therefore, the update predicts the future slot state.

The local node updates the relevant units of all one-hop neighbors of x in the neighbor SUT. Note, these inferred updates are less reliable than the corrected updates.

2) *Updating After Transmit a Packet*: When a local node sends a standard packet at the beginning of a slot, it copies

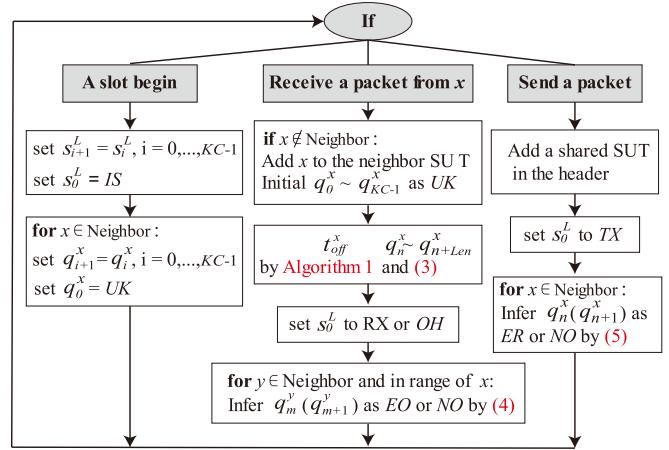


Fig. 5. Flowchart of the distributed channel sensing with SUTs.

the top Len units of the local SUT into the header. When the packet is delivered to the physical layer.

a) *Update the local SUT*: Set the unit s_0^L representing the current slot in the local SUT to *TX*.

b) *Update the neighbor SUT*: Compute slots in which the neighbors will receive the packet. Considering neighbor x as an example, predict the slot number n and t_{ctn}^x of the reception of the packet by

$$\begin{aligned} -n\Delta t \leq \tau + \frac{\|P_x - P_L\|}{c} + t_{\text{off}}^x \\ = (1-n)\Delta t - t_{\text{ctn}}^x < (1-n)\Delta t \\ n \in Z, 0 < t_{\text{ctn}}^x \leq \Delta t. \end{aligned} \quad (5)$$

The principle of (5) is similar to that of (3) and (4). Subsequently, the local node use n, t_{ctn}^x to update q_n^x . If the packet's destination includes x , set $q_n^x = ER$; otherwise, $q_n^x = NO$. If $t_{\text{ctn}}^x > t_g$, set $q_{n+1}^x = q_n^x$.

The aforementioned update is a prediction of the future and is likely incorrect. The prediction may be misleading if there is no shared SUT available for correction. For example, the local node has sent a packet to x and predicted q_n^x as *ER*. Suppose that the local node does not receive any shared SUT from x and does not receive other signals that may collided with the sent signal. In that case, the packet is considered to have been successfully received. However, a packet from a nonneighbor node (a hidden terminal) may collided with the packet on x . Therefore, shared SUTs are necessary in this method.

The complete channel-sensing process for a node is shown in Fig. 5. As the node receives more shared SUTs from the surrounding nodes, it has greater knowledge of the surrounding channel.

V. ADAPTIVE NO-COLLISION TRANSMISSION SCHEDULE

The neighbor SUT includes information regarding the slot-usage habits of the surrounding nodes and the success or failure of local transmission. Using this information, a node can quickly learn and effectively avoid collisions. We use a predictive transmission-value function to achieve adaptive no-collision transmission scheduling.

A. Predictive Transmission Value Function

Considering that most packets in multi-AUV networks are generated periodically, we begin our research with the sending strategy of periodically generating standard data packets. Suppose the local node generates a standard packet every $T_C = C\Delta t$. Whenever the node generates a packet, it selects a slot from the future C slots to send it.

The scheduling algorithm uses the sensed information to deduce the value generated by transmitting the packet in each candidate slot and selects the slot with the highest value as the planned sending slot.

When defining the transmission value, we mainly consider the following three factors.

- 1) Whether the packet will arrive at the destination node when its state is *EO* or *NO*, causing a collision at the destination.
- 2) Whether the packet will arrive at nondestination nodes when their states are *EO* or *NO*, causing collisions at other neighbors.
- 3) Whether the packet will overlap with the reception of other packets to the local node, causing a collision at the local node.

To compute the specific values, we propose mapping $R_n^x \leftarrow q_n^x$ and $R_n^L \leftarrow s_n^L$. Here, R_n^x represents the value generated if the packet reached node x in the n th slot in the past, and R_n^L represents the value generated if the packet was transmitted at the beginning of the n th slot; they are referred to as the *R*-value. The *R*-values corresponding to different elements in \mathbf{Q} should satisfy the following:

$$\begin{aligned} 1 &= R_n^x(\text{SR}) > R_n^x(\text{VC}) > R_n^x(\text{ER}) > R_n^x(\text{UK}) \\ &= 0.5 > R_n^x(\text{NO}) = R_n^x(\text{EO}) = 0, \text{ if } x \in \mathbf{D} \end{aligned} \quad (6)$$

$$\begin{aligned} 1 &= R_n^x(\text{SR}) = R_n^x(\text{VC}) > R_n^x(\text{NO}) > R_n^x(\text{ER}) \\ &> R_n^x(\text{UK}) = 0.5 > R_n^x(\text{EO}) = 0, \text{ if } x \in \mathbf{A} - \mathbf{D}. \end{aligned} \quad (7)$$

The mapping we used in the simulation was as follows:

$$R_n^x = \begin{cases} 1, & \text{if } q_n^x = \text{SR} \\ 0.8, & \text{if } q_n^x = \text{VC} \\ 0.6, & \text{if } q_n^x = \text{ER} \\ 0.5, & \text{if } q_n^x = \text{UK} \\ 0, & \text{if } q_n^x = \text{NO} \text{ or } q_n^x = \text{EO} \end{cases}, \text{ if } x \in \mathbf{D} \quad (8)$$

$$R_n^x = \begin{cases} 1, & \text{if } q_n^x = \text{SR} \text{ or } q_n^x = \text{VC} \\ 0.8, & \text{if } q_n^x = \text{NO} \\ 0.6, & \text{if } q_n^x = \text{ER} \\ 0.5, & \text{if } q_n^x = \text{UK} \\ 0, & \text{if } q_n^x = \text{EO} \end{cases}, \text{ if } x \in \mathbf{A} - \mathbf{D} \quad (9)$$

$$R_n^L = \begin{cases} 1, & \text{if } s_n = \text{TX} \text{ or } s_n = \text{ID} \\ 0.5, & \text{if } s_n = \text{OH} \\ 0, & \text{if } s_n = \text{RX} \end{cases} \quad (10)$$

where set \mathbf{A} consists of all the nodes in the neighbor SUT, and set \mathbf{D} is the packet's destination.

UK means no sensed information can be used, so its value is intermediate. The *R*-value of *SR* is 1 because the use of previously successful transmission slots is most advocated. If x is the destination node, the *R*-value of *VC* is only lower than that of *SR*, because the reach of packets when x is idle

is a suboptimal choice; the *R*-value of *ER* is only higher than *UK* because the prediction is unreliable; the values of *EO* and *NO* are 0 because the packet is not allowed to reach the slots occupied by other packets. If x is a nondestination node, the *R*-value of *NO* is higher than 0.5, because causing collisions at nondestination nodes are allowed, which encourages interference alignment. Local nodes are not allowed to send packets during reception; thus, the *R*-value of *RX* is 0. R_n^L of *OH* is 0.5 because it is not prohibited or recommended to send packets during overhearing, which reduces sensed information.

Subsequently, S_n^x represents the value generated for neighbor x by sending the packet in the n th slot in the past, which is referred to as the *S*-value. To calculate S_n^x , we first need to determine the slot when the packet arrives at x through

$$\begin{aligned} -\Delta n \Delta t &\leq \Delta t + \frac{\|P_L^n - P_x^n\|}{c} + t_{\text{off}}^a \\ &= (1 - \Delta n) \Delta t - t_{\text{ctn}}^x < (1 - \Delta n) \Delta t \\ \Delta n &\in \mathbb{Z}, \quad 0 < t_{\text{ctn}}^x \leq \Delta t \end{aligned} \quad (11)$$

where P_L^n is the position of the local node in slot n , and P_x^n is the position of node x that is nearest to slot n . Then we use Δn and t_{ctn}^x to calculate S_n^x

$$S_n^x = \begin{cases} R_{n+\Delta n}^x, & \text{if } t_{\text{ctn}}^x \leq t_g \\ (R_{n+\Delta n}^x + R_{n+\Delta n+1}^a)/2, & \text{if } t_{\text{ctn}}^x > t_g. \end{cases} \quad (12)$$

If the packet occupies two slots on x ($t_x^n > t_g$), then S_n^x is the average of their *R*-values.

Finally, we introduce the predictive transmission value function $F(i)$, $i = 1, \dots, C$, representing the expected future value of sending a packet in the i th time slot after its generation, which is referred to as the *F*-value

$$F(i) = \frac{F_D(i) + F_N(i) + F_L(i)}{3} - \frac{\alpha(i-1)}{C}. \quad (13)$$

Here, $F_D(i)$, $F_N(i)$ and $F_L(i)$ represent the expected values generated for the destination, nondestination neighbors and local node, respectively, whose Formulas are as follows:

$$F_D(i) = \frac{1}{|\mathbf{D}|} \sum_{x \in \mathbf{D}} \frac{\sum_{k=1}^K S_{k \cdot C - i}^x \cdot \gamma^{(k-1)}}{\sum_{k=1}^K \gamma^{(k-1)}} \quad (14)$$

$$F_N(i) = \frac{1}{|\mathbf{A} - \mathbf{D}|} \sum_{x \in \mathbf{A} - \mathbf{D}} \frac{\sum_{k=1}^K S_{k \cdot C - i}^x \cdot \gamma^{(k-1)}}{\sum_{k=1}^K \gamma^{(k-1)}} \quad (15)$$

$$F_L(i) = \frac{\sum_{k=1}^K R_{k \cdot C - i}^L \cdot \gamma^{(k-1)}}{\sum_{k=1}^K \gamma^{(k-1)}} \quad (16)$$

where $K \cdot C$ is the length of the neighbor SUT, $S_{k \cdot C - i}^x$ is the *S*-value of the i th slot of the past k th transmission circle for node x , and $R_{k \cdot C - i}^L$ is the *R*-value of the i th slot of the past k th transmission circle for the local node. $\gamma \in [0, 1]$ is a discount factor that represents the decay of the historical transmission value.

$F(i)$ is apparently obtained through the weighted average of multiple previous *S*-values and *R*-values, and $\sum_{k=1}^K \gamma^{(k-1)}$ in the denominators is used for normalization.

When a node sends a packet to multiple neighbors, $F_D(i)$ is the average predicted transmission value of all transmission

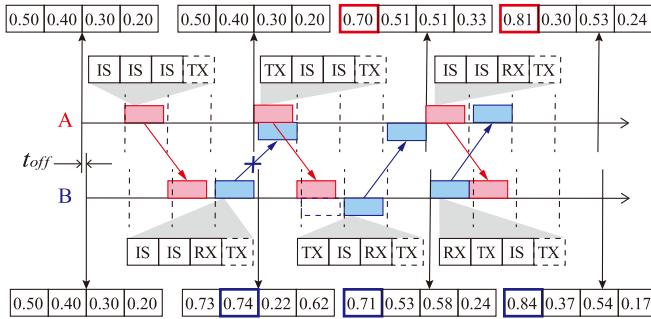


Fig. 6. Example of the schedule based on the predictive transmission value function.

targets. Therefore, the proposed algorithm can be used for multicast transmission.

In addition, $\alpha(i-1)/C$ in (13) indicates the latency cost, and $\alpha \in [0, 0.5]$ is the real-time importance factor, which determines the proportion of the real-time value to the F -value, that is, $\alpha/(1+\alpha)$.

The predictive value function uses the accumulated sensed information to compute the probability of future transmission collisions and consider real-time performance. Whenever a packet is generated, the node chooses $i_s = \text{argmax } F(i)$ as the planned slot and transmits the packet at the beginning of the planned slot.

When the network starts operating, many shared SUTs are lost owing to frequent collisions. A threshold value of $F_{\min} = 0.5$ was used. If the F -values of all the candidate slots are not larger than F_{\min} , the node randomly chooses the planned slot. If the node is busy at the beginning of the planned slot, it randomly selects another slot from the remaining slots with an F -value of larger than F_{\min} , if any. This method offers a certain degree of collision avoidance with minimal sensing information.

As each scheduling requires calculating the F -value for all candidate slots, so the computational complexity for scheduling a packet is indicated by $O(K \cdot C \cdot N)$, where N is the number of nodes within the communication range.

An example of a schedule based on the predictive transmission value function is shown in Fig. 6. Nodes A and B generated a standard packet every four slots and sent it to one another; $\alpha = 0.4$ and $\gamma = 0.8$. When scheduling the first packet, the two nodes had no sensed information; therefore, the F -values of all the four future slots were 0.5, and the second and fourth slots were randomly selected. When scheduling the second packet, node A still had no sensed information and randomly chose the first slot, whereas node B already received a shared SUT and selected the second slot with the maximum F -value. During the second circle, B's packet collided with A's packet at node A, and node B was busy at the beginning of the planned slot; therefore, it randomly selected the third slot to send. When scheduling the third packet, node A already received a shared SUT, and the slots with the maximum F -values of A and B both became the first slot. As the two packets were transmitted without collision during the third circle, the F -value of the first slot increased, and the future schedules converged. If no change occurs, the F -values will converge at [1.0, 0.24, 0.64, 0.30].

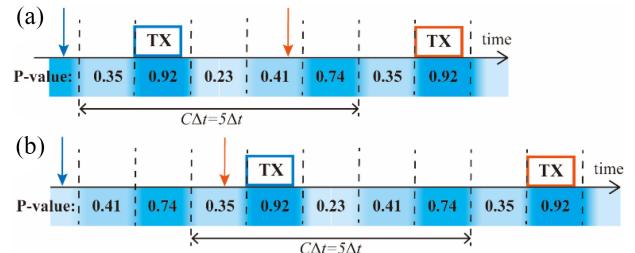


Fig. 7. Two examples of transmission of packets with not fixed generation interval.

This distributed scheduling algorithm has randomness and cannot guarantee an optimal schedule each time, but it can ensure that the converged scheduling is collision-free and can provide substantial opportunities for concurrent transmissions.

Coefficient α and γ affect the scheduling preferences. A larger α results in more concurrent transmission attempts to achieve a lower latency, but it increases the likelihood of collisions during convergence. γ affects the response speed of network changes. The specific impact of α and γ were analyzed through simulations, which is presented in Section VI.

B. Modification for Particular Scenarios

In practical scenarios, the duration of the packets may not be fixed, and their generated time interval may change. Therefore, we modified the aforementioned sending strategy for follow particular scenarios.

1) *The Generation Interval of Packets is Not Fixed:* For example, the sensor of a UUV only generates a packet when certain data are collected. The generation intervals of these data packets are not fixed but have expected or average values.

Nodes can still select transmission slots by comparing the F -value using (14)–(16). However, the random generation time affects the latency cost in (13), which may cause fluctuations in the F -value and render the scheduling unable to converge.

Suppose the expected interval of a packet generation is T_e , such that $C = \lceil T_e/\Delta t \rceil$. When a node generates a packet, (13) is used to calculate the F -values of the future C slots considering $\alpha = 0$. The slot with the largest F -value is selected as the planned transmission slot. As shown in Fig. 7(a), $C = 5$, the generation interval of the two packets is less than the expected value; however, they will both be sent at the slots with the largest F -value.

Note, if a node generates another packet while waiting to transmit, the node will wait to send the current packet before scheduling the next packet. In this manner, the transmission interval remains the same, but the latency fluctuates, occasionally causing significant delays. As shown in Fig. 7(b), the node selects the planned slot of a packet (red) after transmission of the previous packet (blue), and the wait time exceeds the expected interval.

The node should use a small α to reduce the latency and ensure transmission convergence. In this case, although the F -values may fluctuate, the maximum F -value remains concentrate in a few fixed slots within a cycle. This is demonstrated in the simulation experiments.

2) *Occasional Packets*: There is another type of packet whose generation interval has no expected value, which is referred to as an occasional packet. For example, a UUV sends an alarm after detecting the mission target. Usually, such packets have the highest priority. Therefore, if a node generates an occasional packet, it will use the nearest slot planned to be occupied by a general packet. If the node has no other packets to send, the packet will be sent in the same manner as introduced above, and C is the default value.

3) *Long Packets*: A long packet with a duration of $l_D > \tau$, is divided into $k = \lceil l_D / \tau \rceil$ standard packets for transmission. Therefore, the node must select k slots from the future C slots for transmission. The same method is used to calculate the F -value for each slot and to select the slots with the top k largest F -values. The slots are then allocated, and the sending order of these standard packets is consistent with their positional order in the original packet.

C. Adaptive Adjustment of Local Load

The scheduling algorithm based on the predictive value function can adaptive adjustment of the local load according to sensed traffic. The specific operation is as follows.

1) Increase Load in Good Condition and Low Traffic:

The specific conditions for increasing load are as follows. A node transmitting k standard packets in each circle, if these transmissions in the last N_{In} cycles meet the following conditions.

- 1) Number of neighbors in the neighbor SUT remains unchanged.
- 2) The success rate of these transmissions reaches $\sigma\%$ (determined by SR).
- 3) When calculating the F -value, the $(k+1)$ th largest value is greater than F_{min} .

then the node is allowed to send $k + 1$ packets in the next cycle.

2) Decrease Load in Bad Condition and High Traffic:

A node transmitting k standard packets in each circle, if these transmissions in the last N_{De} cycles met all following conditions.

- 1) The success rate of these transmissions is lower than $\rho\%$.
- 2) When calculating the F -value, the k th largest value is lower than F_{min} .

then the number of packets to be sent in the next cycle must be decreased by one.

A smaller N_{In} and larger N_{De} results in a greater the total load; however, the number of collisions will increase. In addition, N_{In} and N_{De} can significantly affect the speed of the protocol response by changes in the network load. Smaller values of N_{In} and N_{De} result in a shorter interval between the nodes changing the load. However, a rapid load change has adverse effects on the schedule convergence. $\sigma\%$ can be set to be slightly lower than the average success rate of the point-to-point transmission of standard packets in a practical environment; $\rho\%$ can be set as the minimum average success rate for task requirements.

Adaptive load adjustment can enable each node to cope with changes in the number of nodes, network density, and traffic demand while improving the network throughput and avoiding channel congestion.

VI. SIMULATION AND DISCUSSIONS

The performance of our proposed MAC protocol was evaluated and analyzed through simulations.

A. Simulation Settings

We conducted simulations on the ns-3 simulator and used the Thorp's formula to calculate the signal attenuation. We used the model of interference calculation based on the FH-FSK model [33]. To facilitate the statistics of packet loss caused by collisions, we use the ideal PER model in which no errors occur if the SNR of the received signal is larger than the threshold of the transducer. The transmission rate of each node is 4 Kb/s, the transmission power is 131 dB re μ Pa, and the communication range is approximately 1500 m.

Because this study did not consider the routing protocol, we used a partitioned-domain ad hoc network in a $4000 \times 4000 \times 200$ m³ sea area, which was divided into four $2000 \times 2000 \times 200$ m³ domains, in which a sink node was located at the center of the surface and N UUVs were randomly placed. There were two-way communications in a domain where each vehicle uploaded sensor data to the sink node, and the sink node transmitted commands to the UUVs. Thus, these $4(N+1)$ nodes formed four partially overlapping collision domains with single hop communications.

Each vehicle remained at a random depth with a fixed speed of v (m/s), and moved by a random direction mobility model, where each object randomly selected a direction every 100 s or when it reaches one of the boundaries of its domain.

B. Comparative Experiment

We first conducted comparative experiments using slotted FAMA [18], UAN-CW [16], CS-MAC [27], and UW-ALOHA-QM [28], all of which are distributed MAC protocols for UANs. In the experiment, each UUV generated a packet of 200 bytes every T second and uploaded it to the sink node. Because Slotted FAMA, UW-ALOHA-QM and CS-MAC only support unicast transmission, in this experiment, each sink node generates a packet of 200 bytes and randomly chose a UUV in its domain as the destination. The UUV's velocity is $v = 2$ m/s. All the protocols operated without retransmission in this experiment; therefore, slotted-FAMA, UAN-CW and CS-MAC did not use ACK packets. Only UW-ALOHA-QM used ACK packets for feedback and learning. Because CS-MAC does not support bidirectional communication, we let the sink nodes to send command packet at the moment originally used to send the ACK packets.

The performance indices include the average packet delivery rate (PDR) and average latency (time interval from packet generation to reception). We adjusted the following two network parameters: 1) the packet generation interval T and 2) the number N of UUVs in a domain.

UAN-CW has two protocol parameters, which are the window length σ and maximum window number w , and we tried different parameter pairs and recorded the best performance. Slotted FAMA also has two parameters: the slot length Δt_S is composed of the maximum propagation delay, control packet duration, and guard time. In this experiment, $\Delta t_S = 1.1$ s and the backoff time was $3\Delta t_S$. The slot length Δt_Q used in UW-ALOHA-QM consists of two maximum propagation delays, a packet duration, an ACK packet duration, and guard time. In this experiment, $\Delta t_Q = 2.5$ s, the number of slots per frame $C_Q = \lceil T/2.5 \rceil$, and the learning rate was 0.1, and the reward $r = 1$. For CS-MAC, the duration of a control packet was 10 ms, and the timer in a handshake phase was 2 s. The time slot length used by DCSM was only composed of the packet duration ($\tau = 0.4$ s) and guard time ($t_g = 0.1$ s), thus $\Delta t = 0.5$ s, and the number of slots per frame $C_D = \lceil T/0.5 \rceil$. The real-time importance factor $\alpha = 0.05$ and discount factor $\gamma = 0.8$.

The packet generation interval is fixed, but the time at which each node generated the first packet was random between zero and T . We conducted 100 repeated experiments using the same network parameters, each lasting for $200 T$.

1) Comparison Under Different Network Traffic Loads: First, we fixed the number N of vehicles in each domain to four and compared the performance under different network loads by adjusting T from 25 s to 10 s.

Fig. 8(a) and (b) show the PDR of the transmissions from the UUVs and sink nodes. The PDR achieved by Slotted-FAMA is the worst because the transmission delay is too long, as shown in Fig. 9(a), and many transmissions were not completed within T . Another handshake-based protocol, CS-MAC, achieved a better PDR than slotted FAMA, because it utilized concurrent scheduling in the UUV transmission. It achieved a significantly high PDR for transmissions from the sink nodes because the transmission had exclusive access. The PDR of the UUV transmission achieved by UW-ALOHA-QW was the second highest, but the PDR of the sink nodes transmission was poor. This may be because the sink nodes occasionally randomly selected the transmission target and the scheduling did not converge timely. The UAN-CW based on the carrier sense and random backoff mechanism, achieved a moderate PDR, which was obtained by testing different protocol parameters for various network parameters. The DCSM proposed in this study achieved the highest average PDR and had an evident advantage in UUV transmission. PDRs $> 90\%$ were achieved when $T > 15$ s. In addition, compared with other protocols, the 95% confidence intervals of the PDRs were small, indicating that DCSM can achieve a stable performance under the dynamic network.

Fig. 9 shows the average delay and throughput of the transmissions from all the nodes. The delay obtained by slotted-FAMA was apparently the longest. The delay began to decrease after T was less than 20 s because many transmissions that suffered long waiting times could not be completed within T and were discarded. The delay of UW-ALOHA-QW and CS-MAC positively correlated with T , and the delay was significant when T was large because their scheduling

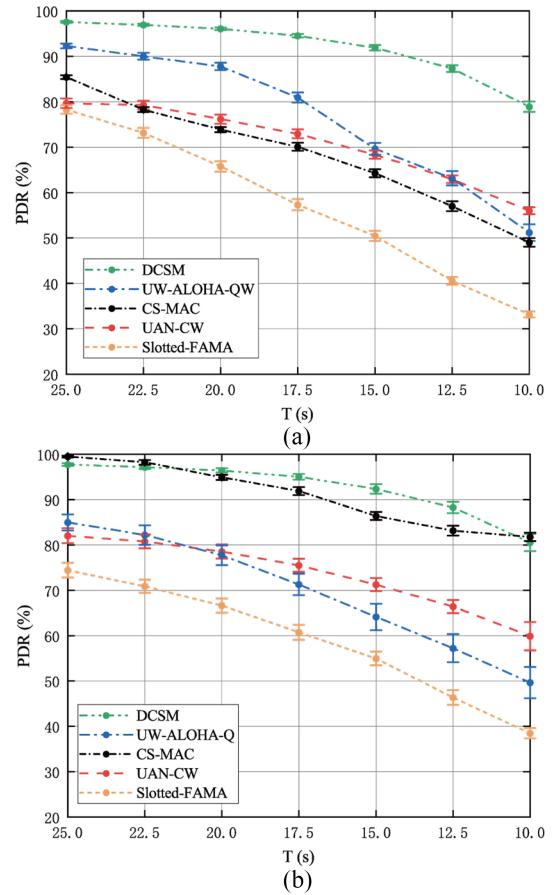


Fig. 8. Variations in the PDR obtained by the five protocols with the packet generation interval T when $N = 4$. (a) PDR of the transmissions from the UUVs. (b) PDR of the transmissions from the sink nodes.

algorithms did not consider the latency. The delay of UAN-CW was minimal because it is a kind of nonpersistent CSMA, and its timer pauses instead of resetting when the channel is busy during the waiting state. The delay of DCSM was similar to that of UAN-CW; however, in this comparative experiment, DCSM used a fixed and small real-time importance factor α .

As shown in Fig. 9(b), slotted FAMA achieved the worst throughput, DCSM achieved the best throughput, and the other protocols achieved similar throughputs. And the throughput of DCSM when $T = 10$ s was 41% higher than that of the second place.

2) Comparison Under Different Network Densities: We then fixed T at 25 s and compared the performances of the five protocols under different network densities by increasing the number N of vehicles in a domain.

Fig. 10 shows the variation of the average PDRs with N . First, the PDRs of the sink node transmissions obtained by the five protocols are higher than that of the UUV transmission. The destinations of the UUVs were concentrated at the four sink nodes, which were more prone to signal collisions, whereas the destinations of sink nodes did not overlap. CS-MAC performed poorly in the transmission of UUVs because the collision of the control packets increased with N , significantly affecting its scheduling algorithm. The PDR performance of DCSM was the best in the UUV transmission

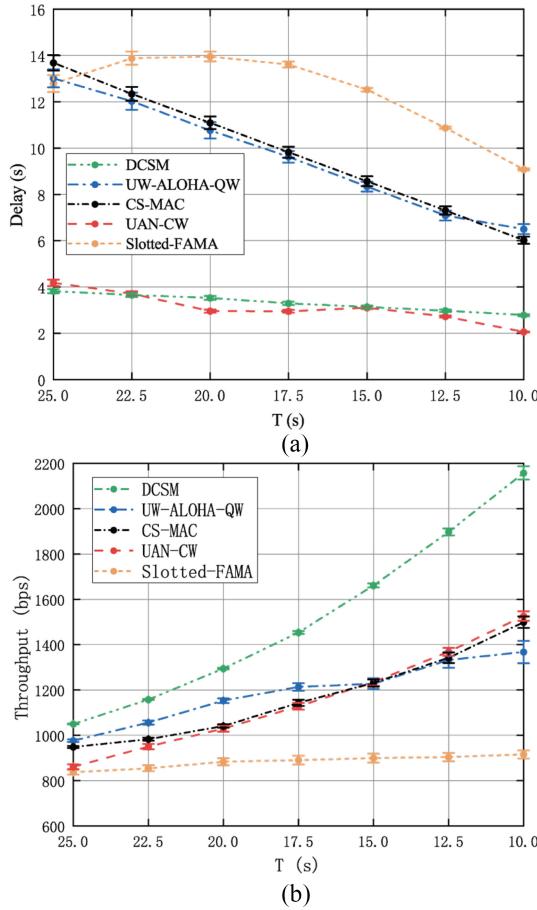


Fig. 9. Variations in the (a) delay and the (b) throughput obtained by the five protocols with the packets' generation interval T when $N = 4$.

and similar to that of CS-MAC in the sink node transmission. Although the PDR also decreased as N increased, it exceeded 70% when $N = 10$.

Fig. 11 shows the variations in the average delay (a) and throughput (b) with N . The delay of slotted-FAMA increased with the increase of N because each node suffered from more waiting time. The delay of UW-ALOHA-QW and CS-MAC is related only to T and therefore, demonstrates no significant change. The delay of UAN-CW did not increase as we adjusted the number of waiting windows w . The delay of DCSM increased slightly because the number of nonidle slots of a node increased with N . In terms of the throughput, DCSM still performed the best and had obvious advantages.

In conclusion, compared with the other four protocols, DCSM achieved high PDRs (15% higher in average) and low latency in both UUV and sink node transmissions, and had a significant throughput advantage (35% higher in average). The performance was stable under the mobile network.

When each node had many idle slots (T was long or N was small), DCSM made the PDR near 100%, and packet collisions occurred only during the learning phase (at the beginning of an experiment or when the topology changed). As the number of idle slots decreased (T decreased or N increased), the duration of the learning phase increased, resulting in a slow increase in the packet loss rate (PLR). When the number of idle slots was zero, the learning time was too long for

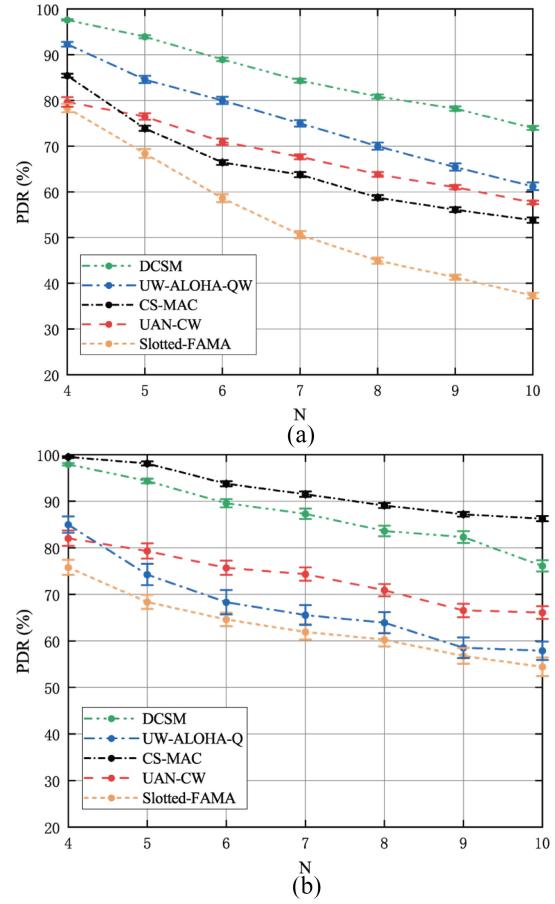


Fig. 10. Variations in the PDR obtained by the five protocols with the number N of UUVs in each domain when $T = 25$ s. (a) PDR of transmissions from the UUVs. (b) PDR of transmissions from the sink nodes.

the schedule to converge in the mobile network; despite the schedule occasionally converging, collisions remained to occur owing to the incomplete information sensed by the nodes. But DCSM still ensured a good PDR in that case.

Because DCSM does not require any exchange of control packets and the time slot used is only composed of the packet duration and guard time, supporting spatial-temporal reuse, DCSM can achieve very high-channel utilization. Considering the real-time performance, the latency achieved by DCSM was significantly lower than those of the other two scheduling-based protocols.

Due to the absence of retransmission in the above simulation, DCSM did not use ACK packets. If necessary, DCSM supports sending ACK packets to achieve shorter ACK delay than the inherent implicit ACK mechanism. Because DCSM supports sending multiple packets within one cycle.

C. Impact of Parameters α , γ , and t_g

Then we analyzed the impact of the DCSM's parameters (α , γ and t_g) on the performance through experiments. The simulation settings were consistent with those described in Section A.

1) *The Impact of the Real-Time Factor α :* First, we only changed α and fixed the other parameters to analyze the impact of α on the PDR and latency. In this experiment,

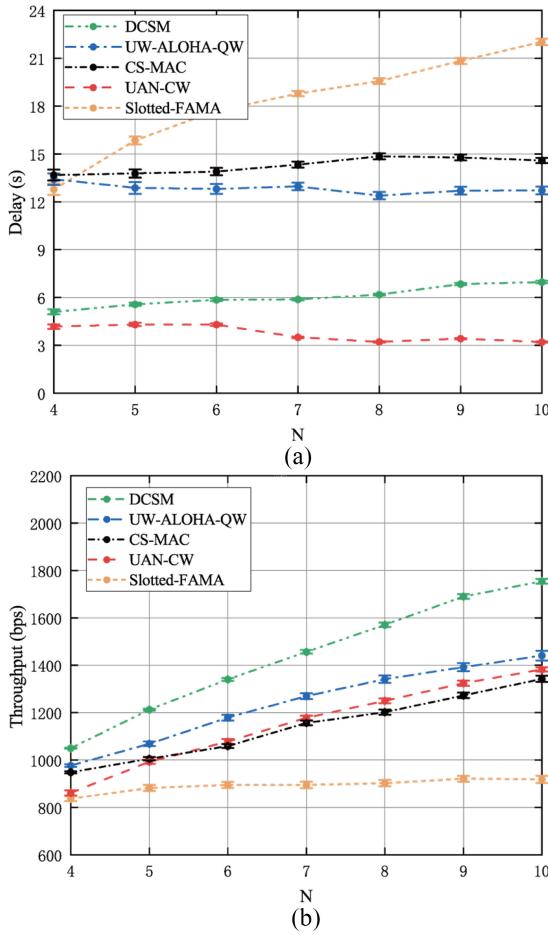


Fig. 11. Variations in the (a) delay and the (b) throughput obtained by the five protocols with the number N of UUVs in each domain when $T = 25$ s.

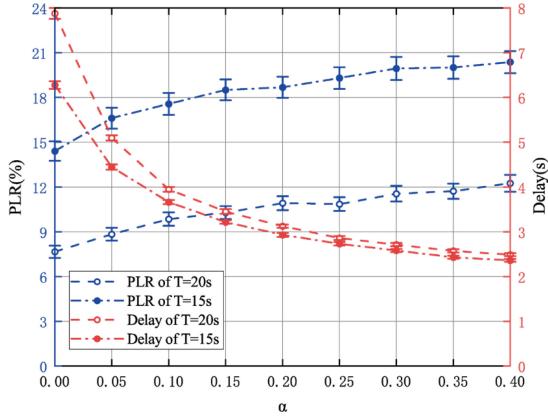


Fig. 12. Variations in the PLR and the delay obtained by DCSM with the parameter α when $N = 5$, $\gamma = 0.8$, and $T = 15, 20$ s.

$T = 20$ s and 15 s, $N = 5$, $\gamma = 0.8$, $t_g = 0.1$ s, and α was increased from 0 to 0.4. Each value was repeated 100 times, each lasting 3600 s. The results are presented in Fig. 12. For convenience, we replaced PDR with the PLR in the figure. Apparently, increasing α slowly increased the PLR because a larger α results in a larger convergence time. When α was small, increasing α significantly reduced the delay, and the delay decreased significantly slowly when α was large. Thus, there is a tradeoff when choosing α , and a range of 0.1 to 0.3 appears to be a good choice.

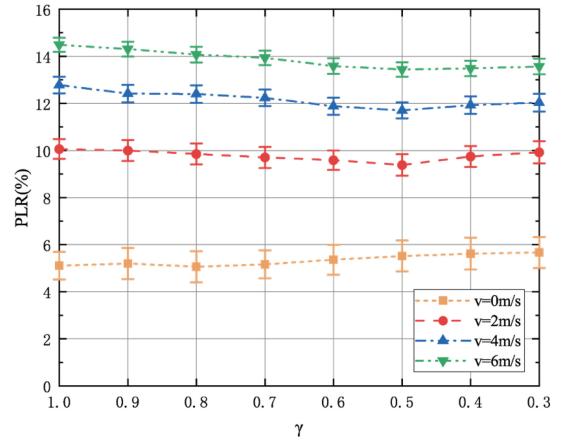


Fig. 13. Variations in the PLR obtained by DCSM with the parameter γ and UUVs' velocity v when $N = 5$ and $T = 20$ s.

2) *The Impact of the Discount Factor γ :* Subsequently, we fixed $\alpha = 0.1$, $T = 20$, and $N = 5$, and changed γ from 1 to 0.3 under different UUV velocities. The results are shown in Fig. 13. Apparently, the faster the vehicles, the higher the PLR, which cannot be solved because despite the sensed information, the nodes cannot predict changes in the network topology. When $v = 0$ m/s, the PLR increased slowly with γ decreasing; when $v > 0$ m/s, the PLR first decreased and then increased. A larger γ changed the F -value slower and was not suitable for a rapid change of topology, whereas a smaller γ made the F -value more prone to fluctuations. Although the impact of γ on PLR is smaller than α , an appropriate value of γ should be chosen. A value within 0.5–0.7 is suitable for both static and mobile networks.

3) *The Impact of Guard Time t_g :* The guard time in the DCSM is used to accommodate the effects of changes in the sound speed, positional error, and UUVs movement on the estimation of the propagation delay. As the first two factors had similar effects, we analyzed the PDR and throughput with different t_g values for different positioning errors. The following parameters were set: $\alpha = 0.1$, $\gamma = 0.7$, $v = 2$ m/s, $T = 15$ s, and $N = 5$. To maintain the same slot length, we increased t_g while reducing the standard packet length. For example, when $t_g = 0.12$ s, the packet length was 190 bytes. The added horizontal positioning error followed a uniform distribution between $[-e, e]$. The results are presented in Fig. 14, which demonstrates that increasing the proportion $t_g/\Delta t$ can effectively increase the PDR, but the side effect is a rapid decrease in the throughput. When the proportion $t_g/\Delta t$ increased from 12% to 40%, the PDR increased by approximately 8%, but the throughput decreased by approximately 30%. In addition, the positioning error had a small impact on the performance. Despite $e = 80$ m, the PDR and throughput only decreased by approximately 2% and 3%, respectively.

D. Simulation for Special Scenario

Finally, we conducted two sets of simulations under particular scenarios to verify the adaptability and flexibility of DCSM.

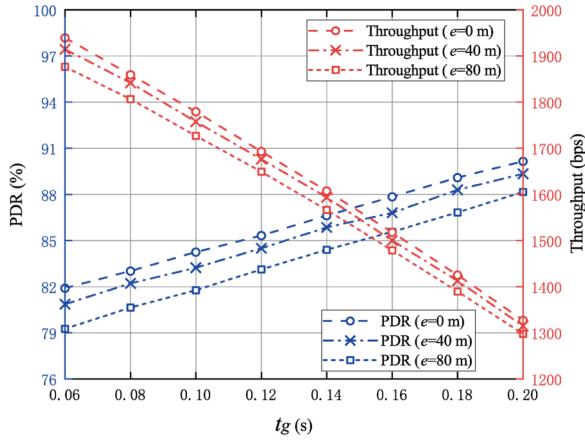


Fig. 14. Variations in the PDR and the throughput obtained by DCSM with the parameter t_g when $N = 5$ and $T = 15$ s.

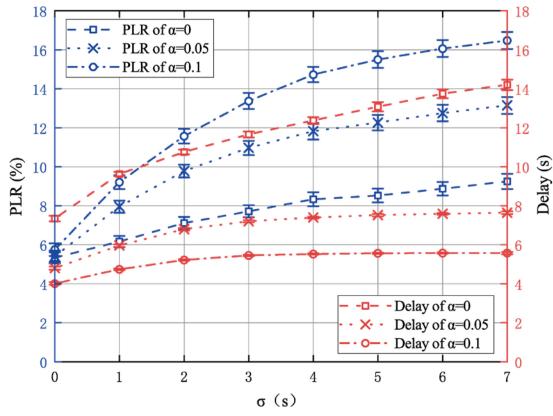


Fig. 15. Variations in the PLR and delay obtained by DCSM with the standard deviation σ of T when $N = 5$ and $T_{\min} = 20$ s.

1) The Impact of the Random Packet Generation Interval:

In Section V-B, we explain how DCSM is applied to cases where the packet generation interval is not fixed. In this set of simulations, we used the same settings as the previous experiments, but generation interval T was changed to follow a normal distribution; its mean value was $T_{\text{mean}} = 20$ s; and $N = 5$, $\gamma = 0.8$, and $v = 2$ m/s. The standard deviation σ of T was statistically analyzed in the experiment; the impact on PLR and delay are shown in Fig. 15. The increase of the standard deviation σ apparently resulted in an increase in both PLR and delay. As previously indicated, compared to a fixed T , a random T requires a smaller α because when $\alpha = 0.1$ and σ is large, the PLR increases by more than 10%. When $\alpha = 0$ and $\sigma = 7$ s, PLR only increases by 4%, but the delay increases by 7 s. Therefore, for a random T , α is recommended to be between 0 and 0.1. If σ is small, a value near 0.1 can be used to achieve a low latency; if σ is large, it is necessary to choose a relatively small α to ensure reliability.

Although a random T has a negative impact on DCSM, which is also a problem for all slot-based scheduling protocols, DCSM still performed sufficiently in this case.

2) The Impact of the Variable Network Load: In the second simulation, the sink nodes changed communication mode to multicast, and the load of the UUVs changed as the experiment progressed. The specific scenario was as follows: $N = 4$, $T = 25$ s, $v = 2$ m/s, and $N_{\text{In}} = N_{\text{De}} = 8$ (in Section V-C). At the beginning of the experiment, each UUV generated one packet every T s and sent it to the sink node. Each sink node generated one data packet every T s and sent it to all UUVs in its domain. In the first phase, a specific UUV in each domain increased the number of generated packets by one every 4000 s until generating five packets every T s and then returned to one packet every T s after 4000 s. In the second phase, every 4000 s, there was a UUV in each domain whose number of generated packets every T s increased from one to two until all the UUVs generated two packets every T s. We recorded the number of packets generated by all UUVs, the number of packets transmitted by all UUVs, and the number of packets received by the sink nodes every 400 s. The changes in the PLRs of the UUV unicast transmission and sink node multicast transmission were also recorded, as shown in Fig. 16.

DCSM apparently allowed the nodes to change load during operation and maintained a low PLR (approximately 5%). When the channel was relatively idle, DCSM caused the number of transmission packets to follow the number of generation packets. When the load was too high, DCSM limited the number of packets sent by the UUVs to avoid channel congestion and ensure a high-transmission success rate. Although the PLR significantly increased when the number of transmission packets changed, DCSM adapted to the change in the load and stabilized the PLR. In addition, although the PLR of the multicast transmission was significantly higher than that of the unicast, it remained at approximately 20%, proving that DCSM can be simultaneously used for unicast and multicast transmissions. Note, the experiment demonstrated that the performance of DCSM in the concentration load (first stage) was not as good as in the dispersion load (second stage), as the maximum throughput in the second phase was significantly higher than that in the first phase. A node with higher load schedules more transmissions in a cycle and has more difficulty of scheduling convergence; thus, it is more likely to decrease the local load.

VII. CONCLUSION

This study proposes a MAC protocol for multi-UUV underwater acoustic networks, called the DCSM. Considering the characteristics of underwater acoustic communication and multi-UUV networks, the DCSM adopts a distributed channel sensing method to avoid signal collisions. DCSM supports spatial-temporal reuse by passively sharing slot usage tables. It adopts a scheduling algorithm based on a predictive transmission value function to converge quickly and cope with various scenarios. In simulation experiments, compared with four existing MAC protocols for UANs (UAN-CW, Slotted FAMA, CS-MAC and UW-ALOHA-QW), DCSM achieved the highest transmission success rate in UUV networks while ensuring a low latency. Because DCSM does not require a control packet

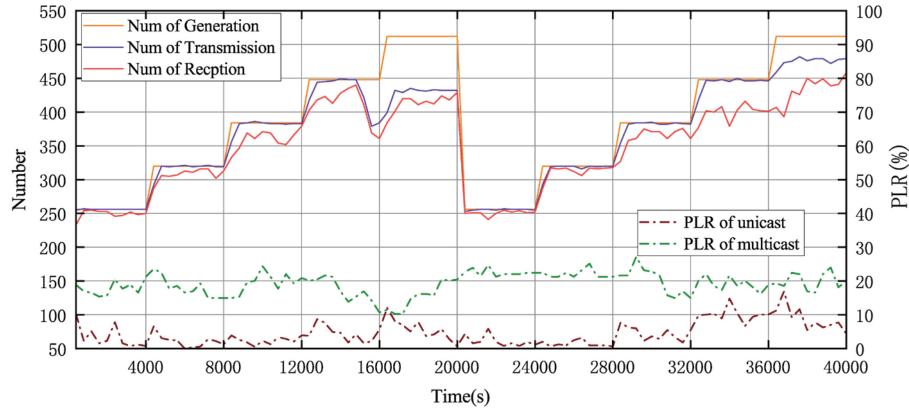


Fig. 16. Variations in the network traffic and the PLR as the network load changed.

exchange and uses a notably short time slot, it has a significant advantage in terms of the throughput. The impact of the DCSM parameters on the performance was also analyzed through simulation experiments, providing a reference for parameter selection under different performance requirements. Finally, special experiments verified the flexibility and adaptability of DCSM and proved that it supposes both unicast and multicast transmission, and can be used in a network with variable traffic loads. However, two drawbacks of the DCSM were discovered in the experiment. First, if the randomness of the transmission interval is too strong, the performance will significantly decrease; second, the throughput under a concentration load is not as good as that under a dispersion load. In the future, we expect to use DCSM in the “Explore-100” AUV platform [1] and conduct sea trials.

ACKNOWLEDGMENT

The authors would also like to thank the Explore-100 AUV team.

REFERENCES

- [1] L. Li, Y. Li, Y. Zhang, G. Xu, J. Zeng, and X. Feng, “Formation control of multiple autonomous underwater vehicles under communication delay, packet discreteness and dropout,” *J. Marine Sci. Eng.*, vol. 10, no. 7, p. 920, Jul. 2022.
- [2] T. Qiu, Y. Li, J. Zeng, G. Xu, and X. Feng, “A underwater acoustic communication network for multi-AUV cooperative target location,” in *Proc. 6th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, 2021, pp. 535–540.
- [3] G. Indiveri, “Geotechnical surveys with cooperative autonomous marine vehicles: The EC WiMUST project,” in *Proc. IEEE/OES Autonom. Underwater Veh. Workshop (AUV)*, 2018, pp. 1–6.
- [4] A. Caiati, V. Calabro, G. Dini, A. L. Duca, and A. Munafò, “Secure cooperation of autonomous mobile sensors using an underwater acoustic network,” *Sensors*, vol. 12, no. 2, pp. 1967–1989, 2012.
- [5] O. Bello and S. Zeadally, “Internet of Underwater Things communication: Architecture, technologies, research challenges and future opportunities,” *Ad Hoc Netw.*, vol. 135, Oct. 2022, Art. no. 102933.
- [6] K. Y. Islam, I. Ahmad, D. Habibi, and A. Waqar, “A survey on energy efficiency in underwater wireless communications,” *J. Netw. Comput. Appl.*, vol. 198, Feb. 2022, Art. no. 103295.
- [7] Y. Yang, Y. Xiao, and T. S. Li, “A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 815–841, 2nd Quart., 2021.
- [8] C. C. Hsu, M. S. Kuo, C. F. Chou, and K. C. J. Lin, “The elimination of spatial-temporal uncertainty in underwater sensor networks,” *IEEE-ACM Trans. Netw.*, vol. 21, no. 4, pp. 1229–1242, Aug. 2013.
- [9] W. Chen, Q. Guan, H. Yu, F. Ji, and F. Chen, “Medium access control under space-time coupling in underwater acoustic networks,” *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12398–12409, Aug. 2021.
- [10] Y. Noh, P. Wang, U. Lee, D. Torres, and M. Gerla, “DOTS: A propagation delay-aware Opportunistic MAC protocol for underwater sensor networks,” in *Proc. 18th IEEE Int. Conf. Netw. Protocols*, Kyoto, Japan, 2010, pp. 183–192, doi: [10.1109/ICNP.2010.5762767](https://doi.org/10.1109/ICNP.2010.5762767).
- [11] M. Chitre, M. Motani, and S. Shahabudeen, “Throughput of networks with large propagation delays,” *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 645–658, Oct. 2012.
- [12] N. Zhao, N. Yao, and Z. Gao, “A message transmission scheduling algorithm based on time-domain interference alignment in UWANs,” *Peer-to-Peer Netw. Appl.*, vol. 14, no. 3, pp. 1058–1070, May 2021.
- [13] S. Jiang, “State-of-the-art medium access control (MAC) protocols for underwater acoustic networks: A survey based on a MAC reference model,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 96–131, 4th Quart., 2018.
- [14] T. Qiu, Y. Li, and X. Feng, “Optimal broadcast scheduling algorithm for a multi-AUV acoustic communication network,” *IEEE ACM Trans. Netw.*, vol. 31, no. 5, pp. 2058–2069, Oct. 2023.
- [15] M. Liu et al., “Adaptive scheduling MAC protocol in underwater acoustic broadcast communications for AUV formation,” *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6887–6901, Apr. 2023.
- [16] S. Ansari, J. Poncela Gonzalez, P. Otero, and A. Ansari, “Analysis of MAC strategies for underwater applications,” *Wireless Pers. Commun.*, vol. 85, no. 2, pp. 359–376, Nov. 2015.
- [17] Y. Guo, M. S. Obaidat, J. Guo, T.-Y. Wu, and F. J. Tey, “Adaptive varying contention window MAC protocol based on underwater acoustic propagation delay,” *IEEE Sensors J.*, vol. 23, no. 13, pp. 15019–15031, Jul. 2023.
- [18] M. Molins and M. Stojanovic, “Slotted FAMA: A MAC protocol for underwater acoustic networks,” in *Proc. Oceans Asia Pac.*, 2006, pp. 1–7.
- [19] S. H. Park, P. D. Mitchell, and D. Grace, “Reinforcement learning based MAC protocol (UW-ALOHA-Q) for underwater acoustic sensor networks,” *IEEE Access*, vol. 7, pp. 165531–165542, 2019.
- [20] A. A. Syed, W. Ye, and J. Heidemann, “Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1731–1743, Dec. 2008.
- [21] X. Zhuo, H. Yang, M. Liu, Y. Wei, G. Yu, and F. Qu, “Data concurrent transmission MAC protocol for application oriented underwater acoustic sensor networks,” *China Commun.*, vol. 19, no. 10, pp. 220–237, Oct. 2022.
- [22] M. Zheng, W. Ge, X. Han, and J. Yin, “A spatially fair and low conflict medium access control protocol for underwater acoustic networks,” *J. Marine Sci. Eng.*, vol. 11, no. 4, p. 802, 2023.
- [23] Y. Ma, H. Wu, F. Dan, X. Yang, B. Yan, and F. Zhou, “A handshake-based low delay MAC protocol in hybrid topology for urban lake monitoring,” *IEEE Sensors J.*, vol. 22, no. 21, pp. 21342–21354, Nov. 2022.
- [24] R. Zhang, X. Cheng, X. Cheng, and L. Yang, “Interference-free graph based TDMA protocol for underwater acoustic sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4008–4019, May 2018.

- [25] M. Liu, X. Zhuo, Y. Wei, Y. Wu, and F. Qu, "Packet-level slot scheduling MAC protocol in underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8990–9004, Jun. 2021.
- [26] A. Roy and N. Sarma, "A synchronous duty-cycled reservation based MAC protocol for underwater wireless sensor networks," *Digit. Commun. Netw.*, vol. 7, no. 3, pp. 385–398, Aug. 2021.
- [27] Y. Shang and X. Du, "CS-MAC: A concurrent scheduling MAC protocol for underwater acoustic networks," in *Proc. Int. Conf. Neural Netw., Inform., Commun. Eng.*, 2022, pp. 416–422.
- [28] S. H. Park, P. D. Mitchell, and D. Grace, "Reinforcement learning based MAC protocol (UW-ALOHA-QM) for mobile underwater acoustic sensor networks," *IEEE Access*, vol. 9, pp. 5906–5919, 2021.
- [29] X. Geng and Y. R. Zheng, "Exploiting propagation delay in underwater acoustic communication networks via deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10626–10637, Dec. 2023, doi: [10.1109/TNNLS.2022.3170050](https://doi.org/10.1109/TNNLS.2022.3170050).
- [30] X. Ye, Y. Yu, and L. Fu, "Deep reinforcement learning based MAC protocol for underwater acoustic networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 5, pp. 1625–1638, May 2022.
- [31] E. P. M. C. Junior, L. F. M. Vieira, and M. A. M. Vieira, "UW-SEEDEX: A pseudorandom-based MAC protocol for underwater acoustic networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3402–3413, Sep. 2022.
- [32] S. Climent, A. Sanchez, J. Capella, N. Meratnia, and J. Serrano, "Underwater acoustic wireless sensor networks: Advances and future trends in physical, MAC and routing layers," *Sensors*, vol. 14, no. 1, pp. 795–833, Jan. 2014.
- [33] N. Parrish, L. Tracy, S. Roy, P. Arabshahi, and W. L. J. Fox, "System design considerations for undersea networks: Link and multiple access protocols," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1720–1730, Dec. 2008.