

Research Article

A MAC Protocol of Concurrent Scheduling Based on Spatial-Temporal Uncertainty for Underwater Sensor Networks

Xin Liu ^{1,2}, Xiujuan Du ^{1,3}, Meiju Li,¹ Lijuan Wang ¹ and Chong Li¹

¹Computer Department, Qinghai Normal University, Xining 810008, China

²Computer Department, Qinghai Nationalities University, Xining 810007, China

³Academy of Plateau Science and Sustainability, Xining 810008, China

Correspondence should be addressed to Xiujuan Du; dxj@qhnu.edu.cn

Received 15 January 2021; Revised 3 March 2021; Accepted 10 March 2021; Published 28 March 2021

Academic Editor: Xavier Vilanova

Copyright © 2021 Xin Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Underwater sensor networks (UWSNs) are characterized by large energy consumption, limited power supply, low bit rate, and long propagation delay, as well as spatial-temporal uncertainty, which present both challenges and opportunities for media access control (MAC) protocol design. The time-division transmissions can effectively avoid collisions since different nodes transmit packets at different period of time. Nevertheless, in UWSNs with long propagation delay, in order to avoid collisions, the period of time is subject to be long enough, which results in poor channel utilization and low throughput. In view of the long and different propagation delay between a receiving node and multiple sending nodes in UWSNs, as long as there is no collision at the receiving node, multiple sending nodes can transmit packets simultaneously. Therefore, in this paper, we propose a MAC protocol of concurrent scheduling based on spatial-temporal uncertainty called CSSTU-MAC (concurrent scheduling based on spatial-temporal uncertainty MAC) for UWSNs. The CSSTU-MAC protocol utilizes the characteristics of temporal-spatial uncertainty as well as long propagation delay in UWSNs to achieve concurrent transmission and collision avoidance. Simulation results show that the CSSTU-MAC protocol outperforms the existing MAC protocol with time-division transmissions in terms of average energy consumption and network throughput.

1. Introduction

Recently, underwater sensor networks (UWSNs) have attracted significant research interest because they promise a broad range of applications such as underwater rescuing, offshore mining, offshore exploration, environmental monitoring, and pollutant content detection [1–4]. The media access control (MAC) protocol has become one of the study focuses in the field of UWSNs [5, 6]. Although UWSNs are a type of wireless sensor network, they are different from the terrestrial wireless sensor networks (WSNs) in which the sensor nodes rely on radio waves to communicate with one another. The underwater sensor nodes use acoustic channels as the communication medium with a speed of 1500 m/s, which is five orders of magnitude lower than that of radio signals; thus, the MAC protocols designed for terrestrial sensor networks are not suitable for UWSNs. The underwater acoustic channel has the characteristics of long

propagation delay, high bit error rate, and low bandwidth [7–9]. Moreover, in addition to high delay and low bandwidth, energy consumption is another major problem. In UWSNs, acoustic sensor nodes usually use batteries as their power supply, and it is difficult to recharge or replace batteries. To provide necessary power for long-term sensing, data collection, and communication, it is highly important to study the efficient and energy-saving wireless sensor network protocol [10–12]. The design of the MAC protocol has become a challenging task in underwater wireless sensor networks [13].

In the design of underwater MAC protocols, the spatial-temporal uncertainty problem is subject to be considered. Due to the time uncertainty (i.e., different transmission time), the receiving time is uncertain; therefore, a collision can be caused at the receiver. Therefore, to avoid collisions, reasonable scheduling of sending and receiving should be considered.

In view of the above problems, some researchers have proposed MAC protocols based on a handshake approach. In this approach, the communication between nodes is reserved and advertised by sending control packets. To date, of the handshake-based MAC protocols, multiple access with collision avoidance (MACA) [14] and floor acquisition multiple access (FAMA) [15] are two classic MAC protocols. The MACA protocol uses a handshake mechanism based on RTS/CTS, which can effectively reduce collisions. However, it is designed for a radio wave network and cannot be successfully used in UWSNs. The FAMA protocol lengthens the transmission delay of RTS/CTS to enable it to operate in UWSNs with long propagation delays. However, the cost of energy consumption for transmitting such long control packets is notably high. Although this method reduces the probability of collision, it consumes a significant amount of energy. While all the contention-based MAC protocols are expected to improve the performance of a network, several works have shown that long propagation delay, narrow bandwidth, and high bit error rate in underwater environments make the contention-based MAC protocol difficult to approximate the optimal performance [16, 17]. In contrast, contention-free MAC protocols are more efficient. Consequently, the TDMA protocol, as one of the contention-free MAC protocols, has attracted significant attention. In TDMA, all nodes use allocated time slots for communication. However, due to the time-space uncertainty of an underwater acoustic channel, a time slot needs to add a protection time to achieve collision avoidance [18, 19], which reduces the channel utilization and throughput of UWSNs greatly.

Furthermore, energy saving is an important consideration in protocol design for UWSNs. The energy consumption of an underwater node is tightly correlated with the working state of the node. If the nodes' working state is reasonably arranged, the energy consumption can be reduced, and the network lifetime can be extended. Particularly for some long-term monitoring applications, the data communication between nodes is not so frequency, which means that the transceivers of the sensor nodes are in idle state most of the time. It is more energy saving when an idle node goes in sleeping; thus, a sleeping scheme is one of the effective solutions to save energy [20–24].

In this paper, taking account into sleeping, we propose a MAC protocol of concurrent scheduling based on spatial-temporal uncertainty. The main contributions of this paper are as follows:

First, due to the spatial-temporal uncertainty in UWSNs, there are different propagating delays between the receiving node and different sending nodes. In our protocol, the intended receiving nodes schedule the receiving time reasonably, map the receiving time into the sending time according to the propagating-delay-difference, and advertise to its child node (i.e., the sending nodes) to achieve concurrency while avoiding collision. Further, the scheduling scheme based on receiving nodes and spatial-temporal uncertainty reduces the idle-interval between two successive packets at the receiver, thus increasing network throughput and improving channel utilization.

Second, we adopt a sleeping scheme in which each node can automatically wake up according to the scheduling scheme to receive or send data. The remainder of the time they can sleep to save energy, resulting in network energy saving.

Third, simulation results show that compared with other existing MAC protocols, the CSSTU-MAC protocol decreases energy consumption and improves the throughput of the network while avoiding collisions.

The remainder of this paper is organized as follows: In Section 2, we reviewed the related work on other MAC protocol. In Section 3, the problems of collision as well as spatial-temporal uncertainty are analyzed. Section 4 presents the CSSTU-MAC protocol. Section 5 evaluates the performance of the CSSTU-MAC protocol through NS3 simulation. Finally, Section 6 summarizes the paper.

2. Related Works

Collision and interference cause the receiving node unable to receive a packet successfully; thus, the sending node has to retransmission and consume more energy [25]. Therefore, many MAC mechanisms for collision resolution have been proposed successively. MAC protocols in UWSNs are generally divided into contention-based MAC and contention-free protocols [26, 27]. ALOHA is the earliest proposed contention-based MAC protocol. Without any collision avoidance mechanism, ALOHA based on random access performs well in a network with sporadic packets [28]. However, it is unsuitable for either moderate or heavy load networks. To improve the ALOHA protocol, a time-slot ALOHA protocol was proposed in [29]. With the slotted ALOHA protocol, nodes transmit packets at the beginning of a time-slot, which can avoid collisions to a certain extent. The MACA-MN protocol employs an RTS/CTS handshake mechanism to avoid the collision in a multihop network, which addresses the hidden terminal problem [30]. However, the long handshake time lengthens the average waiting time of the node for acquiring channel, thus increasing the end-to-end delay.

The slotted-FAMA protocol is another classic contention-based MAC protocol, which combines carrier listening mechanism with handshake mechanism to avoid collision [31]. Considering the long propagation delay of UWSNs, with the slotted-FAMA protocol, all data packets are sent at the beginning of time slot, and an ACK packet is feedbacked to the sender by the receiver after receiving successfully a data packet. However, with the slotted-FAMA protocol, packets are unable to transmit concurrently. The long propagating delay and the resulted long time-slot decrease the channel utilization of UWSNs. The DACAP protocol proposed in [32] also combines handshake with carrier listening. However, the DACAP protocol does not require timing synchronization between nodes. T-Lohi is a contention-based protocol where channel reservation uses short wake-up tones [33]. The T-Lohi protocol exploits latency difference to detect collisions. However, with the T-Lohi protocol, a node in idle state is subject to listen to the channel in each round, which consumes some energy.

Besides those contention-based MAC protocols, some contention-free MAC mechanisms for UWSNs were proposed in recent years. ST-MAC is a MAC protocol based on TDMA [34]. In the ST-MAC protocol, each node maintains a collision table according to the global network topology and then uses the collision table to schedule its transmission. However, it is difficult to get the realtime global topology for a dynamic multihop underwater network. The UWAN-MAC [35] achieves clock synchronization through exchanging sync packets between nodes. According to the received sync packets, each node knows the sleeping and waking timeline of neighbor nodes, thus can calculate its transmission period. However, in order to improve energy efficiency and reduce collision, the nodes with the UWAN-MAC use small duty cycles, which leads to low bandwidth utilization, so the UWAN-MAC protocol is suitable for long-term monitoring UWSNs with sporadic traffic. In [36], a space sharing TDMA protocol was proposed, which takes into account both the reliability and efficiency requirements of the network. The proposed protocol addressed the problems of “near-far” and varying topology by introducing the quality test method. Su et al. proposed an asynchronous wake-up scheme based on combinatorial designs to minimize the working duty cycle of sensor nodes in [37], which minimized the duty cycle of nodes to achieve energy saving. Alfouzan et al. proposed a reservation-based distributed ED-MAC protocol [38], in which each node uses local information to schedule itself independently, and a duty cycle mechanism is employed to address the hidden node problem.

To date, the MAC protocols mentioned above, either the contention-based or the contention-free protocols, employ time-division approach to occupy channel, in which different nodes transmit packets at different period of time.

UWSNs employ acoustic signals to communicate, and the acoustic waves propagate in water is approximately 1500 m/s, and the propagation delay in UWSNs is longer than that in WSNs with RF radios. So, in UWSNs with long propagating delay, in order to avoid collisions, the period of time in time-division transmission approach is subject to be long enough, which results in poor channel utilization and low throughput.

The propagating delay in UWSNs changes with the distance of transmission; thus, the arriving time of a packet at the receiving node is uncertain even if the sending time of the packet is known, which is the spatial-temporal uncertainty problem in UWSNs. The spatial-temporal uncertainty problem can be addressed by determining the sending time and the propagating distance between the sending and receiving node.

Considering the long and different propagation delay between the intended receiving node and multiple sending nodes, we come up with a new idea, that is, as long as there is no collision at the intended receiving node, multiple sending nodes can transmit packets simultaneously. So, in this paper, we propose a MAC protocol of concurrent scheduling for UWSNs based on spatial-temporal uncertainty called CSSTU-MAC. To perform concurrent transmission in the premise of avoiding collisions, an underwater node sends data packets at the time scheduled by its parent node (the

intended receiving node). The parent node designs in advance its nonoverlapping receiving time, then calculate and assign the sending time for each child node (sending node) according to their respective delay.

3. Problem Analysis

In this section, the collision problem and spatial-temporal uncertainty problem are described.

3.1. Collision Problem. In UWSNs, there are three main collision situations. The first is “receive-receive collision,” as shown in Figure 1(a). Two or more packets from different senders arrive a receiver simultaneously. In this case, none of the packets can be received successfully. The second type is “send-receive collision,” as shown in Figure 1(b). When a node is sending a packet, another packet from another node arrives at the node. In this case, the arrived packet is unable to be received successfully. This is due to that underwater acoustic modems usually work in half duplex mode. The third type is “packet crosstalk,” as shown in Figure 1(c), which generally means that the packet receiving is interfered by other packets sent to other nodes.

3.2. Spatial-Temporal Uncertainty Problem. The propagation delay of terrestrial wireless sensor network can be ignored because it is very small compared with the transmission delay. However, the UWSNs employ acoustic waves to transmit information, and the propagation speed of an acoustic wave in water is approximately 1500 m/s; thus, the propagation delay is relatively large and cannot be ignored. Moreover, the propagation delay in UWSNs changes with the distance of transmission; thus, the arriving time of a packet at the receiving node is uncertain even if the sending time of the packet is known. Consequently, multiple packets from different senders can interfere with each other at the same receiving node even if the packets are sent out at different times, which is due to the spatial-temporal uncertainty problem in UWSNs.

The spatial-temporal uncertainty problem is caused by two uncertain factors, the transmission distance and the sending time of the packet. Whether multiple packets collide at the receiving node or not depends on the propagation delays and the sending times of the packets [38].

Figure 2 shows how the long and variable propagation delays in UWSNs result in the spatial-temporal uncertainty. The topology is shown in Figure 2(a), in which nodes S_1 and S_2 are two sending nodes, and node r is the receiving node. Figures 2(b) and 2(c) show two cases of the spatial-temporal uncertainty. In Figure 2(b), nodes S_1 and S_2 send packet simultaneously; however, due to the different propagation delays resulted by different propagation distances, the arriving times of the two packets at the receiving node r are nonoverlapping, and no collision occurs at node r . In contrast, in Figure 2(c), nodes S_1 and S_2 send packets at different times; nevertheless, the two packets arrive at node r at the same time, resulting in collision.

Inspired by Figure 2(b), we can achieve collision avoidance as well as concurrent transmission through arranging

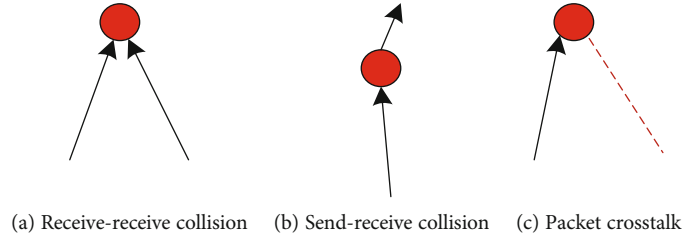


FIGURE 1: Collision types.

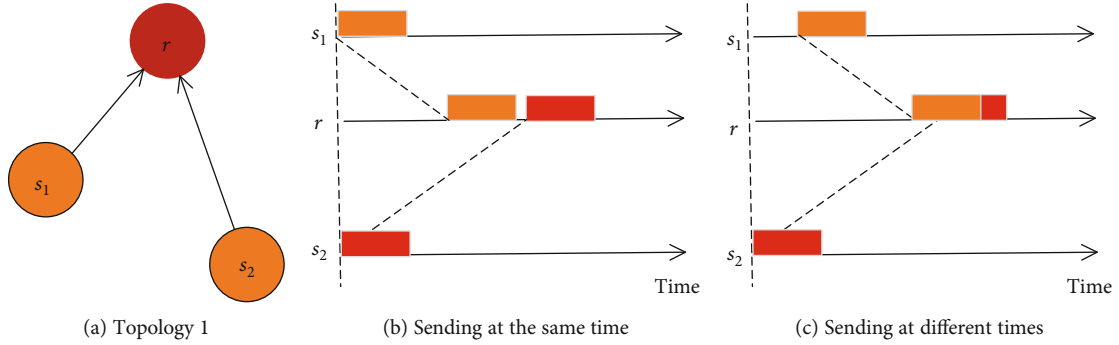


FIGURE 2: The spatial-temporal uncertainty problem.

in advance the nonoverlapping arriving times of packets from different senders and scheduling the sending times of multiple senders according to the variable propagation delay and the arriving times of packets.

4. CSSTU-MAC Protocol

In this section, we provide the relevant definitions and symbols, analyze the relevant models, present the basic ideas of CSSTU-MAC, and describe the design in detail. Then, we discuss how CSSTU-MAC addresses interference and collision; finally, a solution is provided to the problems of node failure and new node joining.

Table 1 shows the relevant symbols used in this paper.

The following terms and symbols are defined and will be used for the remainder of the paper:

Sink node. The node on the water's surface that is responsible for collecting the information from other nodes. It is always in the wake-up state and acts as the root node of the tree in the network topology

Parent node. The next hop node of a data packet. A data packet is always relayed to the sink node through a parent node in multihops UWSNs

Child node. The last hop node of a data packet. In each hop, a data packet is always transmitted from a child node to its parent node, and after multihops, the data packet finally arrives at the sink node

Leaf node. A sensor node that is responsible for collecting information and generating data packets. A leaf node has no child node

Sending node. The node that is sending or forwarding packets

Receiving node. The node that is receiving packets

4.1. Model Analysis. In this subsection, we show and analyze the related network, energy consumption, and delay models.

4.1.1. Network Model. In CSSTU-MAC, the sink node, which is equipped with both acoustic and radio modems, is placed on the surface. Underwater sensor nodes are equipped with acoustic modems and transmit only acoustic signals. All nodes are deployed in a 3D area as shown in Figure 3(a), and data transmission is directional from sensor nodes to the sink node. For example, in Figure 3(b), the network is comprised of a sink node and many underwater sensor nodes as shown (using four layers as an example). Each node connecting to the sink is configured with a layer according to the hop-count from the sink to the node. The layer of the sink is set to "0." There are a large number of source nodes in the network, and each source node always has a large number of packets to transmit. The data transmission is always along the direction from a node with a higher layer to a node with a lower layer, till to the sink.

4.1.2. Energy Consumption Model. UWSNs employ acoustic signals to communicate. The total energy consumption of nodes in UWSNs is shown in formula (1).

$$E_{\text{all}} = E_{\text{send}} + E_{\text{receive}} + E_{\text{idle}}, \quad (1)$$

$$E_{\text{send}} = lP_0A(d) \quad E_{\text{receive}} = \lambda_r l \quad E_{\text{idle}} = T_{\text{idle}}E_{\text{aver-idle}}, \quad (2)$$

TABLE 1: Symbols.

Terms	Explanation
SID	The ID of sending node
sid _{<i>i</i>}	The ID of child node <i>i</i>
N_i^L	The number of all child nodes of node <i>i</i> with layer “ <i>L</i> ”
v_p	The speed of sound in water
R_{bit}	The transmission rate of channel
<i>l</i>	Size of a packet
l_{max}	Maximum packet length in the network
t_{si}	Time the sender starts to send packets
t_{ri}	Time the receiver to finish receiving packets
L_i	Layer of node <i>i</i>
N_i	Neighbor table of the node
<i>T</i>	Duration of one listen/sleep cycle
T_{awake}	Duration of node waking (including idle listening and working times)
T_{work}	Duration of node working (including sending or receiving times)
T_{idle}	Duration of node idle listening
T_{sleep}	Duration of node sleeping
T_{ACK}	Duration of a ACK frame
Pos	Location of the node
D_i	Distance between the sender and receiving nodes
D_{max}	Maximum transmission distance in network
ΔD	Maximum distance offset for node movement

where E_{send} is the transmission energy consumption, E_{receive} is the receiving energy consumption, E_{idle} is the energy consumption for idle waiting, *l* is the length of a packet, P_0 is the receiving power of receiving packets, λ_r is the receiving energy consumption coefficient, $E_{\text{aver-idle}}$ is the average energy consumption of nodes in idle waiting state, and $A(d)$ represents the energy attenuation when the propagation distance of the acoustic is *d* in the water, as shown in formula (3).

$$A(d) = d^\eta a^d, \quad (3)$$

$$a = 10^{\alpha(f)/10}, \quad (4)$$

$$\alpha(f) = 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f^2} + 2.75 * 10^{-4} f^2 + 3 * 10^{-3}, \quad (5)$$

where η is the energy diffusion factor (the value of η is generally 1.5), $\alpha(f)$ is the absorption coefficient, and *f* is the carrier frequency of the acoustic modem. From the above equation, we can find that the total energy consumption can be decreased by reducing the idle time of nodes assuming that other parameters remain unchanged.

4.1.3. Delay Model. Underwater sensor nodes utilize acoustic waves to transmit information, and the propagation speed of an acoustic wave in water is only approximately 1500 m/s, which is several orders of magnitude lower than that of radio signals. Therefore, the propagation delay of an underwater signal is relatively large. The main compositions of network delay are as follows: processing delay T_{proc} , queue delay T_{queue} , propagation delay T_{prop} , and transmission delay T_{trans} . Therefore, the network delay in a communication can be calculated by the following:

$$T_{\text{all}} = T_{\text{proc}} + T_{\text{prop}} + T_{\text{trans}} + T_{\text{queue}}, \quad (6)$$

where $T_{\text{prop}} = \sum_{i=1}^N D_i / v_p$ and $T_{\text{trans}} = N / (l R_{\text{bit}})$ (*N* is the hop-count).

The processing delay and queue delay are far less than the propagation delay in UWSNs; we mainly consider the propagation delay and transmission delay.

In CSSTU-MAC, the operation of each node alternates between the wake-up mode and sleep mode periodically, and nodes have two states when they wake up: working and idle. Given one cycle of sleeping and waking, denoted as *T*, is a fixed value.

$$T = T_{\text{awake}} + T_{\text{sleep}}, \quad (7)$$

$$T_{\text{awake}} = T_{\text{work}} + T_{\text{idle}}. \quad (8)$$

Since *T* is fixed, in formula (7), it is assumed that when T_{work} is constant, the sleep time can be prolonged by reducing the idle time of nodes, thus reducing energy consumption.

As a consequence, reducing the idle time of nodes reasonably plays an important role in improving network performance, especially in terms of network throughput and energy saving.

4.2. Overview of the CSSTU-MAC Protocol. The CSSTU-MAC protocol is based on a scheduling mechanism. First, it adopts a sleeping scheme to reduce the energy consumption caused by the idle times of nodes. When a node has no packet to transmit (receive), it is said to be in idle state. However, the energy consumption by a node in idle state is not negligible. If an idle node is switched to sleeping state, its transducer is closed, so the energy consumption is negligible. Second, the CSSTU-MAC utilizes the spatial-temporal uncertainty of UWSNs to allow the receiver to allocate the senders' sending times. This allows for distributed scheduling. By reasonably scheduling the working time of the nodes, the collision caused by the space-time uncertainty of the underwater acoustic channel is overcome, and concurrent transmission can be implemented, and the idle waiting time of the nodes is reduced. Therefore, the CSSTU-MAC protocol improves channel utilization, network throughput, and energy efficiency while avoiding collisions.

4.3. Initial Phase. In initialization phase, the sink node starts to broadcast hello packets periodically in which the ID, position, layer, and child-node-number of sending node are

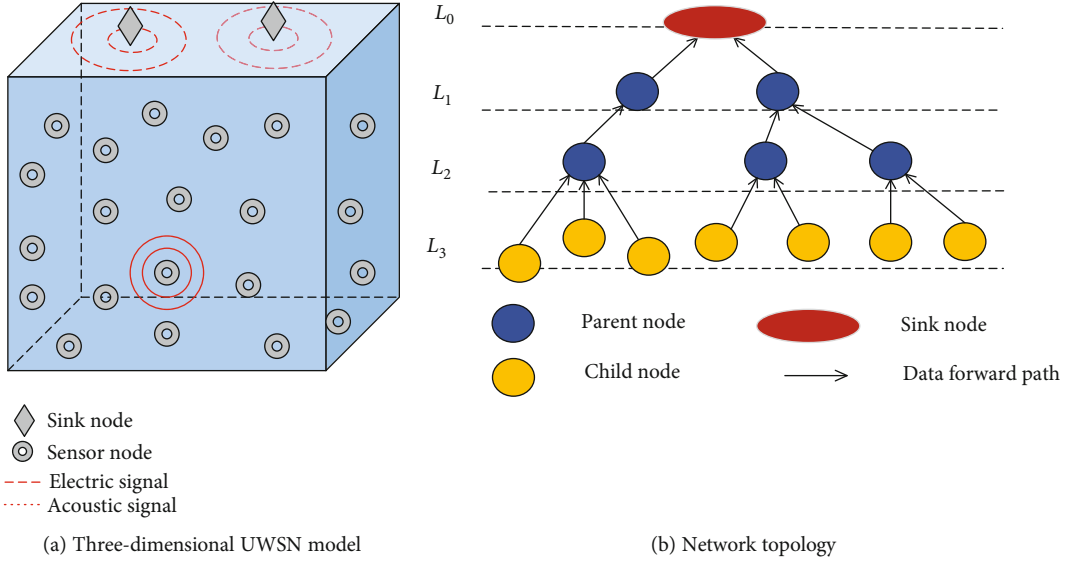


FIGURE 3: Network model.

included to advertise the information of the sender. The format of a hello packet is shown in Table 2. When node i receives a hello message, it extracts the layer, then updates layer and child-node-number of the hello message with its own and continues to broadcast the updated hello message. A hello packet is flooded along the directions from the nodes with lower layer to the nodes with higher layer. Through hello packets, an underwater node obtains its and neighbors' information of ID, layer [8], position [39], and child-node-number. Then, the node updates its neighbor table N_t and then calculates the distances between it and its neighbor nodes according to their positions. The cycle period T is also determined in the initialization phase according to the specific application of the network.

4.4. Allocation Process of the Sending Time. The primary objective of MAC protocols for UWSNs is to avoid multiple packets from arriving the receiving node at the same time. When a packet arrives at, its receiver is not only related to the sending time of the packet but also related to the distance between the receiving and sending nodes, as well as the size of the packet. After the initialization phase is completed, each node obtains a neighbor table N_t , recording the layer and other information of neighbor node, according to which the node can establish a local topology as in Figure 4.

A data packet is always delivered along a direction from a node with a higher layer to a node with a lower layer, till to the sink node with layer "0." So, an underwater node relays the data packets from its child node to its parent node. So, in order to avoid collision, in our scheduling scheme, the sending times of child nodes are allocated by their parent node. As in Figure 4, the sending times of nodes S_3, S_4, S_5 , and S_6 are preallocated by node S_2 . The parent node first determines the sending order of its child nodes according to the priority $P(s_i)$ of the child nodes, the smaller the

TABLE 2: The format of hello packet.

Bit	8	4	8	8
	ID of the sender	Layer	Position	Child-node-number

value of $P(s_i)$, the higher the priority, which is shown in equation (9).

$$P(s_i) = \begin{cases} \frac{D_i}{D_{\max}} & \text{if } D_1 \neq D_2 \neq \dots \neq D_n \\ \frac{D_i}{D_{\max}} + N_i^L \times l_{\max} & \text{else} \end{cases} \quad (9)$$

As seen in equation (9), given the number of child nodes n , the child node closer to the parent node has the highest priority. If the distance is equal, the smaller the value of N_i^L , the higher the priority. So firstly, the sink node with layer "0" presets the time to start receiving packets as $t_{r,1}$, and it calculates the sending time $t_{s,1}$ when the first child node starts to send packets. According to the allocation algorithm (Algorithm 1), the sending time of each child node can be calculated individually. Then, the sink node broadcasts a CONFIG packet as shown in Table 3 that carries the sid and the sending time of each child node. When each child node receives the CONFIG packet from the sink node, it finds the sending time of its own according to the correspondingsid _{i} and $t_{s,i}$ in the CONFIG packet. After determining the sending time, the nodes with level "1" start to act as a parent node, calculate the receiving times of its self and the sending times of its child nodes without "send-receive conflict", and broadcast CONFIG packets to their child nodes, and so on; all the parent nodes complete the calculation and scheduling.

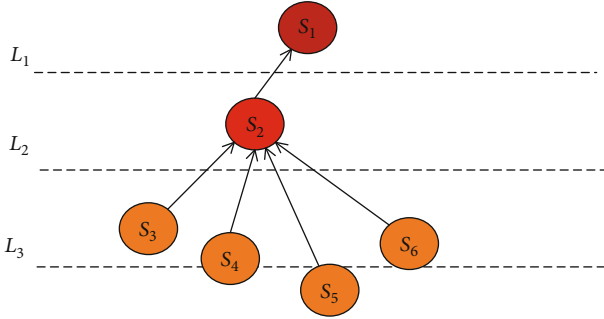


FIGURE 4: Local topology.

4.4.1. Calculation Process. In this subsection, the calculation process for the transmission time will be presented, including some assumptions, definitions, and necessary conditions required in the calculation.

(1) *Assumptions and Definitions.* We assume that the moving model of nodes follows the moving model in reference [39], and the maximum deviation of the node's moving distance is ΔD .

Definition 1. $\Delta T = t_{r(i+1)} - t_{r_i}$, where ΔT is the interval between two successive receiving times. When $\Delta T \geq 0$, there is no collision at the receiver. The smaller ΔT is, the smaller the idle waiting delay of the receiving node. (2) *Objective.* To reduce the idle waiting delay of the sender and the receiver, the concurrency of the transmission is considered. The prerequisite for concurrent transmission is $\Delta T \geq 0$. And if the interval ΔT is infinitesimal, not only the end-to-end delay can be reduced but also the sleeping time of the node can be prolonged (T is a constant), which can further achieve energy saving.

(3) *Calculation and Derivation.* Since a parent node has N_i^L child nodes. Given the amount of data required to be transmitted at node i is denoted by l_i , is given by formula (10).

$$l_i = l + (l \times N_i^L), \quad (10)$$

where $N_i^L = N_1^{L+1} + N_2^{L+1} + \dots + N_k^{L+1} + k$, the k is the number of child nodes with layer "L+1" of node i with layer "L." If we take the maximum value of $N_{i-\max}^L$ in equation (9), we can get that the maximum amount of data required to be transmitted at node i is $l_{i-\max}$, which is given by equation (11).

$$l_{i-\max} = l + (l \times N_{i-\max}^L). \quad (11)$$

The sink node first presets that the time to start receiving packets from the first child node is t_{s1} . Then, the send-

ing time of the first child node is t_{s1} , which is given by formula (12).

$$t_{s1} = \left[T + \left(t_{r1} - \frac{D_1 - \Delta D}{v_p} - \frac{l_{i-\max}}{R_{\text{bit}}} \right) \right] \bmod T. \quad (12)$$

Note: negative results are normalized to positive values, where T is the duration of one cycle.

Meanwhile, the time to finish receiving packets t'_{r1} is given by formula (13).

$$t'_{r1} = t_{s1} + \frac{D_1 + \Delta D}{v_p} + \frac{l_{i-\max}}{R_{\text{bit}}}. \quad (13)$$

Then, when $\Delta T \geq 0$ is satisfied, the next child node will be selected. If a child node that can transmit concurrently with the previous child node is found in the remaining child nodes, the sending time of the current child node is the same as that of the previous child node, and the receiving time of the parent node is $t_{\text{pr}(i+1)}$, which is given by formula (14).

$$t_{\text{pr}(i+1)} = t_{s1} + \frac{(D_i - D_1) + (2i - 1)\Delta D}{v_p} + \frac{l_{i-\max}}{R_{\text{bit}}}. \quad (14)$$

When $t_{\text{ps}(i+1)} = t_{s_i}$ and $t_{\text{pr}(i+1)} \geq t_{r_i}$ are satisfied, that is, when $D_{i+1} \geq D_i + 2\Delta D + (l_{i-\max}/R_{\text{bit}}) \times v_p$, concurrent transmission can be realized. Here, we first determine the node whose distance is $D_{i+1} = \min(D_i + 2\Delta D + (l_{i-\max}/R_{\text{bit}}) \times v_p)$, which is the first concurrent transmission node.

If there is no child node that can transmit concurrently with the previous child node. Then, we obtain the receiving time of a node as t_{r_i} , given by formula (15).

$$t_{r_i} = \left[T + \left(t_{s_i} + \frac{D_i + \Delta D}{v_p} + \frac{l_{i-\max}}{R_{\text{bit}}} \right) \right] \bmod T. \quad (15)$$

Furthermore, we obtain the receiving time of a node as t_{r_i} , given by formula. Furthermore, the sending time of this child node is t_{s_i} , as calculated in formula (16).

$$t_{s_i} = t_{s1} + \frac{(D_i - D_1) + 2(i - 1)\Delta D}{v_p} + \frac{l_{i-\max}}{R_{\text{bit}}} + T_{\text{ACK}}. \quad (16)$$

(4) *Allocation Process.* Here, we introduce several symbols.

$Z(r)$: the set of all child nodes of a given node

$O(s)$: the set of child nodes that have been allocated sending times

$R(s)$: the set of child nodes that have not yet been allocated sending times

We can get the equation given by formula (17).

$$R(s) = Z(r) - O(s). \quad (17)$$

Step 1. At the beginning of the algorithm, all the subnodes of

```

Initialization:
1:  $O(s)=null$ ;
2:  $P(r)=\{sink\}$ ; // the first parent node is sink node
Begin
3:  $O(s)=P(r)$ ;
4: While( $x \in O(s)$  and  $x.attribute=true$ ){
5: calculate  $Z(r)=\{r|r \text{ is sub node of } x \text{ node}\}$ ;
6:  $R(s)=Z(r)-O(s)$ ;
7:  $pNode=sNode=x$ ;
8:  $pNode.t_{r1}=0$ ,  $flag=true$ ;
9: while(  $R(s) \neq null$  and  $s.attribute=true$ ){ // Node  $s$  is not leaf node
10: select  $j \in Z(r)$ , calculate  $P(s_j) = P(s_j) \cdot pNode$ ;
11: for each  $i \in Z(r)-\{j\}$ 
12: if ( $P(s_i) \cdot pNode < P(s_j)$ ) then  $P(s_j) = P(s_i)$ ,  $j=i$ ;
13: if ( $flag=true$ ) calculate  $j.t_{s1} | pNode.t_{r1}=0$ ;
14: else calculate  $j.t_{s(i+1)} | sNode.t_{r1}$ ;
15:  $O(s) = O(s) \cup \{j\}$ ,  $R(s)=Z(r)-O(s)$ ;
16: calculate  $j.t_{r1}$ ;
17: for each  $i \in R(s)$  { //look for concurrent sub nodes
18: if ( $t_{psi}=t_{sj}$ ) and ( $t'_{ri}=t_{ri}$ )
19: {  $O(s) = O(s) \cup \{i\}$ ;
20:  $R(s)=Z(r)-O(s)$ ;
21: calculate  $i.t_{pri}$ ;
22:}
23:}
24:  $sNode=i$ ;
25:  $flag=!flag$ ;
26:}
27:} End

```

ALGORITHM 1: Allocation process of the sending time.

TABLE 3: Format of the CONFIG packet.

Bit	8	8	4	8	4	8	4
	SID	sid ₁	t _{s1}	sid ₂	t _{s2}	sid _i	t _{si}

the current parent node (sink node) are added to $Z(r)$ and $R(s)$, and the earliest receiving time of the parent node is set to $t_{r1}=0$. Then, the subnode with the highest priority is selected as the first sending node from $Z(r)$. Calculate the sending time of this subnode and add this node to $O(s)$. According to formula (17), the set $R(s)$ is updated.

Step 2. The parent node finds a subnode in the current set $R(s)$ that can transmit concurrently with the previous subnode. If such a node is found, add it to $O(s)$ and update the set $R(s)$ according to formula (17). Next, calculate the receiving time when the parent node receives the packets of this subnode as t_{pr} ; then, return (2). If no such node is found, return to Step 3.

Step 3. The parent node selects the subnode with the highest priority from the current set $R(s)$ and makes this subnode the next sending node. According to the receiving time of the previous node, the earliest sending time of this subnode is

calculated as t_s . This node is added to $O(s)$, and set $R(s)$ is updated. Determine if $R(s) = null$. If no, return to Step 2.

Step 4. If $R(s) = null$, the algorithm ends.

The flow diagram of allocation process of the sending time is given in Figure 5. With the proposed CSSTU-MAC protocol, each parent node firstly calculates the earliest sending time of its subnodes. In the worst case, a parent node needs to calculate $M-1$ times, M is the number of its child nodes. Therefore, according to the definition of time complexity of an algorithm, the time complexity of the proposed algorithm is $O(M)$.

4.4.2. An Allocation Instance. Here, we use a layered topology as an example to illustrate the process of allocating transmission times. As shown in Figure 4, node S_2 is a receiving node of data packets, that is, the parent node, and nodes S_3, S_4, S_5 , and S_6 are its child nodes. In the initial stage, let $Z(r) = \{S_3, S_4, S_5, S_6\}$, $R(s) = \{S_3, S_4, S_5, S_6\}$, and $O(s) = null$. First, node S_2 presets its starting receiving time to $t_{r1}=0$, and the child node with the highest priority in set $Z(r)$ is node S_3 . So, node S_3 is determined to be the first sending node, and its earliest sending time is t_s . Then, node S_2 calculates the sending time, t_s , of node S_3 , according to the receiving time and the propagating delay between node S_2 and node S_3 ; then, node S_3 is added to the set $O(s)$, $O(s) = \{S_3\}$, and $R(s) = \{S_4, S_5, S_6\}$. Node S_2 selects a child node in the current set $R(s)$ that can

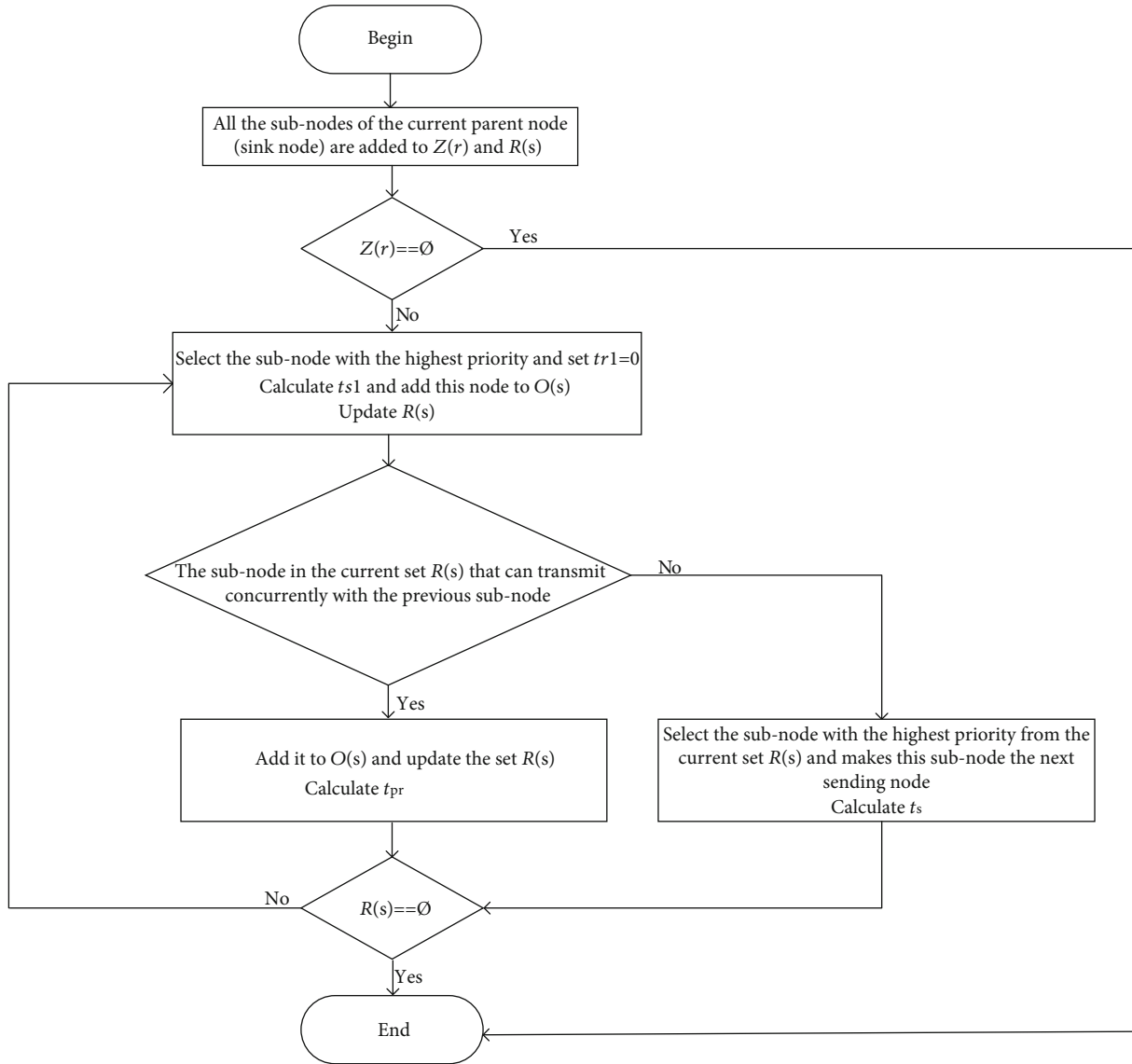


FIGURE 5: Allocation process flow chart.

be concurrent with node S_3 according to different propagating delays. If node S_6 is selected, then $O(s) = \{S_3, S_6\}$ and $R(s) = \{S_4, S_5\}$. t_{pr} is the time node S_2 finishes receiving packets of child node S_6 . Next, node S_2 selects a child node in the current set $R(s)$ that can be concurrent with node S_6 . If node S_5 is selected, then $O(s) = \{S_3, S_6, S_5\}$ and $R(s) = \{S_4\}$. At this moment, all the nodes that meet the concurrency conditions have been selected. Then, node S_2 will select the next sending node from the child nodes that do not meet the concurrency conditions. At this time, node S_4 is selected. Thus, $O(s) = \{S_3, S_6, S_5, S_4\}$ and $R(s) = \text{null}$.

According to the assigned transmission time, each child node starts the data transmission process, which is shown in Figure 6; each node maintains an idle listening time duration of T_{idle} , which is used to listen to the hello packets. In Figure 6, the sending nodes S_3 , S_5 , and S_6 are assigned the same transmission time by their parent node S_2 , and node S_4 is assigned a later transmission time. After finishing the

receiving of data packets, the parent node S_2 sends an acknowledgement (ACK) packet to inform the four nodes that it has received successfully the packets. After receiving an ACK the nodes S_3 , S_5 , S_6 , and S_4 go into sleeping mode to save energy, and the parent node S_2 starts to forward the received packets.

4.5. Handling of Node Interference. Schedule to avoid interference between two nodes in the same layer. When the nodes in the same layer interfere with each other, their transmission time is coordinated through their upper layer nodes (parent nodes). As shown in Figure 7(a), nodes 2 and 3 are in the same layer and interfere with each other. For example, the receiving of node 2 is interfered by the packets from the node 3 since node 3 is the neighbor of node 2. In this case, node 1 (the parent node) coordinates their respective working times.

Schedule to avoid interference between two nodes in different layer. As shown in Figure 7(b), the packets from node

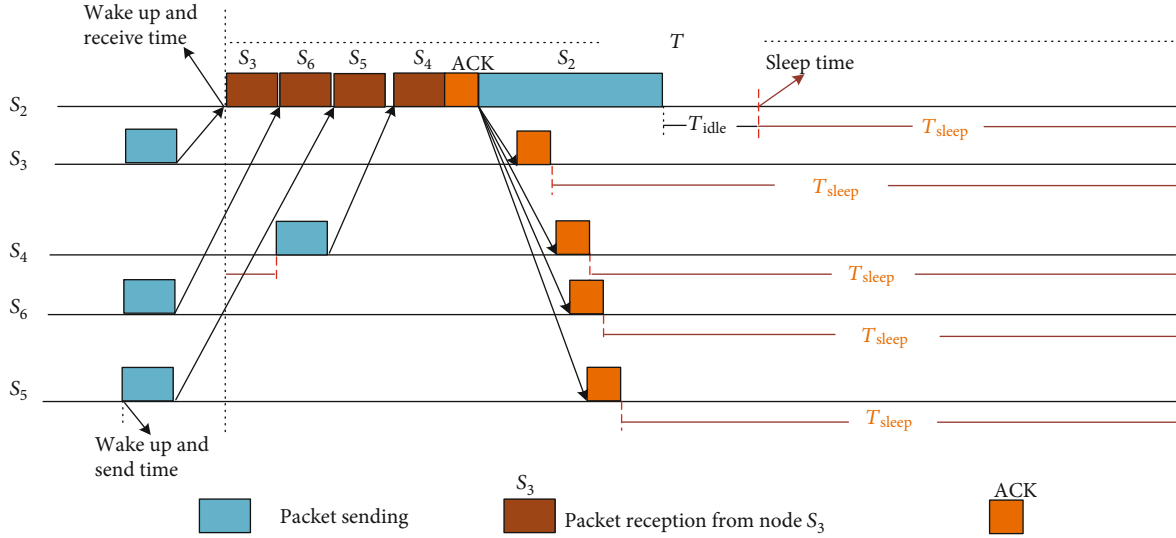


FIGURE 6: An instance of allocation and data transmission process.

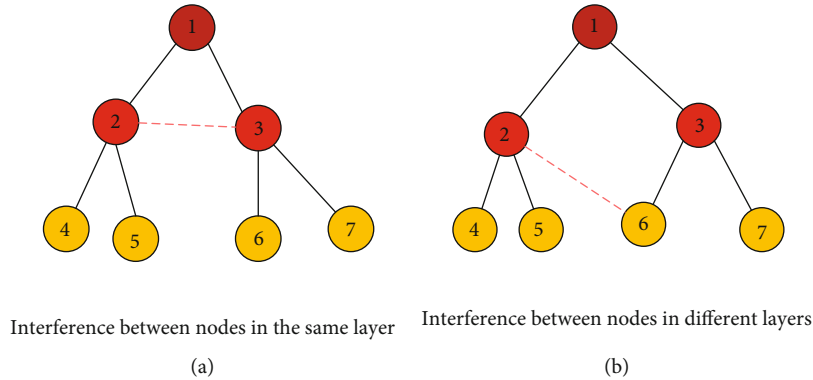


FIGURE 7: Interference between nodes.

6 (its parent node is node 3) can interfere with the reception of node 2 since node 6 is a neighbor of node 2. In this case, their common parent (i.e., node 1) of node 2 and node 3 coordinates the working time of child node 6 of node 3 (as long as the working time of node 6 is outside the working time of node 2).

4.6. Network Maintenance. In UWSNs, the leaving or joining of a node leads to dynamic change of networks topology. When a node does not receive any packet (either hello packets or data packets) from a child node within three hello intervals, it supposes that the child node has left. Through listening to the hello packets, a node can detect the leaving and joining of its child nodes, when a node detects the leaving or joining of its child nodes, it reallocates the sending time for its child nodes and broadcasts the new CONFIG packet. Here, we use Figure 4 as an example to illustrate the process of network maintenance. As shown in Figure 4, node S_2 is a receiving node, that is, the parent node, and nodes S_3, S_4, S_5 , and S_6 are its child nodes. We assume that node S_3 has left. When the node S_2 does not receive any packet (either hello packets or data packets) from node S_3 within three hello intervals, it reallocates the sending time for its child nodes

S_4, S_5 , and S_6 , then broadcasts the new CONFIG packet, that carries the sid and the sending time of its child nodes S_4, S_5 , and S_6 .

5. Performance Evaluation

In this section, the performance of CSSTU-MAC is evaluated and analyzed by comparing with ED-MAC [38], T-Lohi [33], and UWAN-MAC [35]. Firstly, we describe and qualitatively analyze these protocols; then, we provide the simulation environment and parameters; finally, we analyze the simulation results.

5.1. Qualitative Comparison. The performance comparison of the four MAC protocols is shown in Table 4. Among them, the main difference is that CSSTU-MAC allocates the sending time based on the nonoverlapping receiving time, not on competition; thus, the CSSTU-MAC protocol achieves concurrent transmission while avoiding collision. ED-MAC is a reservation protocol based on TDMA. It schedules the slots according to the priority of nodes which based on their depth in the network. Each slot includes a number of sub-slots. The subslots are selected randomly to deal with the

TABLE 4: Qualitative comparison of the four MAC protocols.

Protocol	Throughput	End-to-end delay	Energy consumption	Dynamic	Collision rate
UWAN-MAC	Low	High	Lower	No	Lower
T-Lohi	Lower	High	High	No	High
ED-MAC	Higher	Higher	High	No	Higher
CSSTU-MAC	High	Lower	Low	Yes	Low

possibility of collisions between two nodes if they have selected the same slot. T-Lohi is a typical competitive protocol, in which the nodes use tone at the beginning of each round to retain the channel, and it uses long packet to awaken nodes instead of automatic waking-up. In addition, T-Lohi requires a node to be awake and listen to the channel in each round to compete for the channel. The listening period lasts at least the maximum single-trip propagation time plus the time to detect the contention tone, which results in a low channel utilization [40]. UWAN-MAC has random behavior in the data transmission stage, in which the sender node selects random time to send its packets, which may result in collisions. To improve energy efficiency and reduce the collision, each node with the UWAN-MAC protocol is set to a small duty cycle, which leads to a low throughput. Essentially, both the T-Lohi and the UWAN-MAC protocol employ the time-division transmission approach. Due to the long propagation delay in UWSNs, the interval between two successive data packets in a shared channel is long, and the propagation delay of the control frames to contend channel is long too, which results in poor channel utilization and low throughput in both the T-Lohi and the UWAN-MAC protocol. In contrast, with concurrent transmission, CSSTU-MAC protocol improves greatly the network throughput while avoiding collisions.

5.2. Simulation Settings. We implement CSSTU-MAC simulation with NS3 platform, which includes underwater channel models. We use the following parameters in our simulation. The initial energy of all nodes is set to 1000 J. The packet size is set to 200 Bytes. We set every simulation runs for 3000 s, and every data point is the average of 15 simulations. Nodes are deployed in the area of 5000 m*5000 m*5000 m. The propagation speed of the acoustic signal is 1500 m/s. The transmission range of every node is about 1500 m. We assume that the location of the node will be affected by the wave motion; set node location range ΔD is 5 m. We observe the performance of each protocol in terms of throughput, end-to-end delay, and energy consumption. In the simulation, we use the SPRVA [41] for routing protocol. The simulation parameters are given in Table 5.

We used the following three metrics to compare the performance of CSSTU-MAC with ED-MAC, T-Lohi, and UWAN-MAC protocols:

Throughput is defined as the total amount of data successfully transmitted by the network within a given period of time.

End-to-end delay is defined as the latency of a packet from a source node to the sink node.

TABLE 5: Simulation parameters.

Parameter name	Value
Packet error rate model	Ns3:: UanPhyPerNoCode
Signal noise model	Ns3:: UanPhyCalcSinrDefault
UAN propagation model	Ns3:: UanPropModel thorp
Acoustic propagation speed	1500 m/s
Number of nodes	10~70
Center frequency	22 kHz
Bandwidth	10 kHz
Mode type	FSK
Date length	200 byte
TX power	2 W
RX power	0.75 W
Idle power	8 mW

Energy consumption is defined as the overall energy consumption in the network divided by the successfully delivered packets, which is measured in joules per packet.

5.3. Simulation Results. Firstly, when the network traffic is constant (with 0.25 packets per second), we compare the throughput of these protocols by increasing the number of nodes. Figure 8(a) shows the throughput comparison between the CSSTU-MAC and other three protocols. It can be seen that with the number of nodes increases, the throughput of all the four protocols increases first and then decreases, and the CSSTU-MAC achieves the highest network throughput among the four MAC protocols. When the number of nodes is small, the number of source nodes is small too, and the traffic load is light, resulting in a low throughput. With the number of source nodes increases, the channel tends to be saturated, resulting in a saturated network throughput. It is also seen that the protocol of CSSTU-MAC outperforms the other protocol in term of throughput. This is because the CSSTU-MAC protocol realizes concurrent transmission and reduces idle waiting. Especially, the CSSTU-MAC protocol utilizes the underwater channel efficiently without propagating those control packets to preempt the channel and can maximize the network throughput. Therefore, its throughput increases rapidly with the increasing of the number of nodes before reaching the saturation value. After the saturation value, the throughput decreases slowly. However, both UWAN-MAC and T-Lohi protocols do not consider the spatio-temporal uncertainty problem; their network throughput is lower than that of the CSSTU-MAC protocol. T-Lohi solves the collision of data transmission through reservation mechanism, but with the increasing

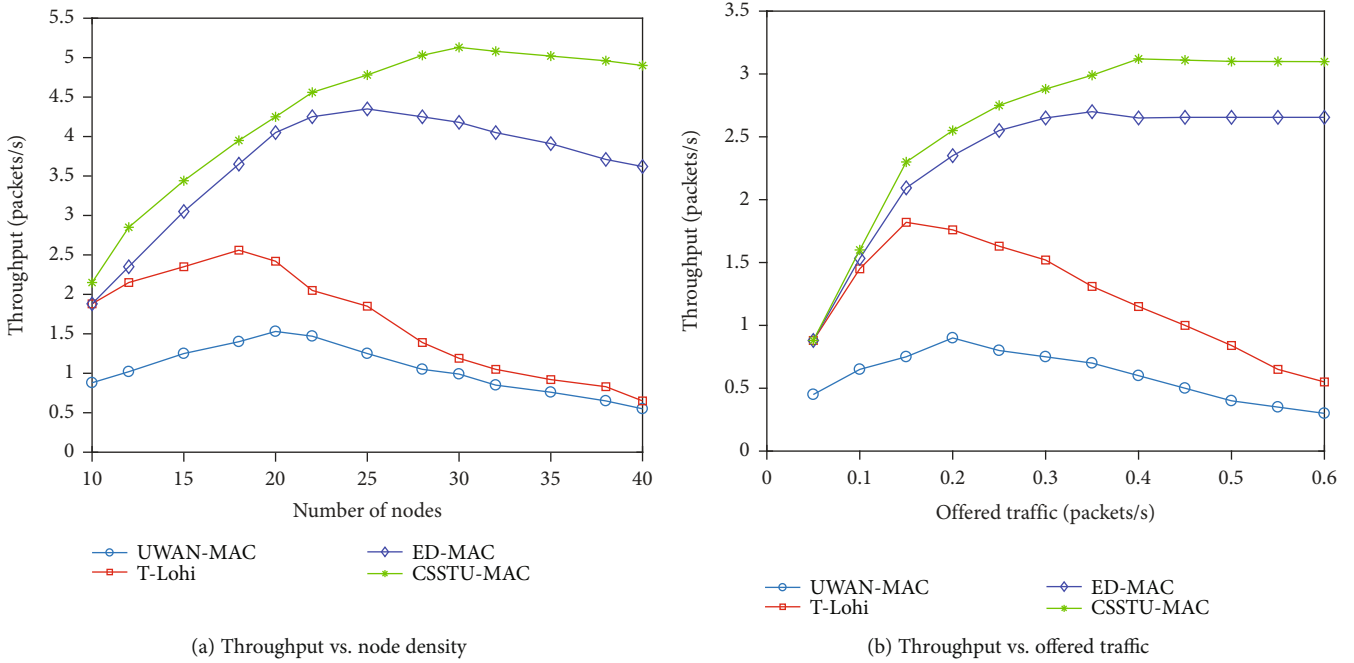


FIGURE 8: Network throughput under different parameters.

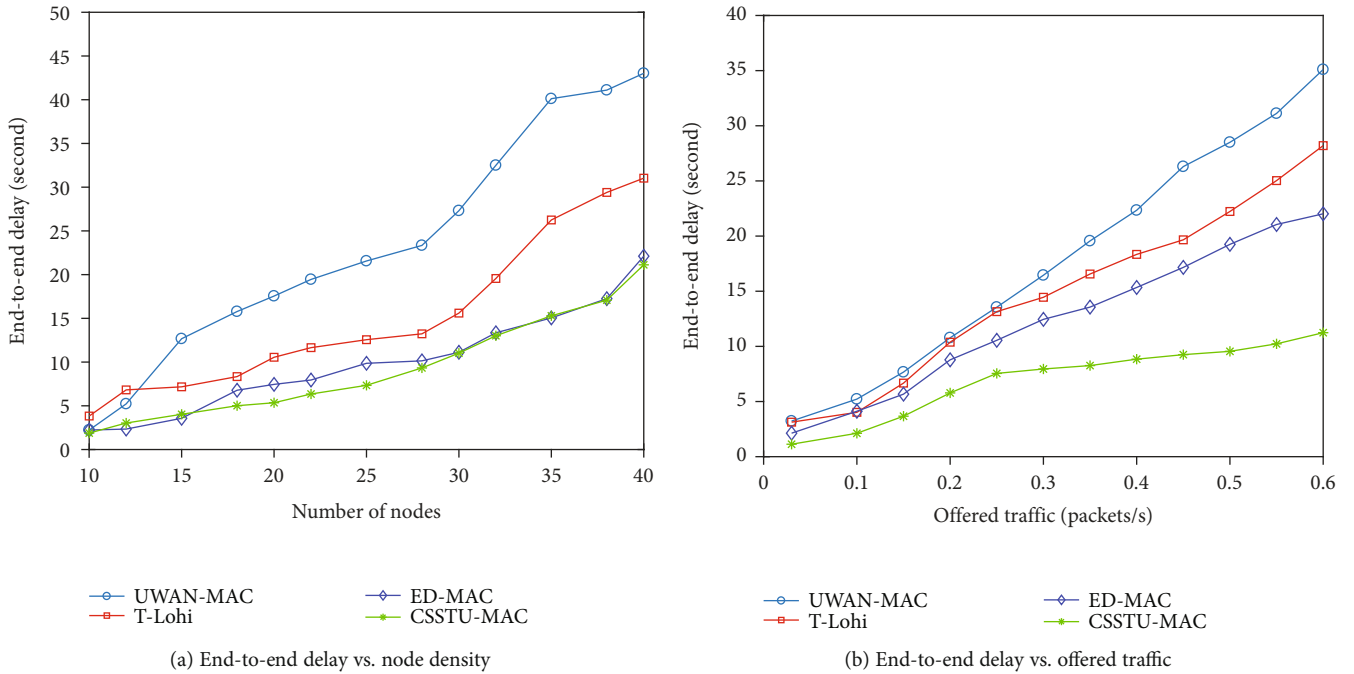


FIGURE 9: End-to-end delay under different parameters.

of the number of nodes, reservation delay increases. ED-MAC, because it selects random subslots when sending data packet, although its performance is higher than UWAN-MAC, there still be collisions with the increasing of node density. The throughput of CSSTU-MAC is improved by

51.8% compared with UWAN-MAC and 26.3% compared with T-Lohi. Therefore, when the network scale is large, CSSTU-MAC protocol shows better performance.

Figure 8(b) shows the effect of the network traffic (the number of packets generated per second) on throughput

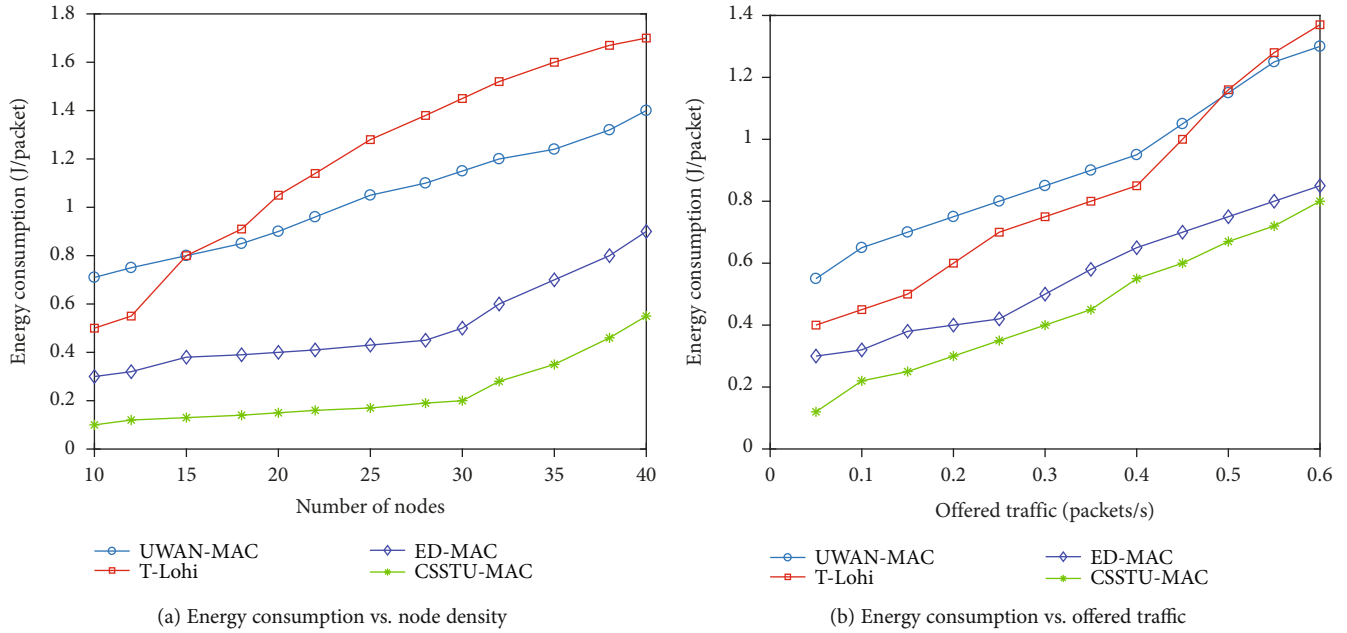


FIGURE 10: Energy consumption under different parameter.

when 40 underwater nodes are deployed in the network. It is observed that the network throughputs with the four protocols increase to their respective maximum values with the increasing of the network traffic and then decrease gently. When the network traffic is the same, the throughput with the CSSTU-MAC protocol is the highest as the CSSTU-MAC can transmit packets concurrently by addressing the spatial-temporal uncertainty problem. However, with ED-MAC, when the offered traffic is increased, some data packets collide at the receiver, resulting in packet loss. Again, either with UWAN-MAC or T-Lohi protocol, the collisions caused by the spatial-temporal uncertainty problem is ineluctable. As a result, their network throughput is lower than that of CSSTU-MAC.

Figures 9(a) and 9(b) show the respective effects of node density and traffic-load on end-to-end delay. From Figure 9, it is seen that the end-to-end delay increases with the increasing of node density as well as traffic-load. As the number of nodes increases, the number of packets generated increases. Considering the low bit rate, the average time of waiting to be transmitted is increased, which resulting in the increased end-to-end delay. The same is true for network traffic. In addition, the CSSTU-MAC employs a time allocation scheme based on concurrent scheduling, which reduces the idle time of the receiving node and the end-to-end delay. Consequently, compared with the other three protocols, the end-to-end latency with the CSSTU-MAC protocol is the lowest which can be observed in Figure 9.

Figures 10(a) and 10(b) show the respective effects of node density and traffic-load on energy consumption. From Figure 10, it is seen that the energy consumption increases with the increasing of node density as well as traffic load. It is observed that the CSSTU-MAC protocol is the most

energy saving though the four protocols all adopt the sleeping mechanism. This is because the CSSTU-MAC protocol uses the channel more efficiently, reduces the idle times, and saves the energy consuming for idle waiting. Furtherly, the CSSTU-MAC protocol avoids collisions by scheduling the sending times according to the different propagating delay, which reduces the energy consumption resulted by collision and retransmission. It is observed that, with the increasing of the traffic-load as well as the node density, the energy consumption of UWAN-MAC increases rapidly. The energy consumption of CSSTU-MAC decreases 48.2% compared with UWAN-MAC. As shown in Figure 10, the UWAN-MAC consumes the most energy among the four protocols.

In order to evaluate fully the performance of the MAC protocols, in the simulation, we change the number of nodes. As the number of nodes increases, the network throughput increases correspondingly and eventually reaches a saturation point. The average end-to-end delay of CSSTU-MAC protocol varies and increases with the increase of node density. As the number of nodes increases, the energy consumption increases correspondingly. Simulation results show that the proposed protocol is suitable for networks with nodes between 30 and 40.

6. Conclusions

In this paper, we propose an energy efficient and concurrent transmission MAC protocol based on the spatial-temporal uncertainty, which is applicable to the UWSNs with large data amount and multiple data sources. CSSTU-MAC mainly utilizes the long propagation delay of UWSNs to achieve concurrency and uses a scheduling scheme to allocate nodes reasonably to reduce energy consumption. In addition,

the CSSTU-MAC protocol reduces the idle waiting delay between nodes in data transmission by using the node concurrent transmission which improves throughput and channel utilization. A large number of simulation experiments show the outstanding performance of CSSTU-MAC compared with two other protocols in terms of energy consumption, throughput, and end-to-end delay. In future work, we will consider network reliability and network performance further in different application scenarios.

Data Availability

The simulation data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Xin Liu, Meiju Li, and Chong Li conceived and designed the experiments. Xin Liu and Lijuan Wang analyzed the data. Xin Liu and Xiujuan Du wrote the paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61962052 and in part by the IoT Innovation Team Foundation of Qinghai Office of Science and Technology under Grant No. 2020-ZJ-903.

References

- [1] J. Heidemann, L. Yuan, and A. Syed, "Research challenges and applications for underwater sensor networking," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006*, vol. 4no. 3, pp. 228–235, Las Vegas, NV, USA, 2006.
- [2] M. S. M. Alamgir, M. N. Sultana, and K. Chang, "Link adaptation on an underwater communications network using machine learning algorithms: boosted regression tree approach," *IEEE Access*, vol. 8, no. 1, pp. 73957–73971, 2020.
- [3] Z. Jin, Q. Zhao, and Y. Su, "RCAR: a reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10881–10891, 2019.
- [4] Y. Su, R. Fan, X. Fu, and Z. Jin, "DQELR: an adaptive deep Q-network-based energy- and latency-aware routing protocol design for underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 9091–9104, 2019.
- [5] S. Jiang, "State-of-the-art medium access control (MAC) protocols for underwater acoustic networks: a survey-based on a MAC reference model," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 96–131, 2018.
- [6] 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, no. 1, pp. 5906–5919, 2021.
- [7] X. Du, "Hierarchical code assignment algorithm and state-based CDMA protocol for UWSN," *China Communications*, vol. 12, no. 3, pp. 50–61, 2015.
- [8] Y. Chen, J. Zhu, L. Wan, S. Huang, X. Zhang, and X. Xu, "ACOA-AFSA fusion dynamic coded cooperation routing for different scale multi-hop underwater acoustic sensor networks," *IEEE Access*, vol. 8, no. 1, pp. 186773–186788, 2020.
- [9] S. Zhang, X. Du, and X. Liu, "A secure remote mutual authentication scheme based on chaotic map for underwater acoustic networks," *IEEE Access*, vol. 8, pp. 48285–48298, 2020.
- [10] F. W. Rouseff, "Energy consumption model for a broadband shallow-water acoustic communications network," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 3, pp. 335–340, 2008.
- [11] J. Zhang, H. Lai, and Y. Xiong, "Concurrent transmission based on distributed scheduling for underwater acoustic networks," *Sensors*, vol. 19, no. 8, p. 1871, 2019.
- [12] M. Wang, Y. Chen, X. Sun, F. Xiao, and X. Xu, "Node energy consumption balanced multi-hop transmission for underwater acoustic sensor networks based on clustering algorithm," *IEEE Access*, vol. 8, no. 1, pp. 191231–191241, 2020.
- [13] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.
- [14] P. Karn, "MACA—a new channel access method for packet radio," in *Proceedings of the ARRL/CRRL amateur radio 9th computer networking conference*, vol. 9no. 22, pp. 134–140, London, ON, Canada, 1990.
- [15] C. L. Fullmer and J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," *The Communication Review*, vol. 25, no. 4, pp. 262–273, 1995.
- [16] Y. Zhu, S. N. Le, Z. Peng, and J. H. CuiDOS, "Distributed on-demand scheduling for high performance MAC in underwater acoustic networks," *Technical Report UbiNet-TR13-07*, University Connecticut, Storrs, CT, U-SA, 2013.
- [17] F. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "Graph colouring MAC protocol for underwater sensor networks," in *Proceedings of the 32nd IEEE International Conference on Advanced Information Net-Working and Applications (AINA)*, vol. 5, Krakow, Poland, 2018no. 16.
- [18] I. Kurtis, B. Kredo, and P. Mohapatra, "A hybrid medium access control protocol for underwater wireless networks," in *Proceedings of the Second Workshop on Underwater Networks*, vol. 9no. 14, pp. 33–40, Montreal, QC, Canada, 2007.
- [19] X. Zhuo, F. Qu, H. Yang, Y. Wei, Y. Wu, and J. Li, "Delay and queue aware adaptive scheduling-based MAC protocol for underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 56263–56275, 2019.
- [20] Z. Li, Z. Guo, H. Qu, F. Hong, P. Chen, and M. Yang, "UD-TDMA: a distributed TDMA protocol for underwater acoustic sensor network," in *Proceedings of IEEE 6th international conference on Mobile ad hoc and sensor systems*, vol. 10no. 12, pp. 918–923, Macau, China, 2009.
- [21] A. F. H. Iii, M. Stojanovic, and M. Zorzi, "Idle-time energy savings through wake-up modes in underwater acoustic networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 770–777, 2009.
- [22] L. Hong, "ECS: efficient communication scheduling for underwater sensor networks," *Sensors*, vol. 11, no. 3, pp. 2920–2938, 2011.
- [23] R. Su, R. Venkatesan, and C. Li, "An energy-efficient relay node selection scheme for underwater acoustic sensor

- networks,” *Cyber-Physical Systems*, vol. 1, no. 2-4, pp. 160–179, 2015.
- [24] F. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, “Efficient depth-based scheduling MAC protocol for underwater sensor networks,” in *Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, vol. 7no. 4, pp. 827–832, Milan, Italy, 2017.
- [25] B. Peleato and M. Stojanovic, “A MAC protocol for ad-hoc underwater acoustic sensor networks,” in *Proceedings of the 1st ACM international workshop on underwater networks*, vol. 9no. 27, pp. 113–115, Los Angeles, CA, USA, 2006.
- [26] Y. Han and Y. Fei, “A delay-aware probability-based MAC protocol for underwater acoustic sensor networks,” in *Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC)*, vol. 2no. 16, pp. 938–944, Garden Grove, CA, USA, 2015.
- [27] S. Bagali and R. Sundaraguru, “Efficient Channel Access Model for Detecting Reactive Jamming for Underwater Wireless Sensor Network,” in *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, March 2019.
- [28] L. G. Roberts, “ALOHA packet system with and without slots and capture,” *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.
- [29] R. T. B. Ma, V. Misra, and D. Rubenstein, “An analysis of generalized slotted-aloha protocols,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 936–949, 2009.
- [30] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “MACA-MN: a MACA-based MAC protocol for underwater acoustic networks with packet train for multiple neighbors,” in *VTC Spring 2008-IEEE Vehicular Technology Conference*, pp. 46–50, Marina Bay, Singapore, 2008.
- [31] M. Molins and M. Stojanovic, “Slotted FAMA: a MAC protocol for underwater acoustic networks,” in *Proceedings of the OCEANS 2006—Asia Pacific*, vol. 5no. 16, pp. 1–7, Singapore, 2007.
- [32] B. Peleato and M. Stojanovic, “Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks,” *IEEE Communications Letters*, vol. 11, no. 12, pp. 1025–1027, 2007.
- [33] A. A. Syed, W. Ye, and J. Heidemann, “T-Lohi: a new class of MAC protocols for underwater acoustic sensor networks,” in *Proceedings of the IEEE INFOCOM 2008—the 27th conference on computer communications*, vol. 4no. 13, pp. 231–235, Phoenix, AZ, USA, 2008.
- [34] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K. C.-J. Lin, “ST-MAC: spatial-temporal MAC scheduling for underwater sensor networks,” in *IEEE INFOCOM 2009*, vol. 4no. 19, pp. 1827–1835, Rio de Janeiro, Brazil, 2009.
- [35] M. Park and V. Rodoplu, “UWAN-MAC: an energy-efficient MAC protocol for underwater acoustic wireless sensor networks,” *IEEE Journal of Oceanic Engineering*, vol. 32, no. 3, pp. 710–720, 2007.
- [36] R. Diamant, M. Pinkhasevich, and I. Achrak, “A novel spatially shared TDMA protocol and quality measure for ad hoc underwater acoustic network,” in *23rd International Conference on Advanced Information Networking and Applications, AINA 2009, Workshops Proceedings*, Bradford, United Kingdom, 2009.
- [37] R. Su, R. Venkatesan, and C. Li, “An energy-efficient asynchronous wake-up scheme for underwater acoustic sensor networks,” *Wireless Communications and Mobile Computing*, vol. 16, no. 9, 1172 pages, 2016.
- [38] F. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, “An efficient scalable scheduling MAC protocol for underwater sensor networks,” *Sensors*, vol. 18, no. 9, p. 2806, 2018.
- [39] Z. Zhou, Z. Peng, J. H. Cui, Z. Shi, and A. Bagtzoglou, “Scalable localization with mobility prediction for underwater sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 335–348, 2010.
- [40] P. Casari, B. Tomasi, and M. Zorzi, “A comparison between the Tonelohi and Slotted FAMA MAC protocols for underwater networks,” in *OCEANS 2008*, Quebec City, QC, Canada, 2008.
- [41] M. Li, D. Xiujuan, X. Liu, and C. Li, “Shortest path routing protocol based on the vertical angle for underwater acoustic networks,” *Journal of Sensors*, vol. 4, no. 7, 15 pages, 2019.