QMCR: A Q-Learning-Based Multi-Hop Cooperative Routing Protocol for Underwater Acoustic Sensor Networks

Yougan Chen[1,2,3,*], Kaitong Zheng[1,3], Xing Fang[4], Lei Wan[1,5], Xiaomei Xu[1,2,3]

[1] Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education, Xiamen 361005, China
[2] Shenzhen Research Institute of Xiamen University, Shenzhen 518000, China
[3] Dongshan Swire Marine Station, College of Ocean and Earth Sciences, Xiamen University, Xiamen 361102, China
[4] School of Information Technology, Illinois State University, Normal, IL 61790, USA
[5] School of Informatics, Xiamen University, Xiamen 361005, China
[*] The corresponding author, email: chenyougan@xmu.edu.cn

Abstract: Routing plays a critical role in data transmission for underwater acoustic sensor networks (UWSNs) in the internet of underwater things (IoUT). Traditional routing methods suffer from high end-to-end delay, limited bandwidth, and high energy consumption. With the development of artificial intelligence and machine learning algorithms, many researchers apply these new methods to improve the quality of routing. In this paper, we propose a Q-learning-based multi-hop cooperative routing protocol (QMCR) for UWSNs. Our protocol can automatically choose nodes with the maximum Q-value as forwarders based on distance information. Moreover, we combine cooperative communications with Q-learning algorithm to reduce network energy consumption and improve communication efficiency. Experimental results show that the running time of the QMCR is less than one-tenth of that of the artificial fish-swarm algorithm (AFSA), while the routing energy consumption is kept at the same level. Due to the extremely fast speed of the algorithm, the QMCR is a promising method of routing design for UWSNs, especially for the case that it suffers from the extreme dynamic underwater acoustic channels in the real ocean environment.

Received: Oct. 03, 2020
Revised: Jan. 22, 2021
Editor: Yang Cao

Keywords: Q-learning algorithm; routing; internet of underwater things; underwater acoustic communication; multi-hop cooperative communication

I. INTRODUCTION

Underwater acoustic sensor networks (UWSNs) are widely adopted in environmental and military sensing undersea areas [1, 2] . With the development of the internet of underwater things (IoUT)[3–5], more and more artificial intelligence (AI) technologies have been adopted for smart ocean [6]. Generally, underwater networks consist of many sensors, which can receive and forward information through multi-hop to the destination. However, due to the harsh environment in underwater acoustic channels [7], it is still a challenging task to design the suitable UWSNs in IoUT for transmitting data.

In order to mitigate the issues such as high bit error rates (BER), high end-to-end delay, and limited frequency associated to the underwater acoustic channels, one-hop transmissions are replaced by multi-hop transmissions, which can transmit data more effectively by increasing relays. This is because multi-hop transmissions can greatly improve performance by decreasing signal attenuation and increasing available bandwidth during the transmission [8].

Transmission energy efficiency has also been a major concern in UWSNs. In underwater environment,

sensors are mainly powered by batteries, which are difficult to recharge. Therefore, energy consumption of the transmissions is a critical issue in the design of UWSNs. In practice, high BER and limited bandwidth will lead to many re-transmissions among the hops and consume additional energy. Thus, energy-efficient transmissions and low BER are the decisive factors in the design of routing protocols for UWSNs. Many routing protocols have been proposed to solve these problems. Xie et al. [9] propose a vector-based-forwarding (VBF) method that uses locations of source, sink, and relay nodes to improve energy efficiency in dynamic underwater environment. In VBF, a self-adaptation algorithm allows each node to estimate the density of nodes in its neighborhood based on local information and choose the next node accordingly. However, the method does not consider the energy status of the forwarder. The power of those frequently working forwarders can be consumed very quickly, which shortens the lifetime of the entire UWSNs. To solve this problem, lifetime-extended vector-based forwarding routing (LE-VBF) has been presented in [10]. LE-VBF takes position as well as energy information into consideration in order to increase the lifetime of UWSNs. Furthermore, the depth-based routing (DBR) protocol has been proposed in [11], which only requires location depth information. A node transmits packets to the next node which has smaller depth than itself. However, using flooding mode to send packets can generate redundant data and increase channel occupancy, which can cause high energy consumption. In depth-based multi-hop routing (DBMR) [12], the multi-hop mode of each node is used to send packets, thereby greatly decreases the energy consumption.

With the development of AI, many intelligent algorithms, such as ant colony algorithm (ACA), simulated annealing algorithm (SAA), artificial fish-swam algorithm (AFSA), and Q-learning algorithm have been adopted for the routing design of UWSNs. ACA [13] can find the optimal routing by simulating the food-seeking activities of ants. However, ACA suffers from long searching time, easily getting stuck in local optima. In [14], a Q-learning-based adaptive routing (QELAR) protocol has been proposed to extend the lifetime of UWSNs. The QELAR uses a reward function based on energy consumption. After the training of an AI agent, the agent can automatically choose the routing which consumes the least energy. QELAR has been proved to be superior to VBF routing protocol. In [15], Jin et al. propose a reinforcement-learning-based congestion-avoided routing (RCAR) protocol to reduce the end-to-end delay and energy consumption for UWSNs, where congestion and energy are both considered for adequate routing decision. In [16], Lu et al. propose an energy-efficient depth-based opportunistic routing algorithm with Q-learning (EDORQ) for UWSNs, which combines advantages of the Q-learning technique and opportunistic routing (OR) algorithm. It is shown to be able to achieve guarantee the energy-saving and reliable data transmission.

On the other hand, cooperative communications [17] have been introduced into UWSNs [18, 19] recreantly, because the broadcast nature of wireless signals can enhance transmission quantity at the receiver side. During the transmission, a source node broadcasts signals to a cooperative node and a receiver. The receiver can then receive signals from both the source node and the cooperative node. With the help of the two identical copies of the signal, the probability of packet reception and success rate of decoding can be increased in the receiver. To the best of our knowledge, until now, Q-learning algorithm has never been applied to multi-hop underwater acoustic cooperative communications to further enhance its performances.

In this paper, we propose a Q-learning algorithm based multi-hop underwater acoustic cooperative routing protocol (QMCR) for the UWSNs. A new concern of the multi-hop UWASNs is that when there are both relay nodes and cooperative nodes, the difficulty of routing selection increases. How to give full play to the gain of cooperative communication and the advantages of Q-learning is the main issue that we consider in this paper. The main contributions of this paper are as follows:

1) We present an improved protocol, named QMCR, based on Q-learning algorithm and cooperative communication technique for the UWSNs. In the proposed QMCR, we set the reward function based on energy consumption for each transmission link of the multi-hop UWSNs; then we train the AI agent to choose the main routing that consumes the minimum energy; finally we apply cooperative communication technique to choose a cooperative routing to make the entire energy consumption the lowest. Extensive experiments have shown that the QMCR can automatically choose

the routing with low transmission energy consumption efficiently.

2) The short running time of the proposed QMCR is especially designed for the harsh underwater acoustic channels in the real ocean environment. In order to timely find a suitable routing when the network topology has been changed due to the rapidly changing underwater acoustic channels, the running time is a major concern. Compared with other routing protocols, the running time of QMCR is less than one-tenth of that of other intelligent algorithms while the energy consumption of the selected route is almost the same. This is because QMCR spends most of its time on training the AI agent, and once the training is finished, the AI agent can automatically choose the best route. However, other algorithms such as ACA and AFSA need to calculate the energy consumption of all the candidates of routing, causing a slow reaction to the dynamic network topology of UWSN. Therefore, the proposed QMCR is quite appealing to the extremely dynamic underwater acoustic channels in the real ocean environment.

The rest of this paper is organized as follows. In Section II, basic concept of the Q-learning and its application in routing design for UWSNs is introduced. Section III presents underwater acoustic channels and energy consumption model for underwater acoustic transmissions and combines Q-learning with cooperative communications. Section IV shows the simulation results. Finally, we conclude this paper in Section V.

II. Q-LEARNING AND ITS APPLICATION TO ROUTING DESIGN IN UWSNS

2.1 Q-Learning Technique

Reinforcement learning (RL) is a method that trains an AI agent to learn from its interaction with the environment where it works. The main purpose of the training is to ensure that the agent can automatically find the action with maximum reward in any situation. After choosing an action in a certain situation, the outcome will be used to give a positive or negative reinforcement to the agent.

Q-learning is one of the RL algorithms that can be seen as a Markov decision process [20]. As Figure 1 shows, at each step the agent selects a random action under the particular environment. Then, the agent will receive a probabilistic reward whose value depends on current state and the corresponding action. At the same time, the current state updates to the next step. After learning from the outcome, agent takes next action based on previous experience. The process does not stop until the agent achieves the goal we set at the beginning. When the process stops, one iteration time is completed. Every iteration operation forms a limited sequence of states, actions and rewards.

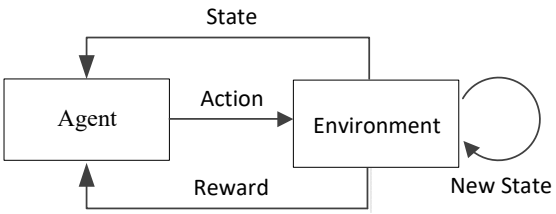


Figure 1. *Framework of Q-learning.*

At each step, the agent learns from its past and remembers it as Q value. The updating of Q value can be defined by the following expression:

$$Q(s,a) = R(s,a) + \delta \cdot \max\{Q(s',a')\}, \quad (1)$$

where s denotes current state; a denotes current action; s' is the next state after taking this action; a' is all the actions that the next state s' can take. $R(s,a)$ denotes the current reward, and $\max\{Q(s',a')\}$ denotes the largest Q value we can get in the next state. It means Q value is not only determined by current reward but also future reward. δ is the discount factor, where $0 < \delta < 1$. If the discount factor is larger, the agent will put more weight on future reward than current reward. In general, the value of $Q(s,a)$, which takes both current reward and future reward into consideration, can measure whether the action in the current state should be taken or not. Q table is a matrix that documents the Q value in each state and each action. The size of the Q table is $m \times n$, where m is the number of states and n is the number of actions. All the Q values are initialized to zero prior to the training.

2.2 Q-Learning for Routing Design in UWSNs

Adopting the Q-learning technique in UWANs for IoUT, a simple example of routing design without considering cooperative nodes can be given as below [21]:

Suppose that there are six nodes in an UWSN, and each of them can receive and transmit packets. The connection between each node is shown in Figure 2(a). The circles represent sensor nodes, and the arrows represent transmission directions. Each sensor node is a state and each arrow represents an action. In this example, there are six states and thirteen actions. We can see from Figure 2(a) that both node 0 and node 2 have only one node to connect. We set node 5 as the destination. At first, a packet is sent from a random node. The reward of an action, which can lead to the destination, is set as 100. Otherwise, the reward is set to be 0. The reward of impossible actions such as node 2 to node 0 is set as -1. As agent can reach the destination by choosing action 5 in state 1, the corresponding reward $R(1,5)$ is 100. If the agent chooses action 4 in state 0, the reward will be 0 since action 4 does not lead the agent to the destination. Impossible actions, such as action 2 in state 0, have the negative reward. The reward distribution and reward matrix are shown in Figure 2(c) and Figure 3(a).

At the beginning of the training process, a Q table is initialized with zeros as Figure 3(b) shown. We set the discount factor δ as 0.8, and choose state 1 as the source node. If the action is randomly selected as 5, $Q(1,5)$ can then be calculated and updated based on Eq. (1):

$$\begin{aligned} Q(1,5) &= R(1,5) + 0.8 \times \max\{Q(5,1), Q(5,2), \\ &\quad Q(5,3), Q(5,4), Q(5,5)\} \\ &= 100 + 0.8 \times \max\{0,0,0,0,0\} \\ &= 100. \end{aligned} \tag{2}$$

Then, we update the Q table shown as Figure 3(c). The training process will be stopped when values in the Q table are almost stable. The final Q table is as shown in Figure 3(d). In order to view the outcome more directly, every Q value can be presented in the connection relation figure as shown in Figure 2(c).

In Figure 2(c), the value of arrow denotes the corresponding Q value. Red arrow is the action with the largest Q value, namely the best action corresponding to each node. After training, the agent is ready for choosing the best routing in any situation. For example, suppose there is a packet sent from node 3. Since $Q(3,1)$ is greater than $Q(3,2)$, the packet will be transmitted to node 1. In state 1, $Q(1,5)$ is greater than $Q(1,3)$, therefore the agent will choose action 5 that leads to the destination. In particular, if there are more than two actions with identical Q values to choose from, one of them will be chosen randomly as the next action. For example, in Figure 2(c), $Q(3,1)$ and $Q(3,4)$ are equal, so action 1 or action 4 will be randomly selected as the next action.

The process of Q-learning for UWSNs routing design in IoUT can be summarized as Algorithm 1.

Algorithm 1. *Q-learning for UWSNs routing design.*

Initialization: According to the number of sensor nodes and the topology of UWSN, set the parameters, including discount factor δ and reward values in the reward table; initialize Q table to record Q value.

1: **for** $i = 1, 2, \ldots,$ **do**
2: **Step 1**: Randomly choose a beginning and randomly take actions according to the reward table and Q table; the goal is to let the agent reach the destination;
3: **Step 2**: Update Q table according to Eq. (1);
4: **Step 3**: Compare its new Q value to the last Q value, and do judgement:
5: **if** Q table is changed **then**
6: update the Q table again from step 1:
7: **else if** the Q table is almost unchanged **then**
8: stop updating the Q table and break out.
9: **end if**
10: **end for**
11: According to the stable Q table, select the optimal routing.

III. DESIGN OF Q-LEARNING ALGORITHM IN MULTI-HOP COOPERATIVE UWSNS

3.1 Underwater Acoustic Communication Energy Consumption Model

The energy consumption model of underwater acoustic communication is different from the model of terrestrial communication. Since it requires much less power in receiving packets than transmitting, we only consider the power required for transmitting at this moment. The energy consumption model of point-to-point underwater acoustic communication in [22] is

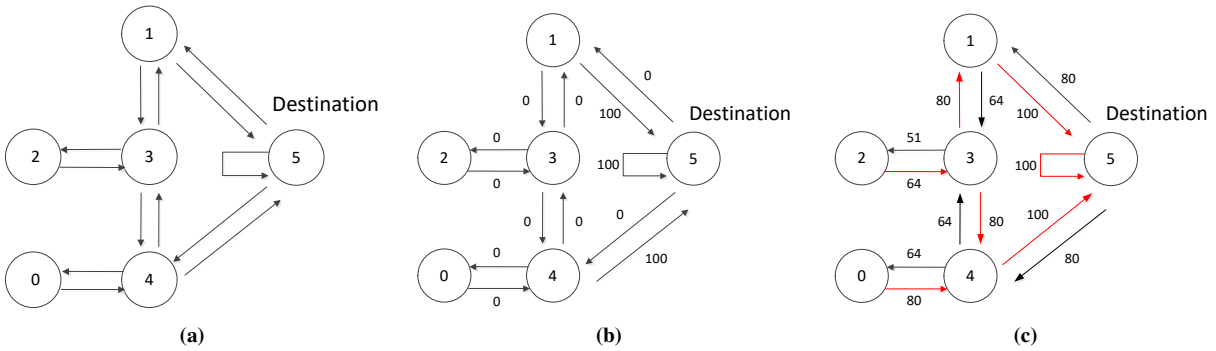


Figure 2. *An example of UWSN for Q-learning: (a) Connection between each node; (b) Reward distribution; (c) Connection relation with Q value.*

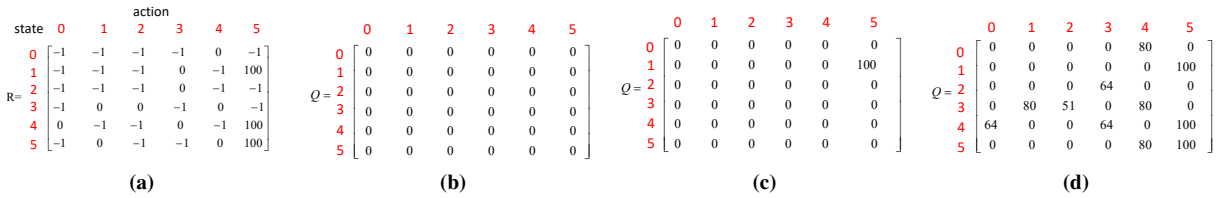


Figure 3. *Reward table and Q table for Q-learning: (a) Reward table; (b) Initialized Q table; (c) Q table after one update; (d) Q table after training.*

shown below.

The attenuation of the power relative to the distance d is $U(d)$, which can be described as

$$U(d) = (1000 \cdot d)^{\kappa} \cdot \alpha^{d}, \tag{3}$$

$$\alpha = 10^{\frac{\gamma(f)}{10}}, \tag{4}$$

$$\gamma(f) = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + \frac{2.75f^2}{10^4} + \frac{3}{10^3}, \tag{5}$$

where $\gamma(f)$ is the absorption coefficient. Under different communication conditions, the value of κ is different. In surface channel or deep-sea channel where wave propagates cylindrically, $\kappa = 1$; in surface channel or deep-sea channel where wave propagates cylindrically with the absorption of seabed, $\kappa = 1.5$; in open water where wave propagates spherically, $\kappa = 2$. The choice of f is based on the empirical formula of the optimal working frequency and working distance [22],

$$f_{opt} = \left(\frac{200}{d}\right)^{\frac{2}{3}}, \tag{6}$$

where d represents the distance between two adjacent nodes under the unit of kilometers.

Let P_0 be the lowest power level at which a data packet can be correctly decoded by the receiver, thus the lowest transmission power of the node P can be written as

$$P = P_0 \cdot U(d). \tag{7}$$

With Eq. (3) to Eq. (7) in hand, we can obtain the transmission power for each transmitting. Then the energy consumption can be calculated and more details can be found in [18]. From the underwater acoustic transmission loss or energy consumption model, we can observe that distance is the most critical parameter in determining energy consumption. For convenience, we can express the energy consumption cost function $L(d)$ as the positive correlation with the attenuation of the power $U(d)$,

$$L(d) \propto U(d). \tag{8}$$

Therefore, we design the reward table of Q-learning technique mainly based on the distance for the multi-hop cooperative UWSNs.

3.2 Combination Design of Q-Learning Technique and Cooperative UWSNs

In the multi-hop cooperative UWSNs of IoUT, the system model can be shown as Figure 4. The source node is denoted as S, the destination node is denoted as D, the relay node is represented as R_i, and the cooperative node is represented as $C_i (i = 1, 2, \ldots, N-1)$, where N is the number of the hops.

In Figure 4, for each hop of the network, say "$R_i - C_{i+1} - R_{i+1}$", it adopts the dynamic coded cooperation (DCC) scheme as we investigated in [18] and [19], where the cooperative node C_{i+1} can enhance the data transmission from relay R_i to relay R_{i+1} by exploiting the benefit of rate-compatible coding. Further details of DCC transmission design for "$R_i - C_{i+1} - R_{i+1}$" group and its application in multi-hop scenario can be found in our previous work [18, 19].

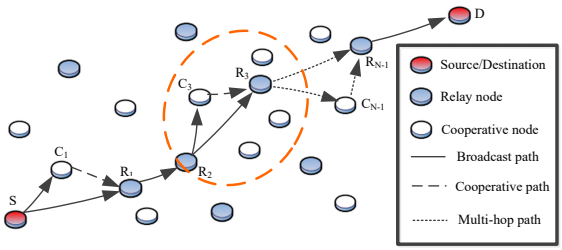


Figure 4. *Multi-hop cooperative UWSNs model.*

The task of design the optimal routing in this multi-hop cooperative UWSN is to find the routing on the lowest transmission energy consumption with the help of cooperative nodes.

The most important thing in Q-learning is the design of reward table. In order to apply Q-learning to routing design, we use the negative distances as the rewards. For example, if the distance from node 1 to node 2 is 2 km, and the distance from node 1 to node 3 is 3 km, then the reward of taking action 2 in state 1 is -2, and the reward of taking action 3 in state 1 is -3. After the iteration, $Q(1,2)$ will be greater than $Q(1,3)$, so the agent will choose to take action 2 in state 1. Since the distance is the shortest, the routing is the one with the smallest transmission energy consumption based on the energy consumption model according to Eq. (8).

For the case without cooperative node, the energy consumption cost function $L(d_{ij})$ in state k is expressed as:

$$L_{k,ij} \propto U(d_{ij}), \tag{9}$$

$$\Psi_{Total} = \Sigma L_{k,ij}, \tag{10}$$

where $L_{k,ij}$ represents the energy consumption cost function from node R_i to node R_j during the transmission in state k, d_{ij} is the distance between the two nodes. The total energy consumption (from source node to destination node) cost function is the sum of each link of $L_{k,ij}$, and denoted as Ψ_{Total}.

For the case with cooperative node, the energy consumption cost function is expressed as

$$L_{k,ij} \propto \frac{U(d_{ij}) + \triangle \cdot U(d_{cjj})}{1 + \triangle}, d_{ij} \leq r_2, d_{cjj} < r_1, \tag{11}$$

$$\triangle = \begin{cases} 0, & d_{ij} \leq r_1 \\ 1, & d_{ij} > r_1 \end{cases}, \tag{12}$$

where r_1 is the furthest distance that the data packet can be decoded correctly without the help of cooperative node, r_2 is the furthest distance that the data can be decoded correctly with the help of cooperative node, d_{ij} represents the distance from the current node R_i to the next node R_j, and d_{cjj} represents the distance from the cooperative node C_j to the next node R_j. In the case where the cooperation is not satisfied, $\triangle$ takes 0 and the cost function is equal to $U(d_{ij})$. If the cooperation is satisfied, $\triangle$ takes 1 and the cost function is equal to the average of $U(d_{ij})$ and $U(d_{cjj})$. The total energy consumption of the network is equal to the summation of the energy consumptions of every two adjacent nodes.

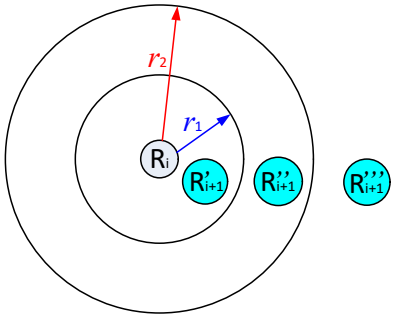


Figure 5. *Node location condition for cooperative communication.*

As shown in Figure 5, when cooperative communication is adopted, the distance d_{ij} between two nodes

should satisfy the range $r_1 < d_{ij} < r_2$. When the distance d_{ij} is too small, such as node $R'_{i+1}(d_{ij} < r_1)$, the receiving node can successfully receive and decode the signal from the source node without the help of cooperative node. When the distance d_{ij} is too large such as node $R'''_{i+1}(d_{ij} > r_2)$, the receiving node can not successfully decode the signal from the source node even with the help of a cooperative node. Therefore, only when the distance d_{ij} between the two nodes satisfies $r_1 < d_{ij} < r_2$, such as node R''_{i+1}, the cooperative node is necessary and effective.

There may be multiple cooperative nodes to choose from between any two nodes. As shown in Figure 6, there are five candidate cooperative nodes around node R_i and node R_{i+1}. Since node C_5 is not between node R_i and node R_{i+1}, it cannot be a cooperative node. In the remaining four candidate nodes, according to the cost function Eq. (11), the energy consumption of each candidate node will be calculated. The one with the lowest energy consumption will be selected as the cooperative node.

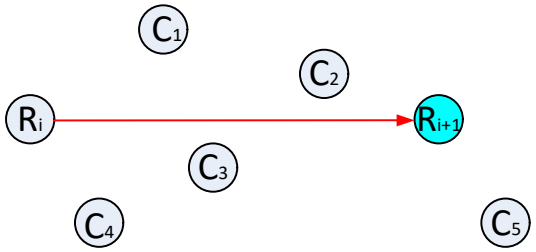


Figure 6. *The case with multiple cooperative nodes.*

The idea of designing Q-learning algorithm for the multi-hop cooperative UWSNs is as follows: First, the main route is selected by Q-learning, and the distance d_{ij} of each node in the main path should satisfy $r_1 < d_{ij} < r_2$ (the range of cooperative communication). Then, according to the main route, the cooperative nodes that can minimize the transmission energy consumption will be calculated. At last, the two are combined to obtain the final routing scheme.

Specific steps are as follows:

1) Parameter initialization. Input node location information and set parameters (including number of nodes, number of iterations, learning rate, discount factor, number of states, number of actions, source node, destination node).

2) Design the reward table. The distance between every two nodes is calculated according to the node locations, and then is stored in a matrix called edge. For example, the distance from node 1 to node 3 is 3 km, then edge(1,3)=edge(3,1)=3. Next, each element in the matrix edge takes a negative value as its reward value for training the agent. Due to the smaller the distance, its value is larger after taking the negative, and the agent tends to get a bigger reward. Hence the agent will preferentially transmit the data to the nearest node. In order to prevent the agent from transmitting data from source node to itself, the reward for performing action k in state k is set to be -1000 (approximately infinitesimal).

For cooperative communications, we set the distance threshold $r_1 = 2.5$ km, $r_2 = 4$ km as an example. The reward between nodes with too small distance ($d_{ij} < r_1$) and too large distance ($d_{ij} > r_2$) is set to be -1000 (approximate infinitesimal) to meet the condition of cooperative communication.

3) The iteration of training the Q table. Create a zero matrix with size equal to the number of states multiplied by the number of actions as the original Q table (the number of states and the number of actions equals the number of nodes in the UWSNs). The iteration process is shown as follows: **i)** randomly select a state and an action with reward greater than -1000; **ii)** the environment feedbacks the current reward value; **iii)** search for the maximum Q value of the next state; calculate the new Q value according to Eq. (1); **iv)** continue to select an action in the next state and repeat the above process until the destination node is reached. Above steps consist one iteration. The final number of iterations required to achieve a stable Q table depends on the specific network environment.

4) Correction of the Q table. It is necessary to correct the Q table; otherwise, the agent may fall into local optimum. For example, the distance between node 3 and node 4 is very small. The Q value of taking action 4 in state 3 and taking action 3 in state 4 is very large so the data packet will be transmitted back and forth between node 3 and node 4. For this problem, if the agent is in state 3, the Q value of action 3 for all other states is set to be -1000, which means the data packet will not return to state 3 in any situation. It ensures that the data packet can be transmitted forward smoothly.

5) Use the agent to select the main routing. For specific source and destination nodes, the agent starts

from the source node and selects the action that maximizes the Q value according to the Q table until it reaches the destination node. Record the nodes selected by the agent each time to form the main route.

6) Select the cooperative nodes. In each hop of the main route, if there are multiple nodes that can be chosen as the cooperative node, use the energy consumption cost function to calculate and select the node that minimizes energy consumption as the cooperative node. Record each cooperative node and then the cooperative routing is obtained.

7) Performance output. Calculate total transmission energy, algorithm running time and draw the routing map.

IV. NUMERICAL RESULTS

In this section, we present the simulation results of the proposed QMCR for the multi-hop cooperative UWSNs. The simulation is carried out based on the MATLAB software platform, the computer operating system is Windows 10 (64-bit), the CPU is i7-7700, and the memory is 8 GB. We compare the performance of QMCR and AFSA protocols to evaluate the superiority of the proposed scheme.

4.1 Simulation Setup

As shown in Figure 7, a total of 18 nodes are arranged, where each circle represents a sensor node, and the number represents the index of nodes. The source node S is node 1 and the destination node D is node 18. In order for the transmission to proceed smoothly, the 18 nodes are located in a rectangular area with around 14 km length and 5 km width.

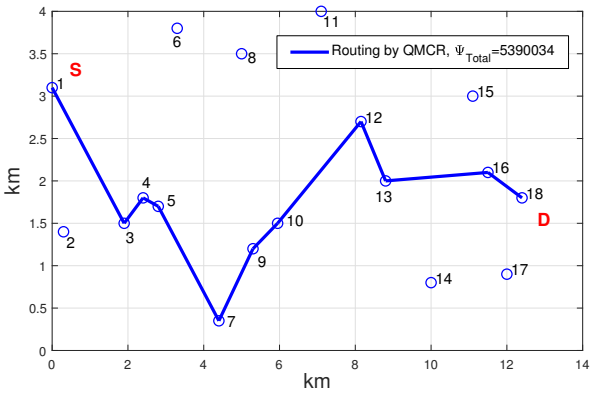


Figure 7. *Routing selected by QMCR without cooperation.*

It is assumed that underwater acoustic communications are carried out in the shallow sea and the sound waves mainly propagate in the form of cylindrical waves, i.e., set $\kappa = 1.5$ in Eq. (3). According to Eq. (11) and Figure 5, we set the distance threshold $r_1 = 2.5$ km, $r_2 = 4$ km in the simulation. If the distance between two nodes in a hop is too far to decode correctly ($d_{ij} > r_2$), the topology structure of the UWSNs should be reconsidered or we should add the transmitting power and reset the cooperative region design.

4.2 Simulation Results

First we evaluate the performance of QMCR and AFSA without the cooperation strategy (According to Eqs. (11) and (12), when there is no cooperative node, QMCR algorithm essentially degenerates into a general Q learning algorithm). The maximum number of iterations is set as 1000, the discount factor δ is set as 0.8.

Figure 7 shows the selected routing result of QMCR without cooperative nodes. The solid blue line represents the selected routing, which is 1-3-4-5-7-9-10-12-13-16-18. The total value of energy consumption cost function for the transmission is 5390034. The running time of QMCR algorithm is 0.23 seconds.

Figure 8 is the result of AFSA without cooperative nodes. The selected routing is 1-3-4-5-7-9-10-12-13-14-16-18. The total value of energy consumption cost function for the transmission is 5724767. The running time of AFSA algorithm is 17.51 seconds.

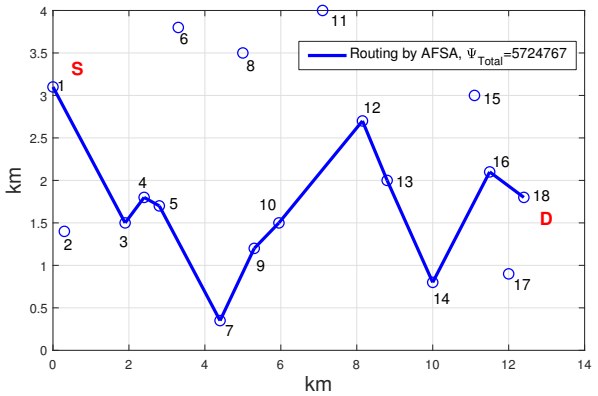


Figure 8. *Routing selected by AFSA without cooperation.*

It can be concluded that the routing selected by Q-learning algorithm is one hop less than that selected

by AFSA. Especially, compare to AFSA, the running time of Q-learning algorithm reduces from 17.51 seconds to 0.23 seconds, which is a drop of 98%, and the total transmission energy consumption has also reduces 5%. It can be seen that the Q-learning algorithm performs faster and the selected routing consumes less energy.

Next, we evaluate the performance of these two algorithms for the case with cooperative nodes.

In Figure 9, the solid blue line represents the main routing and the dotted red line represents the cooperative routing selected by QMCR. The trained agent first selects the main routing 1-4-9-13-18, and then finds the best cooperative nodes 3, 7, 12, 16 with the lowest transmission energy. The total value of energy consumption cost function for the transmission is 3251576. The running time of QMCR algorithm is 0.8 seconds. Compared with the results in Figure 7, it can be observed that the transmission energy consumption of QMCR is only 60% of that of non-cooperative Q-learning algorithm.

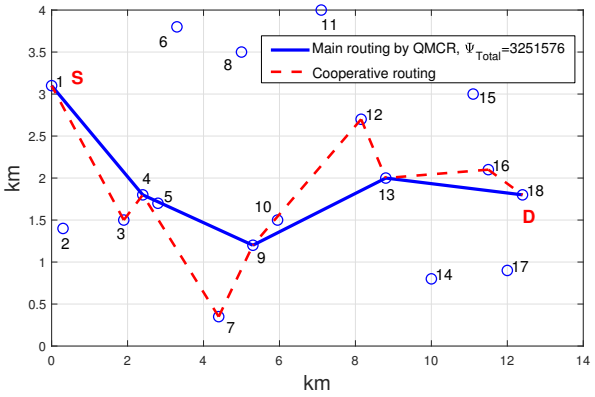


Figure 9. *Routing selected by QMCR for the case with cooperation.*

Figure 10 is the routing selected by AFSA with cooperative nodes. The main routing selected is 1-5-10-13-18, and the cooperative nodes are 4, 9, 12, 16. The total value of energy consumption cost function for the transmission is 3071395. The AFSA algorithm takes 10.81 seconds in this case.

Compared to the AFSA, the running time of QMCR is decreased into 7.4% of that in AFSA. The transmission energy consumption of QMCR is 5% higher than that of AFSA. In the cooperative case, the running time of QMCR is still much smaller than that of AFSA while the transmission energy consumption is similar.

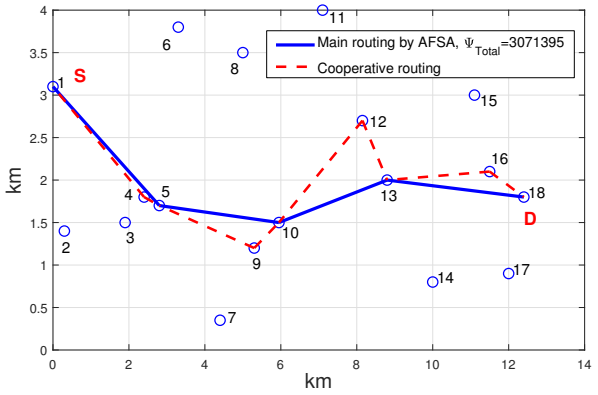


Figure 10. *Routing selected by AFSA for the case with cooperation.*

4.3 Comparison and Discussion

In order to analyze the advantages and disadvantages of QMCR and AFSA, we apply the two algorithms to the same underwater acoustic communication environment to observe the main differences.

The resulting comparisons are shown in Table 1.

From the comparison above, it can be concluded that QMCR is superior to AFSA in terms of algorithm running time, transmission energy consumption and stability for the case without cooperation. Especially the running time of QMCR is much less than that of AFSA with a drop from 17.51 seconds to 0.23 seconds, which is a 98% reduction. This is because the core of QMCR is using the reward table to continuously optimize the Q table. When selecting the routing, the agent selects the action with the largest Q value to complete routing selection in one time. However, the AFSA first selects a routing, and then continuously selects a different routing to choose a better one. That is to say, the QMCR only needs to choose the routing and calculate the energy consumption once while the AFSA needs to evaluate multiple routings. Therefore, the AFSA takes much more time than the QMCR.

For the case with cooperation, the running time is 0.8 seconds for QMCR, and 10.8 seconds for AFSA. The running time of QMCR is about $1/10$ of that of AFSA. In addition, the QMCR has a high convergence. Hence both the running time and stability of QMCR are better than those of AFSA, but the transmission energy consumption is slightly larger than that

Table 1. *Comparison between AFSA and QMCR.*

Cases	Items	AFSA	QMCR
Non-Cooperation	Time	17.51 s	0.23 s
	Energy	5724767	5390341
	Routing	1-3-4-5-7-9-10-12-13-14-16-18	1-3-4-5-7-9-10-12-13-16-18
	Convergence	Low	High
Cooperation	Time	10.8 s	0.8 s
	Energy	3071396	3251576
	Routing	1-5-10-13-18	1-4-9-13-18
	Cooperative node	4,9,12,16	3,7,12,16
	Convergence	Normal	High

of AFSA. This is because the AFSA continuously optimizes and iterates the routing as a whole. However, the QMCR is too complicated to directly take cooperative nodes into consideration in the reward table. The QMCR first uses the Q-learning algorithm to calculate the main routing and then selects the cooperative node that minimizes the transmission energy. Hence the transmission energy consumption is slightly higher.

In summary, if we need to quickly choose the routing in a dynamic underwater environment, the QMCR is the better choice since its computational complexity and running time are much lower than other algorithms. The AFSA can be used when there is enough time to run the algorithm.

4.4 Iterations Analysis of QMCR Algorithm

Figure 11 shows the energy consumption of the selected routing of QMCR with respect to the number of iterations. The energy consumption of the selected routing is recorded during iterations of training the Q table. If the energy consumption of the selected routing becomes stable, it means that the Q table has converged, and the training is completed. Therefore, we can estimate the times of training required for QMCR to avoid unnecessary training steps and reduce the running time.

Since the Q table has not been completed in the first 300 iterations, the energy consumption of the routing is too large to be displayed. Therefore, the horizontal axis starts from the 300-th iteration. After the number of iterations reaches about 550, the energy consumption of the selected routing becomes stable. In summary, through about 550 iterations, the Q table is basically stable, which indicates the iteration is over

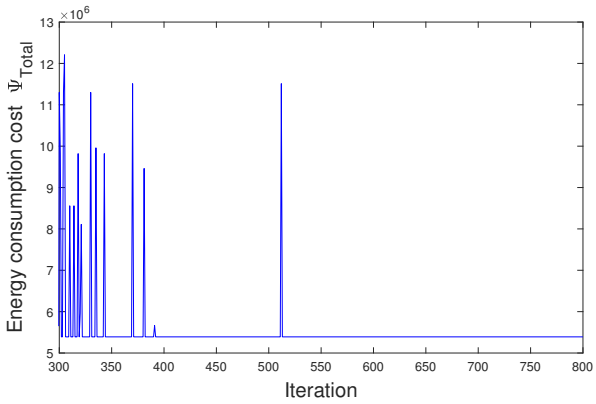


Figure 11. *The iterations analysis of QMCR.*

and the agent training process is completed.

4.5 Performance in Dynamic Underwater Environments

In the case where the sea state is stable, the underwater transducers are in relatively fixed positions; in the case where the sea wind is heavy, the positions are relatively unstable, as shown in Figure 12.

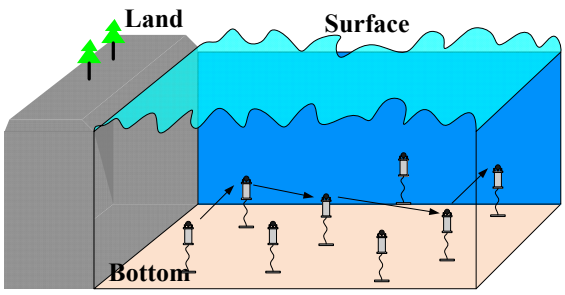


Figure 12. *A UWSN when the sea state is not stable.*

Due to the complex and varied marine environ-

ments, the drift of the nodes may occur. Therefore, in order to simulate the dynamic drift of the network nodes, the horizontal and vertical coordinates of the relay nodes can be randomly added by -0.5 km to $+0.5$ km based on the fixed positions of the 18 nodes during the simulation. Note that, in the actual ocean scenario, the drifts of network nodes are complicated with horizontal, vertical, and oblique movements. In our simulation analysis, it is assumed that the node moves around a fixed position and moves randomly within the range of ± 0.5 km. That is to say, no matter to which direction the node moves, it only moves within the range of ± 0.5 km around the fixed position. In fact, because the acoustic wave radiates omni-directionally, it covers a three-dimensional space. No matter the node moves in horizontal, vertical, and oblique directions, as long as the receiver node is in the three-dimensional space and within an appropriate range of the acoustic radiation of the transmitter node, it can receive and decode the signal.

In the simulation process, the possible drift ranges of relay nodes and cooperative nodes are shown in Figure 13. Hence the topology of the dynamic network in Figure 13 can be used to verify the effectiveness of the proposed QMCR when the nodes are drifting with the ocean current.

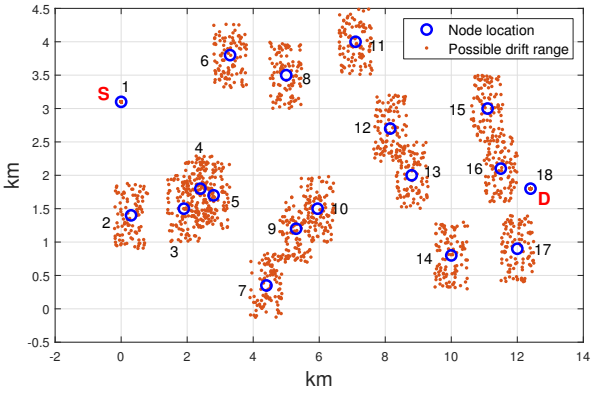


Figure 13. *Possible drift ranges of sensor nodes.*

Figure 14 shows the routing selected by QMCR under the condition of network nodes drift. It can be observed that after nodes movement, the QMCR can still select a new route quickly according to the new network topology. According to the comparison in Table 1, we know that the running time of QMCR is much less than that of other algorithms. Therefore, the proposed QMCR is very suitable for the actual situation of network node drift caused by the dynamic changes in the marine environment. It means that the proposed QMCR has strong adaptivity and can be applied to complex and variable marine environment.

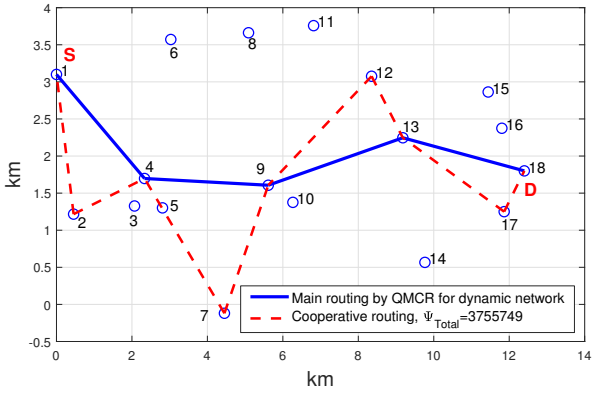


Figure 14. *Routing selected by QMCR for dynamic network.*

V. CONCLUSION

In this paper, we have proposed a novel routing protocol, named QMCR, for UWSNs in IoUT based on cooperative technique and Q-learning algorithm. A suitable reward table is designed for the underwater acoustic communication model. Through the continuous updating and optimization of Q table, the agent can quickly and automatically select the optimal transmission routing compare with other routing algorithms. The method greatly reduces the amount of calculation and saves the running time of the algorithm. Through simulation and comparison with the AFSA, it can be clearly observed that the proposed QMCR can reduce the time consumption of the algorithm by ten or even dozens of times while ensuring that the transmission energy of the selected routing is approximately unchanged. In addition, the proposed QMCR is a general framework, which can be easily adapted for different applications by adding new factors into the reward function. In the future, we can take metrics such as residual energy of each node, the density of nodes and even topography of UWSNs into consideration. Also, we can tune the weight of each parameter to ensure the highest performance of the protocol, which is more attractive for the complex underwater acoustic channels in UWSNs.

ACKNOWLEDGEMENT

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2016YFC1400200, in part by the Basic Research Program of Science and Technology of Shenzhen, China under Grant No. JCYJ20190809161805508, in part by the Fundamental Research Funds for the Central Universities of China under Grant No. 20720200092, in part by the Xiamen University's Honors Program for Undergraduates in Marine Sciences under Grant No. 22320152201106, and in part by the National Natural Science Foundation of China under Grants No. 41476026, 41976178 and 61801139. Shenzhen Research Institute of Xiamen University and the Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education, contributed equally to this work.

References

[1] H. Tran-Dang and D.-S. Kim, "Channel-aware energy-efficient two-hop cooperative routing protocol for underwater acoustic sensor networks," *IEEE Access*, vol. 7, no. 1, 2019, pp. 63 181–63 194.

[2] F. Shi, Z. Chen, *et al.*, "Behavior modeling and individual recognition of sonar transmitter for secure communication in uasns," *IEEE Access*, vol. 8, no. 1, 2020, pp. 2447–2454.

[3] J. Yan, Y. Gong, *et al.*, "AUV-aided localization for internet of underwater things: a reinforcement learning-based method," *IEEE Internet Things J.*, vol. 7, no. 10, 2020, pp. 9728–9746.

[4] X. Zhuo, M. Liu, *et al.*, "AUV-aided energy-efficient data collection in underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 7, no. 10, 2020, pp. 10 010–10 022.

[5] T. Qiu, Z. Zhao, *et al.*, "Underwater internet of things in smart ocean: system architecture and open issues," *IEEE Trans Industr. Inform.*, vol. 16, no. 7, 2020, pp. 4297–4307.

[6] Y. Gou, T. Zhang, *et al.*, "DeepOcean: A general deep learning framework for spatio-temporal ocean sensing data prediction," *IEEE Access*, vol. 8, no. 1, 2020, pp. 79 192–79 202.

[7] J. Zhou, H. Jiang, *et al.*, "Study of propagation channel characteristics for underwater acoustic communication environments," *IEEE Access*, vol. 7, no. 1, 2019, pp. 79 438–79 445.

[8] W. Zhang, M. Stojanovic, *et al.*, "Analysis of a linear multihop underwater acoustic network," *IEEE J. Oceanic Eng.*, vol. 35, no. 4, 2010, pp. 961–970.

[9] P. Xie, J. Cui, *et al.*, "VBF: Vector-based forwarding protocol for underwater sensor networks," in *NETWORKING 2006*. IFIP, 2006, pp. 1216–1221.

[10] X. Xiao, X. P. Ji, *et al.*, "LE-VBF: Lifetime-extended vector-based forwarding routing," in *2012 International Conference on Computer Science and Service System*. IEEE, 2012, pp. 1201–1203.

[11] H. Yan, Z. J. Shi, *et al.*, "DBR: Depth-based routing for underwater sensor networks," in *NETWORKING 2008*. IFIP, 2008, pp. 72–86.

[12] G. Liu and Z. Li, "Depth-based multi-hop routing protocol for underwater sensor network," in *The 2nd International Conference on Industrial Mechatronics and Automation*. IEEE, 2010, pp. 268–270.

[13] J. Zhang, H. Liu, *et al.*, "The improvement of ant colony algorithm and its application to tsp problem," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009, pp. 1–4.

[14] Y. Hu, Tiansi an Fei, "QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE T. Mobile Comput.*, vol. 9, no. 6, 2010, pp. 796–809.

[15] Z. Jin, Q. Zhao, *et al.*, "RCAR: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 19, no. 2, 2019, pp. 10 881–10 891.

[16] Y. Lu, R. He, *et al.*, "Energy-efficient depth-based opportunistic routing with q-learning for underwater wireless sensor networks," *Sensors*, vol. 20, no. 4, 2020, p. 1025.

[17] C. Carbonelli and U. Mitra, "Cooperative multihop communication for underwater acoustic networks," in *Proceedings of the 1st ACM international workshop on Underwater networks*. ACM, 2006, pp. 97–100.

[18] Y. Chen, X. Jin, *et al.*, "Selective dynamic coded cooperative communications for multi-hop underwater acoustic sensor networks," *IEEE Access*, vol. 7, no. 1, 2019, pp. 70 552–70 563.

[19] Y. Chen, Z.-H. Wang, *et al.*, "OFDM modulated dynamic coded cooperation in underwater acoustic channels," *IEEE J. Oceanic Eng.*, vol. 40, no. 1, 2015, pp. 159–168.

[20] C. Watkins and P. Dayan, "Technical note: Q-learning," *IEEE J. Oceanic Eng.*, vol. 3-4, no. 1, 1992, pp. 279–292.

[21] K. Teknomo, "Q-learning by examples," 2019, https://people.revoledu.com/kardi/tutorial/ReinforcementLearning/ Accessed Dec 5, 2019.

[22] R. J. Urick, *Principles of Underwater Sound*. Los Altos, CA,USA: Peninsula Pub, 1983.

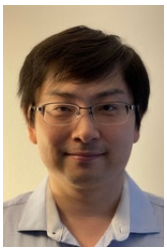
Biographies

Yougan Chen (Senior Member, IEEE) received the B.S. degree from Northwestern Polytechnical University (NPU), Xi'an, China, in 2007, and the Ph.D. degree from Xiamen University (XMU), Xiamen, China, in 2012, all in communication engineering.

He visited the Department of Electrical and Computer Engineering, University of Connecticut (UCONN), Storrs, CT, USA, from November 2010 to November 2012. Since 2013, he has been with the College of Ocean and Earth Sciences, XMU, where he is currently an Associate Professor of applied marine physics and engineering. He has authored or coauthored more than 70 peer-reviewed journal articles/conference papers and holds

more than 16 China patents. His research interest includes the application of electrical and electronics engineering to the oceanic environment, with recent focus on cooperative communication and artificial intelligence for underwater acoustic channels.
Dr. Chen has served as the Secretary for IEEE ICSPCC 2017 and the TPC Member for IEEE ICSPCC 2019. He received the Technological Invention Award of Fujian Province, China, in 2017. He has served as the Technical Reviewer for many journals and conferences, such as IEEE Journal Of Oceanic Engineering, IEEE Transactions On Communications, IEEE Access, Sensors, IET Communications, and ACM WUWNet Conference. He has been serving as an Associate Editor for IEEE Access, since 2019, and the Youth Editorial Board Member for the Journal of Electronics and Information Technology, since 2021.

Kaitong Zheng obtained the B.S. degree in marine technology from Xiamen University (XMU), Xiamen, China, in 2019. He is now pursuing his M.S. degree study in The Institute of Acoustics of the Chinese Academy of Sciences, Beijing, China. His research interests focus on underwater acoustic communications and machine learning.



Xing Fang received the B.S. degree in electrical engineering from the Northwestern Polytechnical University (NPU), Xi'an, China, in 2007 and the Ph.D. degree in computer science from North Carolina A&T State University in 2016.
He has been an Assistant Professor at the School of Information Technology, Illinois State University since 2016. His research interests include deep learning, machine learning, and natural language processing.

Lei Wan (Member, IEEE) received the B.S. degree in electronic information engineering from Tianjin University (TJU), Tianjin, China, in 2006, the M.S. degree in signal and information processing from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2009, and the Ph.D. degree in electrical engineering from the University of Connecticut (UCONN), Storrs, CT, USA, in 2014.
Currently, he is an Associate Professor with the School of Informatics, Xiamen University (XMU), Xiamen, China. His research interests include the algorithm design, system development and performance analysis for underwater acoustic communication systems.
Dr. Wan is the associate editor for the IEEE Open Journal of Communications Society. He has served as a technical reviewer for many journals and conferences, and he received the IEEE Communications Society's Exemplary Reviewer Award for the IEEE Communications Letters, in 2013.

Xiaomei Xu received the B.S., M.S., and Ph.D. degrees in marine physics from Xiamen University (XMU), Xiamen, China, in 1982, 1988, and 2002, respectively. She was a Visiting Scholar with the Department of Electrical and Computer Engineering, Oregon State University, Corvallis, OR, USA (1994–1995).
She visited the Department of Electrical and Computer Engineering, University of Connecticut (UCONN), Storrs, CT, USA, as a Senior Visiting Scholar in 2012. She is now a Full Professor with the Department of Applied Marine Physics and Engineering, XMU. Her research interests lie in the fields of marine acoustics, underwater acoustic telemetry and remote control, underwater acoustic communication, and signal processing.