Computers and Electrical Engineering 123 (2025) 110211

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A Q-learning-based hierarchical routing protocol in underwater acoustic sensor networks

Amir Masoud Rahmani [a,1], Jawad Tanveer [b,1], Abdulmohsen Mutairi [c], May Altulyan [d], Entesar Gemeay [e], Mahfooz Alam [f], Mohammad Sadegh Yousefpoor [g], Efat Yousefpoor [g], Mehdi Hosseinzadeh [h,i,*]

[a] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan
[b] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea
[c] College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[d] Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al Kharj, Saudi Arabia
[e] Department of Computer Engineering, Computer and Information Technology College, Taif University, Taif, Saudi Arabia
[f] Department of Mathematics and Statistics, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
[g] Center of Research and Strategic Studies, Lebanese French University, Kurdistan Region, Iraq
[h] School of Computer Science, Duy Tan University, Da Nang, Viet Nam
[i] Jadara Research Center, Jadara University, Irbid 21110, Jordan

ARTICLE INFO

Keywords:
Underwater acoustic sensor networks (UASNs)
Acoustic communication
Reinforcement learning (RL)
Artificial intelligence (AI)
Decision-making systems

ABSTRACT

In order to guarantee a reliable data forwarding process, underwater acoustic sensor networks (UASNs), which are widely used in water environments like oceans and seas, need efficient routing protocols. Because sensor nodes are expensive to deploy in underwater environments and have limited energy capacities, energy optimization is a significant and practical issue, particularly for extending network lifetime. Today, various energy-efficient routing strategies have been suggested by combining opportunistic routing (OR) and reinforcement learning (RL). However, this subject deals still with different challenges. This paper introduces a Q-learning-based hierarchical routing protocol (QHRP) in UASNs. This approach builds a Q-learning-based routing tree, which contains a state set filtered by a two-step filtering process. It effectively increases the convergence speed of the Q-learning algorithm and lowers delay due to the tree construction process. In QHRP, the reward function considers network conditions and is obtained based on four metrics, namely remaining energy, strategic depth, the size of the state set, and successful transmission probability. Moreover, QHRP solves the void area problem in the routing tree by redefining the set of states and reward function. To evaluate QHRP compared to the three routing methods, namely RLOR, EE-DBR, and MURAO, various experiments are performed in terms of packet delivery rate (PDR), end-to-end delay (EED), data integrity, consumed energy, and the number of hops in the forwarding routes. These results show that

* Corresponding author at: School of Computer Science, Duy Tan University, Da Nang, Viet Nam.
E-mail addresses: rahmania@yuntech.edu.tw (A.M. Rahmani), jawadtanveer@sejong.ac.kr (J. Tanveer), mutairi@ksu.edu.sa (A. Mutairi), m.altulayan@psau.edu.sa (M. Altulyan), esgemeay@tu.edu.sa (E. Gemeay), mahfooz.alam@vupune.ac.in (M. Alam), mohammad.sadegh@lfu.edu.krd (M.S. Yousefpoor), efat.yousefpoor@lfu.edu.krd (E. Yousefpoor), mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compeleceng.2025.110211
Received 14 October 2024; Received in revised form 13 January 2025; Accepted 18 February 2025
Available online 7 March 2025

QHRP improves PDR, delay, data integrity, energy consumption, and the number of hops by 9.068%, 9.03%, 9.84%, 15.61%, and 10.31%, respectively.

1. Introduction

Nowadays, underwater acoustic sensor networks (UASNs) have become a hot research issue for many researchers due to the focus on oceans and their information. UASNs are three-dimensional networks that effectively identify and monitor underwater environments [1,2]. They are applied in port monitoring and security, environmental monitoring, prevention of natural disasters, and navigation assistance. In recent years, these networks have attracted the attention of many governments, industries, and universities [3,4]. Moreover, due to the increasing growth of the Internet of underwater things (IoUT), UASNs have become complex and multi-hop networks consisting of highly dense sensors equipped with data sensing, information processing, and acoustic communication capabilities [5,6]. To guarantee efficient data transmission in UASNs, underwater routing schemes have become a key research subject. Due to the fundamental differences in the underwater environment and the specific features of UASNs, the data forwarding process in these networks is much more complex than wireless sensor networks (WSNs) because UASNs deal with the energy limitations of underwater nodes [7,8]. Excessive use of sensor nodes to forward data to the sink node reduces energy efficiency in UASNs and leads to their early death. In addition, due to the high cost of deploying sensors in the underwater environment, they can be rarely replaced in UASNs. As a result, the lack of sufficient sensors to cover the target area causes void areas and decreases the packet delivery rate [9,10]. Furthermore, underwater sensor nodes may move due to the waves and water currents. This leads to changes in the network topology. Therefore, the forwarding routes will be invalid and the data forwarding process will be unreliable due to the disconnection of communication links. Hence, existing routing protocols in WSNs cannot be used directly in UASNs [11,12].

Routing is to select a path from the set of forwarding routes based on the desired criteria to improve the performance of UASNs. Due to problems such as high bit error rate (BER), high latency, limited bandwidth, and energy restriction in UASNs, designing a reliable, robust, energy-efficient, low-delay routing protocol improves the adaptability to UASNs [13,14]. By combining opportunistic routing (OR) and reinforcement learning (RL), several forwarding routes can be provided for transferring data packets and guaranteeing the reliable data forwarding process in UASNs [15,16]. Reinforcement learning enables UASNs to interact with the underwater environment and select the best route to the destination. RL provides a decision-making framework through a continuous interaction with the underwater environment [17,18]. Note that the information collected by underwater sensor nodes is local and limited in distributed networks. Thus, RL assists underwater sensor nodes to learn from the environment and increase their general interests to decide on the optimal route. Q-learning is a well-known and popular RL strategy and can solve routing challenges in UASNs [19,20].

This paper presents a Q-learning-based hierarchical routing scheme (QHRP) in UASNs. The scheme builds a Q-learning-based routing tree, which contains a state set filtered using two filtering steps. It effectively increases the convergence rate of the Q-learning algorithm and manages delay in the routing tree construction process. This routing tree seeks to balance the energy consumption of sensor nodes and increase the stability of different routes to the sink node. In general, the main contributions of QHRP are as follows.

- In QHRP, an information exchange process and the format of the exchanged packets, namely hello packet, data packet, and ACK packet are demonstrated. Hello packets are exchanged to collect information about neighboring nodes around the desired node. Furthermore, the data packet stores the collected data and carries this data from the source node to the destination and the ACK packet is sent to the source node to confirm the data received by the destination.
- In QHRP, a decentralized Q-learning-based routing tree construction process is created to exchange information between sensor nodes and the sink node. This process defines a state set filtered by a two-step filtering so that each node can select a set of possible parent nodes in the routing tree. The purpose of the first filtering is to make free-loop paths to form a tree topology between sensor nodes. The second filtering defines a new concept called strategic depth that combines the depth of sensor nodes and the number of hops to the sink node. According to this new concept, each node prefers to filter neighboring nodes that have more strategic depth than it because these nodes are farther away from the sink node and have more hops to the sink.
- In QHRP, the reward function is obtained from four parameters, namely remaining energy, strategic depth, the size of the state set, and successful transmission probability, to choose the best parent node in the routing tree.
- QHRP solves the void area problem in the tree construction process. The recovery process seeks to modify the state set related to this void node to find a parent node in the routing tree. At this step, the reward function is redefined based on remaining energy, strategic depth, hop counts, and successful transmission probability.

In the following, the organization of the present paper includes different sections: Section 2 illustrates the research works related to routing in UASNs. Section 3 demonstrates system settings in QHRP in three sub-sections, network model, acoustic propagation model, and energy model. Section 4 introduces a Q-learning-based hierarchical routing protocol in UASNs. Section 5 presents simulation settings, evaluation parameters, and simulation results. In Section 6, the conclusions obtained from this paper are summarized.

2. Related works

In [21], a centralized clustering method (CCCS) is provided in UASNs to optimize energy consumption in the network. CCCS benefits from an adaptive strategy and considers network density to find intra-cluster controllers and choose relay nodes and their associated clusters to balance energy consumption when finding intra- and inter-cluster routes. In CCCS, the sink node controls inert-cluster communications throughout the network, and inter-cluster controllers are responsible for determining the role of nodes (cluster members and cluster heads) and keeping intra- and inter-cluster communications. In addition, the adaptive clustering algorithm considers network density and chooses CHs according to energy consumption. It filters relay nodes to distribute energy consumption uniformly. The results obtained from the simulation process show that CCCS efficiently manages energy consumption and extends network lifespan.

In [22], a Q-learning-based hierarchical routing technique along with an unequal clustering strategy called QHUC are introduced in UASNs. It focuses on the selection of optimal paths and expands the network lifespan. Initially, QHUC creates a hierarchical topology to organize sensor nodes in the network. QHUC mixes an unequal clustering strategy and the Q-learning algorithm to build this hierarchical topology to distribute the energy consumption uniformly in the network. QHUC determines CHs and the next forwarders using a Q-learning-based greedy approach. In addition, Q-values can be calculated without imposing additional costs by combining the Q-learning algorithm and clustering. The simulation results show that QHUC works successfully compared to other routing methods.

In [23], the efficient routing approach called DROR is presented to guarantee the reliable data forwarding process and solve the void region problem in UWSNs. DROR utilizes a Q-learning-based opportunistic routing strategy, which takes into account energy restrictions and the specific features of the underwater environment. It merges Q-learning and opportunistic routing to ensure rapid data forwarding and energy efficiency in the network. In this approach, a recovery technique is designed to address the void region issue. According to this technique, void nodes are modified and do not disturb the data forwarding process. In addition, DROR proposes a dynamic Q-learning strategy to ensure that data packets are effectively directed toward the sink node. Simulation results report the successful performance of DROR.

In [24], a fuzzy layered routing approach (FCLR) is suggested in UASNs. FCLR applies underwater nodes, which continuously learn their layer and get information about their neighbors by sharing control packets. To choose the optimal forwarder, FCLR employs a fuzzy control system (FCS) with two input variables, namely remaining energy and the density of nodes. Moreover, FCLR presents an effective solution for the void region issue so that only neighboring nodes in the lower layers can be nominated as forwarders. The simulation results report the superiority of FCLR over other protocols. In addition, the authors have tested FCLR in the largest saltwater lake in China, Qinghai, to evaluate its performance in a real environment. The results obtained from this experiment show the good performance of FCLR.

In [25], a Q-learning-based opportunistic routing approach called RLOR is defined in UASNs. This scheme merges opportunistic routing and Q-learning and is a decentralized routing method, which constantly considers the environmental conditions of underwater nodes to find the most appropriate forwarders. A recovery strategy has been designed in RLOR so that data packets get away from the void region and move toward the destination. This strategy increases the number of packets delivered to the destination. The simulation results clearly show the proper performance of RLOR compared to other routing approaches.

In [26], an energy-efficient and low delay depth-based routing scheme (EE-DBR) is introduced in UASNs. It is inspired by DBR, which is a depth-based routing technique that does not pay attention to energy efficiency and propagation delay. It is not enough to rely only on the depth information of underwater nodes to limit the data forwarding process in a particular area because data packets can be sent to the sink node through various paths. This wastes energy and increases delay in the routing process. As a result, EE-DBR utilizes the underwater time of arrival (ToA) ranging strategy for dealing with this challenge. This approach tries to manage energy consumption in some blind areas by exploiting an efficient mechanism. Additionally, EE-DBR delivers data packets to the sink node through an optimal path. The evaluation results illustrate the good performance of EE-DBR.

In [27], an acoustic-optical structure and a Q-learning-based multi-level routing scheme (MURAO) are proposed in UWSNs. This approach physically organizes underwater nodes into several groups and logically divides them into two layers. The upper layer includes underwater nodes (group leaders), which control the routing process in the lower layer so that the members belonging to the lower layer perform the data transfer procedure successfully. These leaders at the upper layer monitor the border of UWSN; all groups perform the learning process simultaneously. Additionally, MURO optimizes the routing strategy compared to the flat Q-learning-based routing approaches. The evaluation results illustrate that MURAO works well in networks with high topological changes and improves PDR and latency compared to the flat Q-learning-based routing approaches.

In [28], a multi-agent Q-learning-based routing approach called MC-DBR is designed in UWSNs. This approach seeks to lower energy consumption and latency, and increase reliability in the data forwarding procedure. MC-DBR utilizes adaptive modulation and coding in the depth-based routing process. In MC-DBR, underwater nodes share hello packets with each other to be aware of adjacent channels. Then, each node refreshes its Q-table using a RL-based modulation and coding algorithm. Additionally, this algorithm applies various metrics such as energy consumption, delay, modulation and coding methods, and packet collisions to optimize the performance of UWSN. The evaluations indicate the proper performance of MC-DBR compared to other routing approaches.

Table 1 outlines the most significant benefits and drawbacks of the related works. Accordingly, many routing protocols such as [22,23,25,27,28], and QHRP use reinforcement learning techniques, especially Q-learning, to improve the data transmission process and guarantee the reliable data forwarding process in UASNs. Reinforcement learning enables UASNs to interact with the underwater environment and select the best route to the destination. On the other hand, the void area problem and low energy efficiency in UASNs are important challenges that must be solved. All research works reviewed in this section consider energy

efficiency. Especially, hierarchical routing protocols such as [21,22,27], and QHRP can efficiently manage the energy consumption of nodes in the network. Furthermore, because sensor nodes in the ocean are expensive, there are not enough nodes to cover the network environment, and the deployment of nodes is excessively sparse, which leads to routing voids. However, most existing routing protocols such as [21,22,24,26,27], and [28] often ignore the void problem. Whereas, [23,25], and QHRP tries to solve this important routing issue. QHRP is a Q-learning-based hierarchical routing scheme, which builds a Q-learning-based routing tree, which contains a state set filtered using two filtering steps. It effectively solves the void area problem in the routing tree construction process. This routing tree seeks to balance the energy consumption of sensor nodes and increase the stability of different routes to the sink node.

3. System settings

Here, system settings related to QHRP are accurately stated in three subsections, namely network model, acoustic propagation model, and energy model.

3.1. Network model

Fig. 1 illustrates a simple schematic of the network model in QHRP. This network consists of a sink node and n acoustic sensor nodes, which are randomly distributed in a three-dimensional area. The set of all nodes $N = \{an_1, an_2, \dots, an_i, \dots, an_n\}$ is expressed in the network, where $|N| = n$. an_i is a symbol of ith acoustic sensor node in UASN. In this network model, the sink node is floated on the water surface and has high processing power and a stable energy source. Each sensor node such as an_i is connected to the sink through a single- or multi-hop path to dispatch its collected data to the sink. A specific identification number such as ID_i is assigned to each sensor node, like an_i. In addition, the equipment installed on each sensor node is the hydraulic pressure gauge for measuring the depth of sensor nodes, an acoustic modem to connect to other nodes, and different sensors for collecting (sensing) underwater data. Furthermore, the sink node consists of the acoustic modem and the radio frequency modem (RF), which are used to connect with acoustic sensor nodes and the seaside, respectively. Due to the water currents, the acoustic sensor nodes may move slowly. This movement can make changes to the network topology and cut off communication links between the nodes. In summary, the network model in QHRP contains the following assumptions:

- Acoustic sensor nodes are homogeneous. This means that they have the same battery capacity, processing power, communication radius, and hardware specifications.
- Acoustic sensor nodes have a limited energy capacity because it is very difficult to recharge or replace their battery in the underwater environment.
- The sink node has a stable energy source and does not deal with energy restrictions.
- The position of the sink is unchanged on the water surface.
- The depth of the sink is equal to zero, while the sensor nodes' depth varies and is larger than zero.
- Communication between sensor nodes is provided by acoustic signals. However, communication with the seaside is provided by the radio frequency modem.

3.2. Acoustic propagation model

QHRP utilizes the Thorp-based acoustic propagation model [29] to calculate the loss of acoustic propagation. Accordingly, Eq. (1) is employed to evaluate the path loss due to underwater acoustic signals with the signal frequency f and the propagation distance d_{ij} (i.e. the distance between the sender node (an_i) and the recipient node (an_j).

$$A\left(d_{ij}, f\right) = {d_{ij}}^{k} \alpha(f)^{d_{ij}} \tag{1}$$

Here, k and $\alpha(f)$ are the expansion coefficient and the acoustic absorption coefficient, respectively. Note that $1 \le k \le 2$ is used to determine the emission geometry so that $k = 1$ means the cylindrical dissemination geometry and $k = 2$ indicates spherical emission geometry. Eq. (2) also obtains the acoustic absorption coefficient according to the Thorp equation.

$$10\log\alpha(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\times10^{-4}f^2 + 0.003 \tag{2}$$

Then, Eq. (3) is used to estimate the signal-to-noise ratio (SNR).

$$SNR\left(d_{ij}, f\right) = \frac{E_b / A\left(d_{ij}, f\right)}{N_0} = \frac{E_b}{N_0 {d_{ij}}^{k} \alpha(f)^{d_{ij}}} \tag{3}$$

where d_{ij}, E_b, and N_0 indicates the propagation distance between an_i and an_j, the energy consumed to transmit a bit of data, and the noise power spectral density under the additive White Gaussian noise (AWGN), respectively.

In this paper, QHRP estimates the bit error rate by utilizing the most widely used modulation technology, namely the binary phase shift keying (BPSK), for transmitting acoustic signals, as expressed in Eq. (4) [30].

$$p_e\left(d_{ij}\right) = \frac{1}{2}\left(1 - \sqrt{\frac{SNR\left(d_{ij}, f\right)}{1 + SNR\left(d_{ij}, f\right)}}\right) \tag{4}$$

Table 1
Benefits and drawbacks of various methods.

Approach	Publication year	Used technique	Centralized or distributed	Void area problem	Topology structure	Energy efficiency	Requirement	Benefits	Limitations
CCCS [21]	2023	Centralized clustering scheme and Dijkstra routing algorithm	Centralized	No	Cluster-based topology	Yes	Location information	Balancing energy consumption between sensor nodes, high scalability, and high data reliability	High routing overhead, low adaptability to the UASN environment
QHUC [22]	2023	Q-learning-based routing technique and unequal clustering strategy	Distributed	No	Cluster-based topology	Yes	Distance information based on the received signal strength (RSS) of the received packets	Balancing energy consumption between sensor nodes, high scalability, high data reliability, and adaptability to the UASN environment	Low convergence rate due to the large size of the state space in the Q-learning algorithm
DROR [23]	2023	Q-learning-based and opportunistic routing (OR)	Distributed	Yes	Flat network topology	Yes	Depth information	Balancing energy consumption between sensor nodes, high data reliability, adaptability to the UWSN Environment	Low scalability, low convergence rate due to the large size of the state set in the Q-learning algorithm
FCLR [24]	2022	Fuzzy control-based layering routing scheme	Distributed	No	Flat network topology	Yes	–	Balancing energy consumption between sensor nodes, adaptability to the UWSN Environment	Low scalability
RLOR [25]	2021	Q-learning-based and opportunistic routing (OR)	Distributed	Yes	Flat network topology	Yes	Depth information	Balancing energy consumption between sensor nodes, high data reliability, adaptability to the UWSN Environment	Low scalability
EE-DBR [26]	2015	Depth-based routing along with the time of arrival (ToA) technique	Distributed	No	Flat network topology	Yes	Depth information	Balancing energy consumption between sensor nodes	Low adaptability to the UWSN environment, low scalability, and low data reliability
MURAO [27]	2012	Multi-level Q-learning based routing scheme	Distributed	No	Cluster-based topology	Yes	–	High scalability, adaptability to the UWSN environment	Not paying residual energy of nodes in the network
MC-DBR [28]	2022	Multi-agent reinforcement learning-based modulation and coding method (MARL-MC) algorithm	Distributed	No	Flat network topology	Yes	Depth information	Balancing energy consumption between sensor nodes and high data reliability	Low scalability, low convergence speed due to the large size of the state space, and low adaptability to the UWSN environment

(*continued on next page*)

As a result, Eq. (5) can calculate the successful transfer probability of a m-bit packet.

$$p\left(d_{ij}, m\right) = \left[1 - p_e\left(d_{ij}\right)\right]^m \tag{5}$$

Table 1 (*continued*).

Approach	Publication year	Used technique	Centralized or distributed	Void area problem	Topology structure	Energy efficiency	Requirement	Benefits	Limitations
QHRP	–	Q-learning-based hierarchical routing scheme	Distributed	Yes	Tree-based network topology	Yes	Depth information	Balancing energy consumption between sensor nodes, high scalability, high data reliability, and adaptability to the UASN environment	–

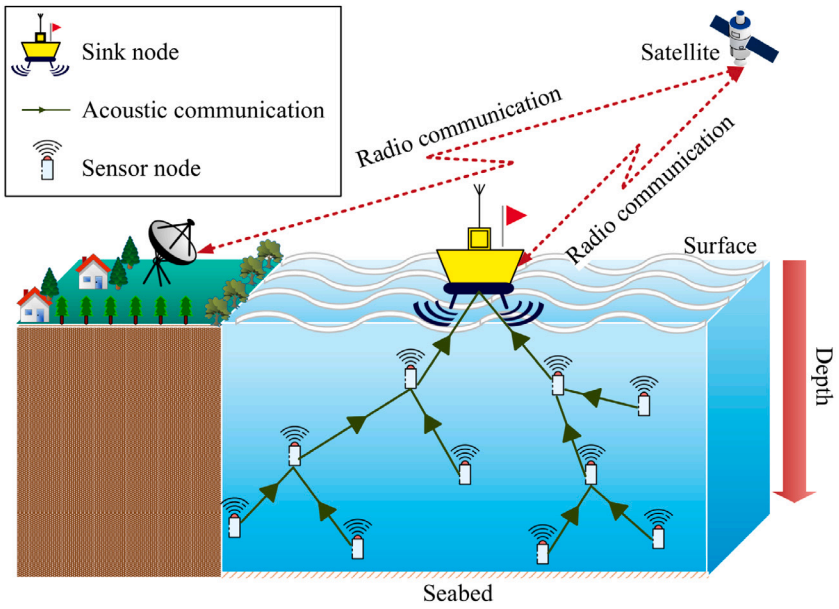

Fig. 1. Network model in QHRP.

3.3. Energy model

In underwater acoustic sensor networks, sensor nodes often consume their energy for sending or receiving data packets. To calculate their energy consumption in the data transfer procedure, QHRP uses Eq. (6) to obtain the energy consumed by the sender node (an_i) to send a m-bit packet to the receiver node (an_j).

$$E_{Tx}\left(m, d_{ij}\right) = mP_0 A\left(d_{ij}, f\right) = mP_0 {d_{ij}}^k \alpha(f)^{d_{ij}} \quad \left(E_{Tx} < E_{remain}\right) \tag{6}$$

here P_0 is the minimum energy required by an_i for sending its data. E_{remain} indicates the residual energy of an_i and shows the upper border related to E_{Tx}. Moreover, Eq. (7) calculates the energy required by an_j to receive the m-bit packet.

$$E_{Rx}(m) = mP_r \tag{7}$$

so that P_r means the receipt coefficient that is defined based on the desired device.

4. Proposed scheme

Here presents the Q-learning-based hierarchical routing protocol (QHRP) in UASNs accurately. In this approach, the hierarchical routing technique constructs a Q-learning-based routing tree in which the state set is a subset of neighboring nodes, selected using two filtering steps. This effectively accelerates the convergence speed of the Q-learning algorithm and lowers the delay caused by the tree construction procedure. This routing tree seeks to balance the energy consumed by sensor nodes and increase the stability of the paths toward the sink. Generally, QHRP includes three steps, namely information exchange, Q-learning-based tree creation, and recovery process. Fig. 2 shows a schematic design of the proposed approach. Table 2 illustrates the most important notations used in this paper.

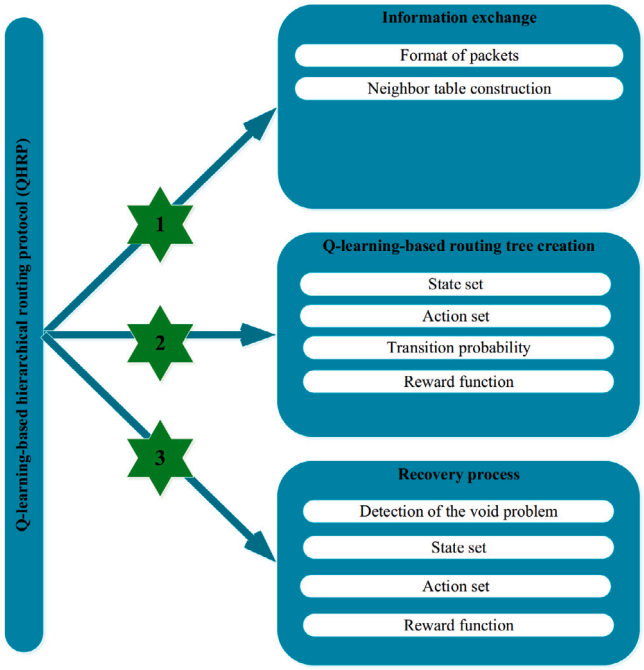

Fig. 2. A schematic design of QHRP.

Table 2
Most important notations used in this paper.

Notation	Description
N	The set of all nodes in the network
an_i	A symbol of ith acoustic sensor node
n	The number of all nodes in the network
ID_i	The identification number assigned to an_i
f	Signal frequency related to underwater acoustic signals
d_{ij}	Distance between an_i and an_j
$SNR\left(d_{ij}, f\right)$	Signal-to-noise ratio (SNR)
E_b	Energy consumed to transmit a bit of data
N_0	Noise power spectral density under the additive White Gaussian noise (AWGN)
$p\left(d_{ij}, m\right)$	Successful transfer probability of a m-bit packet
E_{Tx}	Energy consumed by the sender node
E_{Rx}	Energy consumed by the receiver node
an_s	Source node
an_d	Destination node
h_i	The number of hops from an_i to the sink
NT_i	Neighboring table related to an_i
E_i	Remaining energy of an_i
d_i	Depth information of an_i
S_i	State set related to an_i in the routing tree
S	The set of states in the Q-learning algorithm
A	The set of actions in the Q-learning algorithm
P	Transition probability of states in the Q-learning algorithm
R	Reward matrix in the Q-learning algorithm
F_i^1	First filtering set related to S_i
F_i^2	Second filtering set related to S_i
SD_{ij}	Strategic depth of an_j with regard to an_i

4.1. Information exchange

In QHRP, sensor nodes need to exchange information with other nodes in two modes. The first mode is when sensor nodes do not have access to the updated information about their neighboring nodes in UASNs, their neighboring table is empty, or some fields of this table have expired. In this case, sensor nodes apply hello packets, record their private information in these packets, and share them with their neighboring nodes in the network. The second mode occurs when sensor nodes collect their data and intend to transmit them to the sink. In this case, sensor nodes apply two packets, namely data packet and ACK packet, so that the first holds

the collected data to transmit it to the destination, and the second is returned to the source to confirm data delivered to the receiver. In UASNs, in the data transmission process, the source node (an_s) forwards data packets towards the destination node (an_d). To detect the success of this process, a popular and well-known solution is to send ACK packets back to the previous-hop node upon receiving a packet. Nevertheless, this simple approach needs a lot of network bandwidth and results in significant communication overheads. When a node transfers a data packet, it temporarily keeps this packet in its memory and does not delete its buffer. Then, if the next-hop node (except the destination node) correctly gets this packet, it forwards this packet to the next hop, and the previous-hop node considers the returned packet as an acknowledgment. Next, the previous-hop node deletes this packet from its memory upon getting the ACK packet. Fig. 3-(a) displays the successful data transmission from an_s to an_d. Nevertheless, data packets may not be delivered to the destination due to noisy channels, collisions, and others. If the data transmission is not successful, the previous-hop node does not delete this packet from its memory and transfers it to the next-hop node again. Fig. 3-(b) displays this process. Note that the number of retransmissions must not exceed a predefined threshold. Otherwise, the data transmission process fails. Generally, the information exchange process is dependent on three types of packets, namely hello, data, and ACK whose formats are illustrated in Fig. 4. The most important fields of these packets are demonstrated as follows.

- **Packet type:** This field includes three values to distinguish hello, data, and ACK packets. Accordingly, in hello, data, and ACK packets, this field is adjusted on one, two, and three, respectively. This field will never change during the packet lifetime.
- **Packet ID:** The source node assigns a unique ID to each packet so that duplicated packets are distinguishable. This field will be constant during the packet lifetime.
- **Source ID:** The source node inserts its ID into this field. The amount of this field will remain unchanged during the packet lifetime.
- **Destination ID:** The source node registers the identifier of the destination in this field. The amount of this field is always constant.
- **Q-value:** This field is updated in each hop and the maximum Q-value related to the forwarding node is inserted in this field.
- **V-value:** This field is refreshed in each hop, and the V-value related to the forwarding node is recorded in this field.
- **Residual energy:** This field is updated in each hop and includes the remaining energy of the transmitter node.
- **Depth:** This field is updated in each hop and maintains the depth information of the transmitter node.
- **State set:** This field is refreshed in each hop and stores the state set of the transmitter node.
- **Hop count:** This field is refreshed in each step and specifies the number of hops from the transmitter node to the sink. When launching the network operation, the sink node prepares a hello packet and the hop count field sets to zero, i.e. $h_{Sink} = 0$. Then, it disseminates this packet to its adjacent nodes in UASN. When the distance between a node such as an_i and the sink is one hop, it gets this packet directly from the sink, adds one unit to the hop count field (i.e. $h_i = h_{Sink} + 1$), and disseminates this packet again. After receiving this packet by another node, like an_j, this node examines whether it has previously received this hello message. If an_j gets this packet for the first time, then one unit is added to the hop count field, i.e. $h_j = h_i + 1$. Then, an_j disseminates this packet to its adjacent nodes. If an_j has already received the hello message from other nodes, then it compares its hop counts with the number of hops inserted into the new hello packet. If $h_j > h_i + 1$, it will update its hop counts i.e. $h_j = h_i + 1$. Otherwise, this parameter will remain unchanged. Note that the sink node sends the hello packet only once when launching the network so that each node is aware of its number of hops to the sink.
- **The previous-hop node:** This field is updated in each hop and keeps the information about the previous-hop node.
- **The next-hop node:** This field is refreshed in each hop and keeps the information about the next-hop node.

In general, the structure of each packet consists of two parts, namely the packet header and the payload so that the first includes two subsections, the packet information and the routing information, and the second contains sensed data. Payload is specialized to data packets and other packets do not need this field. In QHRP, any sensor node can hear the packets disseminated around it and refresh its neighbor table according to the information stored in the packet header. Then, if this packet has another destination, this node deletes that packet. Otherwise, it processes other stored information. Hello packets are periodically broadcast in the network so that the neighbor table always is fresh. The periodicity of this process is rooted in the movement of sensor nodes due to water currents, which make topological changes in UASN and affect connections between sensor nodes. The period of each hello packet is determined based on network conditions. Note that a short period improves the adaptability of QHRP to the dynamic environment of UASNs and guarantees that the neighbor table is timely updated, but this short time increases energy consumption and overhead in the network.

After receiving each packet, the neighbor table is updated according to the following steps. Table 3 shows the structure of the neighbor table.

- **Step 1:** When a sensor node such as an_i looks a packet from another sensor node such as an_j, first an_i examines the packet ID to guarantee that this packet is fresh.
- **Step 2:** If this packet is duplicate, an_i will remove this packet. Otherwise, it will process information inserted into this packet.
- **Step 3:** an_i checks whether personal information of an_j is already recorded in this table NT_i.
- **Step 4:** If NT_i includes the identifier of an_j, then its private information will be refreshed in accordance with the information available in the packet header.
- **Step 5:** If NT_i does not include the identifier of an_j, then an_i considers a new entry specialized for an_j in NT_i to records the information available in the received packet header.

Table 3
The structure of the neighbor table (NT_i).

Identifier	Remaining energy	Depth information	Hop count	State set	Q-value	V-value	Validation time
an_j	E_j	d_j	h_j	S_j	Q_j	V_j	T_{Valid}

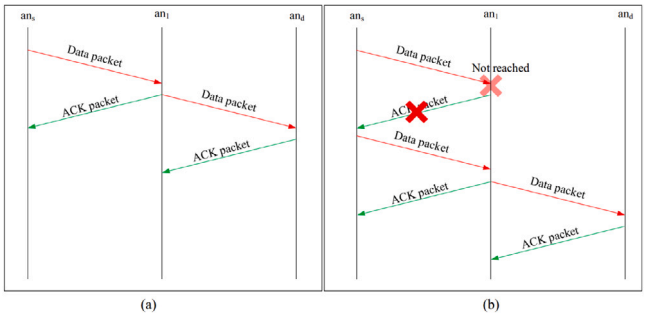

Fig. 3. Information exchange (a) Successful data transmission, (b) Unsuccessful data transmission.

- **Step 6:** If NT_i includes the identifier of an_j and an_i does not receive any new packet from this node, then an_i will remove the related entry from NT_i as soon as the information about an_j is expired and invalid.

Algorithm 1 shows the pseudo-code related to the information exchange process. Time complexity of this algorithm is $O(n_i^2)$, where n_i indicates the number of neighbors of an_i.

Algorithm 1 Information exchange

Input: an_i: i-th acoustic sensor node belonging to $N = \{an_1, an_2, \ldots, an_i, \ldots, an_n\}$ in UASN.
n: All number of acoustic sensor nodes.
T_{Net}: Simulation duration.
t: A timer, which measures the consumed time so far.
Output: NT_i: Neighbor table saved in an_i
Begin
Sink: Turn on the network timer i.e. $t = 0$;
Sink: Determine the hello period proportional to network conditions;
repeat
  if the period of hello message reaches **then**
    an_i: Create a hello packet according to Fig. 4 (a);
    an_i: Broadcast its hello packet to the surrounding acoustic sensor nodes in UASN;
  end if
  while (an_i acquires a fresh packet from its neighboring node (like an_j) **do**
    an_i: Check the packet ID;
    if this packet is duplicated **then**
      an_i: Delete this packet;
    else
      an_i: Process this packet to extract its header information;
      an_i: Search NT_i to find the identification number of an_j;
      if NT_i contains the identification number of an_j **then**
        an_i: Update the personal information of an_j recorded in NT_i based on the header information of the packet;
      else
        an_i: Append the entry related to an_j to NT_i;
        an_i: Enter the personal information of an_j to its entry in NT_i;
      end if
    end if
  end while
  for each sensor node such as an_j, which is belonging to NT_i **do**
    if an_i does not acquire a fresh packet from an_j **and** the personal information of an_j in NT_i is out-of-date **then**
      an_i: Obliterate the entry related to an_j from NT_i;
    end if
  end for
  $t = t + 1$;
until $t \leq T_{Net}$
End

4.2. Q-learning-based tree creation process

Here, the Q-learning-based tree creation process is explained. It is a decentralized routing tree and executes the data exchange process between sensor nodes and the sink node. This employs the Q-learning algorithm, which is a common and value-based RL strategy and can solve many routing issues such as [31,32], and [33] in UASNs. This Q-learning-based model can be formulated

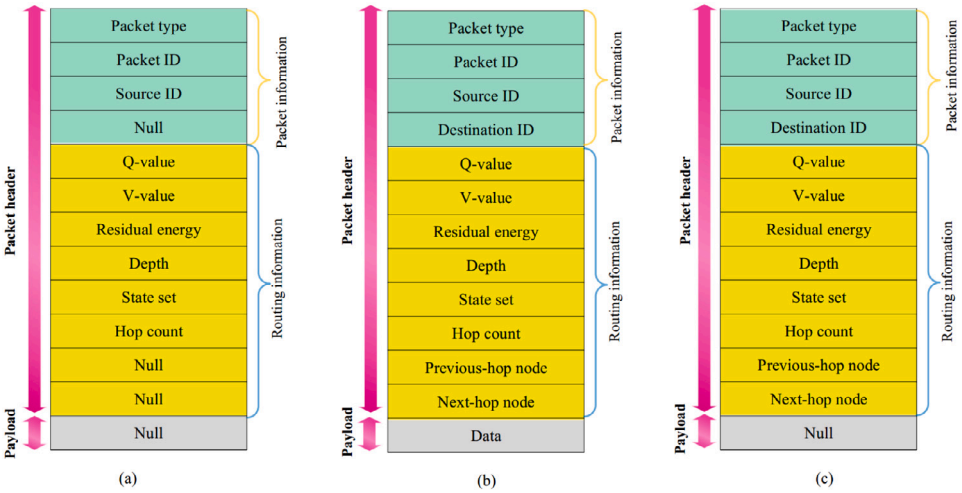

Fig. 4. The packet structure (a) Hello packet, (b) Data packet, and (c) ACK packet.

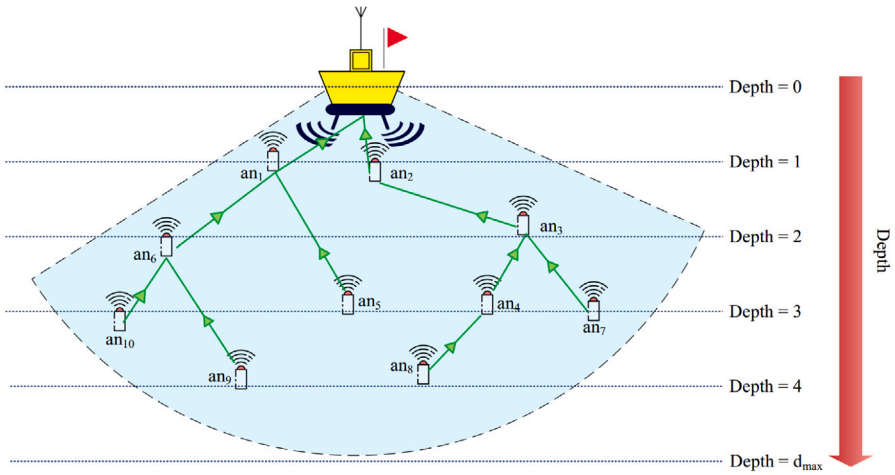

Fig. 5. Evaluation of sensor nodes in terms of depth information.

as a Markov decision process with a quadruple $\langle S, A, P, R\rangle$, where S, A, P, and R mean the set of states, the set of actions, the transition probability of states, and rewards, respectively. In QHRP, the UASN environment and sensor nodes are corresponding to the learning environment and the agent, respectively. Moreover, S, A, P, and R are defined as follows.

***State set* (S).** This set is defined in Eq. (8).

$$S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\} \tag{8}$$

Here, s_i corresponds the state of an_i in the routing tree, and n is the total number of sensor nodes in UASN. In the routing tree, when the agent goes from the current state s_i to the next state s_j, this means that an_i selects an_j as its parent in the routing tree. In QHRP, $S_i \subset S$ indicates the set of states corresponding to each sensor node such as an_i. To determine this state set, an_i considers its neighboring nodes belonging to NT_i such as an_j and performs two filtering steps on this set of neighboring nodes to obtain S_i.

- **First filtering set:** In this step, an_i examines each member such as an_j belonging to NT_i and determines whether an_i belongs to S_j (i.e. the state set related to an_j). The purpose of this filtering is to make free-loop paths to form a tree topology between sensor nodes. Thus, the first filtering set contains a set of nodes that an_i belongs to their state set. This set is defined through Eq. (9).

$$F_i^1 = \{an_j | an_j \in NT_i \text{ and } an_i \in S_j \text{ and } i \neq j\} \tag{9}$$

- **Second filtering set:** In this step, an_i examines each member such as an_j belonging to NT_i in terms of a new concept called the strategic depth (SD_{ij}) to identify sensor nodes that have more strategic depth than an_i (i.e. $SD_{ij} > 0$) and put it inside the

Table 4
Classification of sensor nodes in terms of depth and number of hops.

Sensor node	Classification in terms of depth	Classification in terms of hop counts
an_1	$d_1 = 1$	$h_1 = 1$
an_2	$d_2 = 1$	$h_2 = 1$
an_3	$d_3 = 2$	$h_3 = 2$
an_4	$d_4 = 3$	$h_4 = 3$
an_5	$d_5 = 3$	$h_5 = 2$
an_6	$d_6 = 2$	$h_6 = 3$
an_7	$d_7 = 3$	$h_7 = 4$
an_8	$d_8 = 4$	$h_8 = 3$
an_9	$d_9 = 4$	$h_9 = 4$
an_{10}	$d_{10} = 3$	$h_{10} = 4$

second filtering set. This filtering set is calculated based on Eq. (10).

$$F_i^2 = \left\{ an_j | an_j \in NT_i \ and \ SD_{ij} > 0 \ and \ i \neq j \right\} \tag{10}$$

The strategic depth is a combination of the depth information and the number of hops to the sink. In QHRP, the depth information related to each neighboring node such as an_j is recorded in NT_i. Obviously, an_i prefers to filter neighboring nodes that are more depth than itself because these sensor nodes are farther away from the sink, and experience more delay in the data forwarding process toward the sink. In Fig. 5, sensor nodes are categorized in terms of depth so that the depth of an_1 and an_2 is $Depth = 1$, that of an_3 and an_6 is $Depth = 2$, that of an_4, an_5, an_7, and an_{10} is $Depth = 3$, and that of an_8 and an_9 is $Depth = 4$. Note that, in Fig. 5, purple lines represent different depths of sensor nodes in UASN.

However, it is not enough to filter sensor nodes only based on their depth. In this case, this filtering cannot guarantee that data packets reach the sink faster because the nodes with lower depth do not necessarily have less hops to the sink. For example, in Fig. 6, sensor nodes are classified based on the number of hops to the sink. As shown in Fig. 6, the hop count of an_1 and an_2 is $Hop\ count = 1$, that of an_3 and an_5 is $Hop\ count = 2$, that of an_6, an_8, and an_4 is $Hop\ count = 3$, that of an_7, an_9, and an_{10} is $Hop\ count = 4$. Note that, in Fig. 6, black lines show the number of hops to the sink node.

Information extracted from Figs. 5 and 6 is presented in Table 4. As shown in this table, some nodes such as an_4 and an_5 have the same depth, but are different in terms of the number of hops.

This subject reveals the importance of a new concept called strategic depth because the concept combines depth information and the number of hops to the sink (see Fig. 7). According to this new concept, an_i prefers to filter neighboring nodes that have more strategic depth than it because these sensor nodes are farther away from the sink and have more hops to the sink. As a result, these nodes experience more delay in the data forwarding process. Thus, filtering these sensor nodes based on strategic depth ensures that data packets will reach the sink faster. Hence, sensor nodes with lower depth and smaller hop counts remain in the state set and can play the role of the parent node in the routing tree. The strategic depth of an_j with regard to an_i is calculated using Eq. (11).

$$SD_{ij} = \left(\frac{d_j - d_i}{R} \right) e^{-\left(\frac{1 + \left(\frac{h_i - h_j}{\max_{an_k \in NT_i} \{h_i - h_k\}} \right)}{2} \right)} \tag{11}$$

here d_i, d_j, and R indicate the depth of an_i, that of an_j, and the communication radius of sensor nodes, respectively. Moreover, h_i and h_j demonstrate the number of hops corresponding to an_i and an_j, respectively. In this equation, $e^{-\left(\frac{1 + \left(\frac{h_i - h_j}{\max_{an_k \in NT_i} \{h_i - h_k\}} \right)}{2} \right)}$ acts as an adjustment coefficient. When the number of hops to the sink node is low, the strategic depth must be reduced. The strategic depth helps each sensor node to select a node with the optimal depth and hop counts as its parent in the routing tree.

Now, the union of two filters is calculated through Eq. (12).

$$F_i = F_i^1 \bigcup F_i^2 \tag{12}$$

Then, an_i performs the filtering operation through Eq. (13) to obtain the state set S_i.

$$S_i = NT_i - F_i = \left\{ an_j | an_j \in NT_i \ and \ an_j \notin F_i \right\} \tag{13}$$

Algorithm 2 shows the pseudo-code related to the filtered state set. The time complexity of this algorithm depends on Algorithms 3 and 4. The time complexities of these two algorithms depend on the size of the state space. In the worst case, the size of the state space is n, where n is the number of sensor nodes in the network. Thus, the time complexity of these algorithms is $O(n^2)$.

Algorithm 2 Filtered state set

Input: an_i: i-th acoustic sensor node belonging to $N = \{an_1, an_2, \ldots, an_i, \ldots, an_n\}$.
n: All number of acoustic sensor nodes.
T_{Net}: Simulation duration.
t: A timer, which measures the consumed time so far.
d_i: Depth of an_i
h_i: Number of hops of an_i to the sink node.
E_i: Remaining energy of an_i
SD_{ij}: Strategic depth an_j with regard to an_i
NT_i: Neighbor table related to an_i extracted from Algorithm 1.
Output: S_i: State set of an_i
Begin
for each an_j, which is belonging to NT_i **do**
  if $an_i \in S_j$ **then**
    an_i: Append an_j to the first filter set F_i^1;
  end if
  an_i: Calculate the strategic depth an_j with regard to an_i (i.e. SD_{ij}) based on Eq. (11);
  if $SD_{ij} > 0$ **then**
    an_i: Append an_j to the second filter set F_i^2;
  end if
end for
an_i: Calculate the union of two sets F_i^1 and F_i^2 ($F_i = F_i^1 \bigcup F_i^2$) based on Eq. (12);
an_i: Obtain the state set of an_i from Eq. (13);
if $S_i = \varnothing$ **then**
  an_i: Go to Algorithm 4;
else
  an_i: Go to Algorithm 3;
end if
End

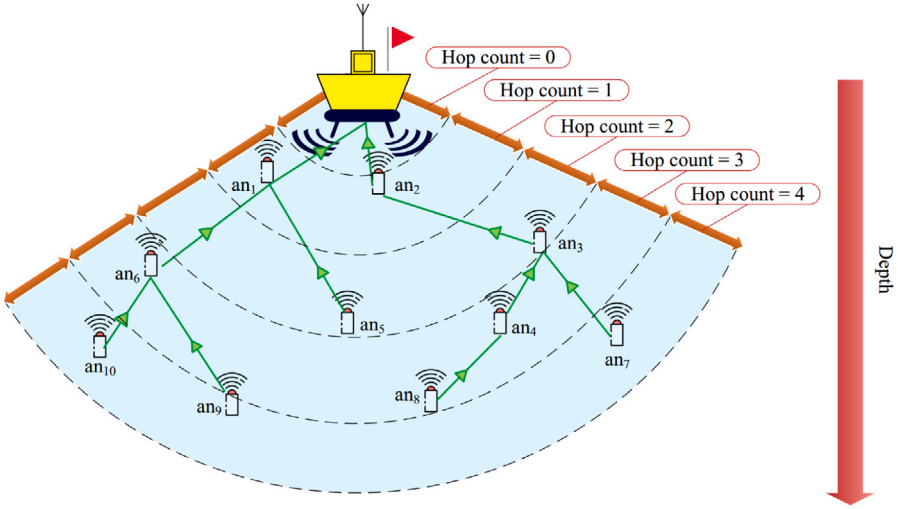

Fig. 6. Classification of sensor nodes in terms of hop counts.

***Action set* (A).** This set is stated in Eq. (14).

$$A = \{a_1, a_2, \ldots, a_i, \ldots, a_n\} \tag{14}$$

Here, a_i indicates the choice of an_i as a parent node in the routing tree.

***Transition probability* (P).** The transition probability set is expressed in Eq. (15).

$$P = \{P\{s_1\}, P\{s_2\}, \ldots, P\{s_i\}, \ldots, P\{s_n\}\} \tag{15}$$

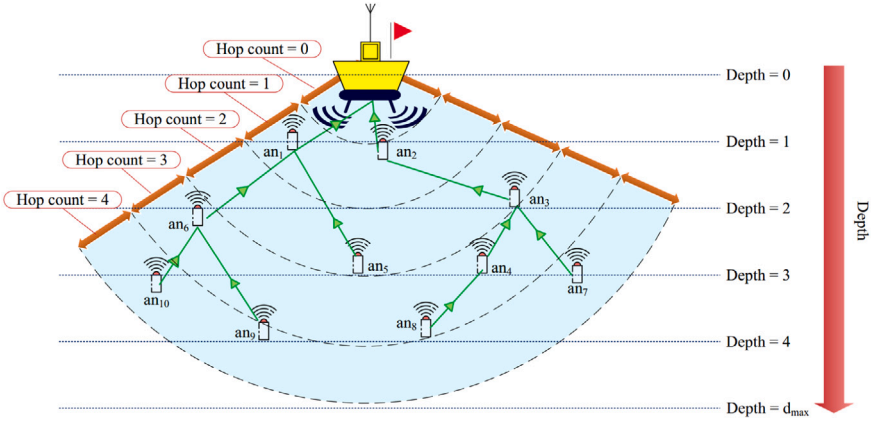

Fig. 7. Strategic depth as a combination of depth information and hop counts.

Here, $P\{s_i\}$ represents a transition probability matrix defined in Eq. (16).

$$P(s_i) = \begin{bmatrix} P_{s_i s_1}^{a_1} & 0 & \cdots & P_{s_i s_i}^{a_1} & \cdots & 0 \\ 0 & P_{s_i s_2}^{a_2} & \cdots & P_{s_i s_i}^{a_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \vdots \\ 0 & 0 & \cdots & P_{s_i s_i}^{a_i} & \cdots & 0 \\ \vdots & \vdots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{s_i s_i}^{a_N} & \cdots & P_{s_i s_N}^{a_N} \end{bmatrix} \tag{16}$$

Here, $P_{s_i s_j}^{a_j}$ means the probability that an_i performs the action a_j to successfully move from the state s_i to the state s_j. In QHRP, this probability is obtained via Eq. (17) [34].

$$P_{s_i s_j}^{a_j} = \frac{R_{s_i s_j}^{a_j}}{\sum_{k \in S_i} R_{s_i s_k}^{a_k}} \tag{17}$$

where, $R_{s_i s_j}^{a_j}$ is the reward value that is obtained after taking the action a_j by an_i and transferring from the current state s_i to the next state s_j. This award is calculated through Eq. (18). Moreover, S_i shows the state set calculated by Eq. (13).

In addition, the unsuccessful transition probability is obtained from Eq. (18).

$$P_{s_i s_i}^{a_j} = 1 - P_{s_i s_j}^{a_j} \left(s_j \neq s_i\right) \tag{18}$$

***Reward* (R).** The set of rewards is defined in Eq. (19).

$$R = \{R(s_1), R(s_2), \ldots, R(s_i), \ldots, R(s_n)\} \tag{19}$$

so that $R(s_i)$ denotes the reward matrix obtained via Eq. (20).

$$R(s_i) = \begin{bmatrix} R_{s_i s_1}^{a_1} & 0 & \cdots & R_{s_i s_i}^{a_1} & \cdots & 0 \\ 0 & R_{s_i s_2}^{a_2} & \cdots & R_{s_i s_i}^{a_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \vdots \\ 0 & 0 & \cdots & R_{s_i s_i}^{a_i} & \cdots & 0 \\ \vdots & \vdots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{s_i s_i}^{a_N} & \cdots & R_{s_i s_N}^{a_N} \end{bmatrix} \tag{20}$$

Here $R_{s_i s_j}^{a_j}$ means an immediate reward for doing the action a_j by an_i and changing from the state s_i to the state s_j. In QHRP, the reward function considers network conditions and is calculated based on remaining energy, strategic depth, the size of the state set and successful transmission rate. The reward value directly affects the selection of parent node in the routing tree. This function

is illustrated in Eq. (21).

$$R_{s_i s_j}^{a_j} = 1 - e^{-\left(\frac{E_j - \min\limits_{an_k \in NT_i}\{E_k\}}{\max\limits_{an_k \in NT_i}\{E_k\} - \min\limits_{an_k \in NT_i}\{E_k\}}\right)\left(\left|SD_{ij}\right|\right)(P(d,l))\left(\frac{\left|S_j\right| - \min\limits_{an_k \in NT_i}\{|S_k|\}}{\max\limits_{an_k \in NT_i}\{|S_k|\} - \min\limits_{an_k \in NT_i}\{|S_k|\}}\right)} \tag{21}$$

where E_j denotes the residual energy of an_j. SD_{ij} is the strategic depth of an_j with regard to an_i. $p(d,l)$ represents the probability that transferring from an_i to an_j successfully. It is obtained from Eq. (5) in Section 3.2. Furthermore, $\left|S_j\right|$ represents the size of the state set related to an_j.

Additionally, Eq. (22) calculates the direct reward ($r_t\left(a_j\right)$) for taking the action a_j and changing from the current state s_i at the moment t.

$$r_t\left(a_j\right) = \sum_{s_j \in S} P_{s_i s_j}^{a_j} R_{s_i s_j}^{a_j} \tag{22}$$

In this Q-learning model, the agent selects an action using the exploration-exploitation greedy scheme according to the transition matrix $P\left(s_i\right)$. In this case, the environment state changes to the corresponding state, and the environment calculates the reward function. Then, the learning agent evaluates the environment based on the calculated reward to complete its knowledge from the environment. Thus, to explore the optimal behavior strategy, the learning agent tries to maximize state–action pairs $Q^{\pi}\left(s_t, a_t\right)$, where t means a specific time slot.

$$Q^{\pi}\left(s_t, a_t\right) = E\left[G_t | S = s_t, A = a_t\right] \tag{23}$$

Here, G_t, calculated in Eq. (24), means the accumulated reward obtained from the action a_t taken by the learning agent under the policy π in the state s_t.

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots = R_t + \gamma G_{t+1} \tag{24}$$

here $\gamma \in (0,1)$ means the discount factor and is responsible for balancing between the current and future rewards. Then, the state–action pairs $Q^{\pi}\left(s_t, a_t\right)$ are calculated through Eq. (25).

$$\begin{aligned} &Q^{\pi}\left(s_t, a_t\right) = E\left[R_t + \gamma G_{t+1} | S = s_t, A = a_t\right] \\ &= R_t + \gamma \textstyle\sum_{s_{t+1} \in S} P_{s_t s_{t+1}}^{a_t} \max\limits_{a} Q^{\pi}\left(s_{t+1}, a\right) \end{aligned} \tag{25}$$

In Q-learning, $Q^{\pi}\left(s_t, a_t\right)$ is obtained from Eq. (26).

$$Q^{\pi}\left(s_t, a_t\right) \leftarrow Q^{\pi}\left(s_t, a_t\right) + \alpha\left[R_t + \gamma \max_{a} Q^{\pi}\left(s_t, a\right) - Q^{\pi}\left(s_t, a_t\right)\right] \tag{26}$$

where $\alpha \in [0,1]$ indicates the learning rate and is usually adjusted on one. Thus,

$$Q^{\pi}\left(s_t, a_t\right) = R_t + \gamma \max_{a} Q^{\pi}\left(s_t, a\right) \tag{27}$$

Algorithm 3 expresses the pseudo-code related to the Q-learning-based tree creation process. The time complexity of this algorithm depends on the size of the state space. In the worst case, the size of the state space is n, where n is the number of sensor nodes in the network. Thus, the complexity of the algorithm is $O(n^2)$.

4.3. Recovery process

When making a routing tree, a problem is that sensor nodes may fall into a void area. The void area problem means that a sensor node follows the tree creation instructions stated in Section 4.2 and cannot choose a parent node in the routing tree. Hence, the data forwarding process from this sensor node and the nodes located in its sub-tree is disrupted. This causes data loss. The void area issue is displayed in Fig. 8. As shown in this figure, an_3 deals with the void area problem because its state set is emptied after performing the two filtering steps on the set of neighboring nodes. This is because any neighboring nodes cannot satisfy the strategic depth condition (i.e. the second filtering). As a result, according to Section 4.2, an_3 cannot find any node as its parent in the routing tree. In addition, an_6, which selects this void node (an_3) as its parent, cannot successfully execute the data transfer process to the sink.

In QHRP, the void area problem is an important challenge. However, the reward function designed in the tree creation process contains the size of the state set ($\left|S_j\right|$) that can effectively solve this problem because nodes with an empty set of states obtain the minimum reward (i.e. zero) and the probability of their choice as a parent node in the tree is very weak or even close to zero. However, this problem must be solved completely by designing a powerful recovery process to increase data reliability. The recovery process tries to modify the state set related to the void node to select a suitable parent node in the routing tree. The recovery process in QHRP includes the following steps:

- **Step (1) Void area detection:** In this step, the void node disseminates a hello packet for its neighbors in UASN. In this hello packet, the void node adjusts the state set on null to informing its neighbors of the void area. Then, the neighboring nodes that are informed of the void area issue remove the void node from their state set. See Fig. 9.

Algorithm 3 Q-learning-based tree creation

Input: an_i: i-th acoustic sensor node belonging to $N = \{an_1, an_2, \ldots, an_i, \ldots, an_n\}$ in UASN.
n: All number of acoustic sensor nodes.
S_i: State set of an_i extracted from Algorithm 2.
A: Action set
P: Transition probability
R: Reward function
ε: The parameter in ε-greedy strategy.
α: Learning rate
γ: Discount factor
M: Number of episodes
Output: Q-table stored in an_i
Begin
an_i: Get a random number from the interval $[0, 1]$ to determine ε;
an_i: Set the primary amounts of Q-values on zero;
repeat
  an_i: Select the next state (i.e. an_j) from the state set S_i randomly;
  for $t = 1$ to N **do**
    an_i: Elect a random number (N_r) in $[0, 1]$;
    if $N_r \leq \varepsilon$ **then**
      an_i: Carries out its action based on the ε-greedy strategy;
    else
      an_i: Designate its action according to the best Q-value in Q-table;
    end if
    an_i: Obtain the reward value based on Eq. (21);
    an_i: Calculate the transition probability using Eq. (17);
    an_i: Get the next state from the state set S_i randomly;
    an_i: Change the current state to the next state;
    an_i: Refresh Q-table according to the reward value;
  end for
  $episode = episode + 1$;
until $episode \leq M$
End

- **Step (2) Selection of the state set:** In this step, the void node modifies its state set. In QHRP, the state set corresponding to a void node such as an_i is S_i. To get this set, an_i considers all neighboring nodes such as an_j in NT_i and performs two filtering steps on these nodes.

  – **First filtering set:** In this step, an_i examines an_j (a member of NT_i) and determines whether an_i is belonging to S_j (the state set related to an_j) or not. As stated in Section 4.2, this filtering attempts to create free-loop paths to form a tree topology between underwater sensor nodes. The first filtering set is defined in Eq. (28).

  $$F_i^1 = \{an_j | an_j \in NT_i \ and \ an_i \in S_j \ and \ i \neq j\} \tag{28}$$

  – **Second filtering set:** In this step, an_i evaluates each member of NT_i such as an_j in terms of the depth information (d_j) and determines which neighboring nodes have more depth than an_i. Then, it puts these sensor nodes into the second filtering set, which is also stated in Eq. (29).

  $$F_i^2 = \{an_j | an_j \in NT_i \ and \ d_j - d_i > 0 \ and \ i \neq j\} \tag{29}$$

  If this set includes all neighboring nodes in NT_i, then the second filtering set must be re-modified and defined based on the number of hops to the sink. Thus, an_i evaluates each member of NT_i such as an_j in terms of the number of hops to the sink (i.e. h_j) and put sensor nodes with a higher number of hops within the second filtering set.

  $$F_i^2 = \{an_j | an_j \in NT_i \ and \ h_j - h_i > 0 \ and \ i \neq j\} \tag{30}$$

  If this set includes all neighboring nodes in NT_i, then the second filtering set must be ignored, and the second filtering set is equal to empty. This is stated in Eq. (31).

  $$F_i^2 = \varnothing \tag{31}$$

  Now, the union of two filtering sets is calculated through Eq. (32).

  $$F_i = F_i^1 \bigcup F_i^2 \tag{32}$$

  Now, an_i carries out the filtering operation on the state set through Eq. (33).

  $$S_i = NT_i - F_i = \{an_j | an_j \in NT_i \ and \ an_j \notin F_i\} \tag{33}$$

- **Step 3) Reward function:** In this step, the reward function is also redefined so that the void node can select the best parent node from its state set. In QHRP, the reward function is calculated based on remaining energy, strategic depth, hop counts,

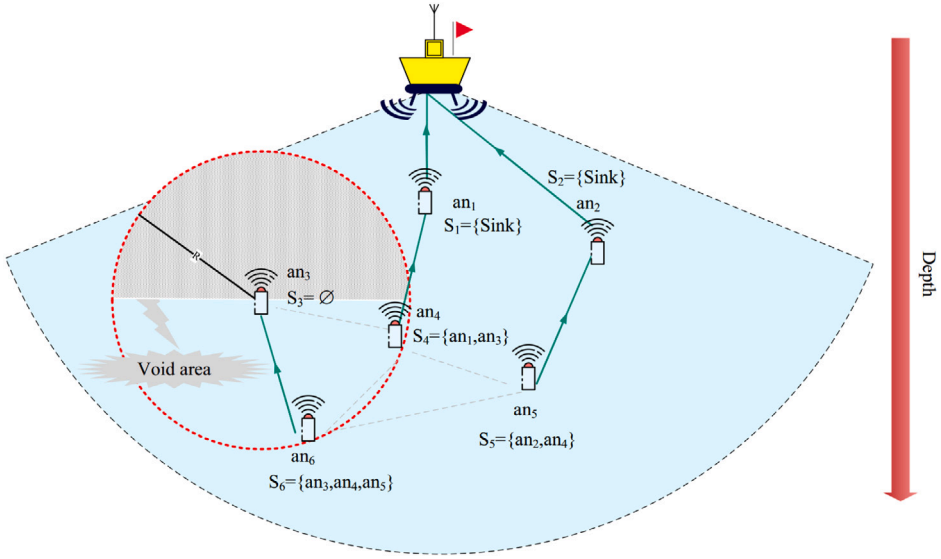

Fig. 8. Void area problem.

and successful transmission probability. This function is corrected via Eq. (34).

$$R_{s_i s_j}^{a_j} = \begin{cases} 1 - e^{-\left(\frac{E_j - \min\limits_{an_k \in NT_i}\{E_k\}}{\max\limits_{an_k \in NT_i}\{E_k\} - \min\limits_{an_k \in NT_i}\{E_k\}}\right)^{(P(d,l))^{\left(\frac{d_i - d_j}{R}\right)}}} & d_j \le d_i \\ 1 - e^{-\left(\frac{E_j - \min\limits_{an_k \in NT_i}\{E_k\}}{\max\limits_{an_k \in NT_i}\{E_k\} - \min\limits_{an_k \in NT_i}\{E_k\}}\right)^{(P(d,l))^{\left(\frac{h_i - h_j}{\max\limits_{an_k \in NT_i}\{h_i - h_k\}}\right)}}} & , \ d_j > d_i \ \ and \ \ h_j \le h_i \\ 1 - e^{-\left(\frac{E_j - \min\limits_{an_k \in NT_i}\{E_k\}}{\max\limits_{an_k \in NT_i}\{E_k\} - \min\limits_{an_k \in NT_i}\{E_k\}}\right)^{(P(d,l))^{\left(1 - \frac{|d_i - d_j|}{R}\right)}}} & , \ d_j > d_i \ \ and \ \ h_j > h_i \end{cases} \tag{34}$$

Where E_j indicates the residual energy of an_j. Furthermore, d_i and d_j represent the depths of an_i and an_j, respectively. $p(d,l)$ is the successful transmission probability calculated based on Eq. (5) in Section 3.2. Also, h_i and h_j indicate the number of hops related to an_i and an_j, respectively.

According to Fig. 10, after executing the recovery process and modifying the state set, an_3 can decide on the best parent node, an_4, in the routing tree, and solve the void area problem. After solving this problem, each sensor node in UASN continues the tree creation process in accordance with Section 4.2.

Algorithm 4 expresses the pseudo-code related to the recovery process and its time complexity is related to the size of the state space. In the worst case, the size of the state space is n, where n is the number of sensor nodes in the network. Thus, the complexity of Algorithm 4 is $O(n^2)$.

5. Performance evaluation

This section includes three subsections, assumptions, compared parameters, and simulation results. To compare the proposed scheme, three routing approaches, namely RLOR [25], EE-DBR [26], and MURAO [27] are chosen. This selection has different reasons mentioned below.

- The performance of the three selected methods, namely RLOR, EE-DBR, and MURAO, is acceptable in various parameters and can be compared to QHRP.
- Three routing approaches, namely QHRP, RLOR, and EE-DBR, use depth information in the routing process and do not need the location information of sensor nodes.
- Four routing approaches are energy efficient, and focus on the limited resources of sensor nodes in the network to optimize their energy consumption.
- QHRP and MURAO belong to the category of hierarchical routing approaches.
- QHRP, RLOR, and MURAO use the Q-learning algorithm to decide on the best forwarding paths in the network.

Algorithm 4 Recovery process

Input: an_i: i-th acoustic sensor node belonging to $N = \{an_1, an_2, \ldots, an_i, \ldots, an_n\}$ in UASN.
n: All number of acoustic sensor nodes.
NT_i: Neighbor table extracted from Algorithm 1.
S_i: State set of an_i extracted from Algorithm 2.
A: Action set
P: Transition probability
R: Reward function
ε: The parameter in ε-greedy strategy.
α: Learning rate
γ: Discount factor
M: Number of episodes
Output: Q-table stored in an_i
Begin
an_i: Create a hello packet according to Fig. 4 (a);
an_i: Broadcast the hello packet to the surrounding nodes in UASN;
for each node such as an_j, which is belonging to NT_i **do**
  if $an_i \in S_j$ **then**
    an_i: Delete an_i from S_j;
  end if
end for
for each node such as an_j, which is belonging to NT_i **do**
  if $an_i \in S_j$ **then**
    an_i: Append an_j to the first filter set F_i^1;
  end if
  if $d_j - d_i > 0$ **then**
    an_i: Append an_j to the second filter set F_i^2;
  end if
end for
if $F_i^2 = NT_i$ **then**
  for each node such as an_j, which is belonging to NT_i **do**
    if $h_j - h_i > 0$ **then**
      an_i: Append an_j to the second filter set F_i^2;
    end if
  end for
end if
if $F_i^2 = NT_i$ **then**
  $F_i^2 = \varnothing$;
end if
an_i: Calculate the union of two sets F_i^1 and F_i^2 ($F_i = F_i^1 \bigcup F_i^2$) based on Eq. (32);
an_i: Obtain the state set of an_i from Eq. (33);
an_i: Get a random number from the interval $[0, 1]$ to determine ε;
an_i: Initialize the initial values of Q-values on zero;
while $episode \leq M$ **do**
  an_i: Select the next state (i.e. an_j) from the state set S_i randomly;
  for $t = 1$ to N **do**
    an_i: Elect a random number (N_r) in $[0, 1]$;
    if $N_r \leq \varepsilon$ **then**
      an_i: Carries out its action based on the ε-greedy strategy;
    else
      an_i: Designate its action according to the best Q-value in Q-table;
    end if
    an_i: Obtain the reward value based on Eq. (34);
    an_i: Calculate the transition probability using Eq. (17);
    an_i: Get the next state from the state set S_i randomly;
    an_i: Change the current state to the next state;
    an_i: Refresh Q-table according to the reward value;
  end for
  $episode = episode + 1$;
end while
End

5.1. Simulation assumptions

Here, a network simulator (NS2)-based simulation operation is carried out to evaluate the performance of QHRP. Specially, the simulation process uses Aqua-Sim [35], which is a popular NS2-based object-oriented design style because a wide range of fundamental and sophisticated protocols are also available in Aqua-Sim, which facilitates the deployment of three-dimensional networks. This section assumes that the underwater acoustic sensor network is a cube, whose length, width, and height are $500 \times 500 \times 500\ m^3$. The number of sensor nodes in this cubic area varies from 50 to 600, and the simulation process presents the evaluation results in two modes, namely scattered (low-density) and dense network. When there are 50 to 300 nodes in the network, it is a scattered network, and when the number of nodes is more than 300, it is considered a dense network. Each sensor node is distinguished from other nodes via a unique ID, and the hydraulic pressure gauge installed on this sensor node can measure its depth at any moment. The network consists of a powerful, high energy, and immobile sink node, which is located on the center of

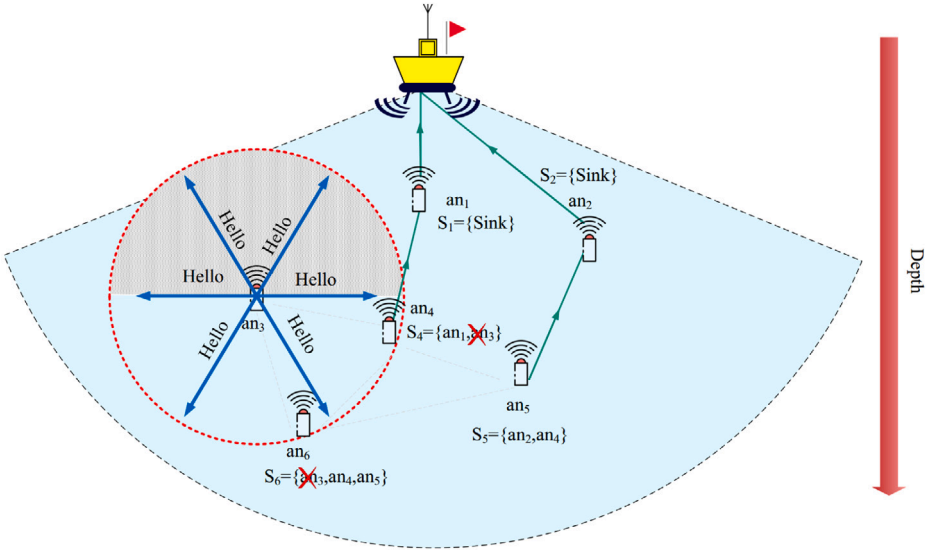

Fig. 9. Detection of the void area issue.

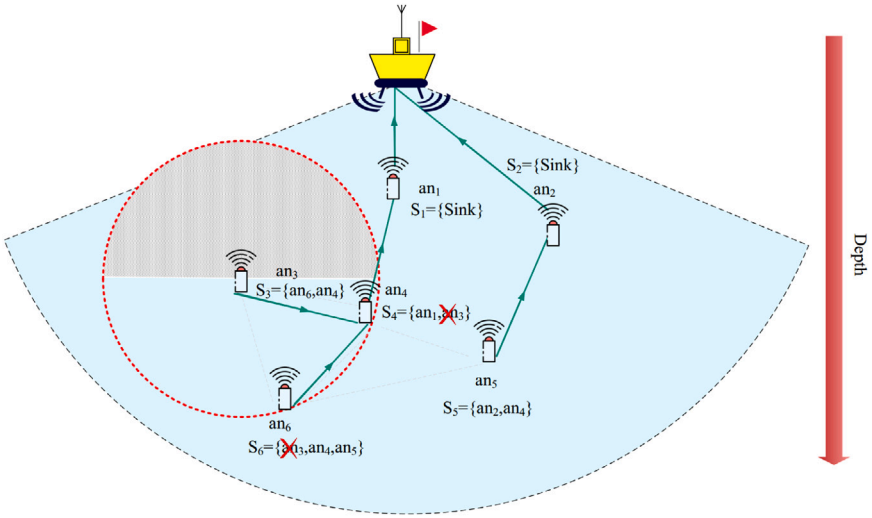

Fig. 10. Solving the void area problem.

the water surface and its depth is zero. The source underwater nodes horizontally move at a speed of 0 to 3 m/s, and this movement is simulated using the random walk mobility model. Here, only the horizontal movement of sensor nodes is considered because the nodes obtain their depth information at any moment. By repeating each test 20 times, presenting results on average, and ensuring a confidence interval of 95%, the evaluation process becomes more accurate. Table 5 briefly demonstrates the simulation parameters.

5.2. Compared parameters

The parameters considered in the performance evaluation process are the packet delivery rate (PDR), end-to-end delay (EED), data integrity, consumed energy, and the number of hops.

- **Packet delivery rate:** Eq. (35) obtains PDR, i.e. the percentage of packets delivered to the sink node with regard to the packets sent by the source node [25].

$$PDR = \left(\frac{n_{PK_d}}{n_{PK_s}} \right) \times 100 \tag{35}$$

Table 5
Simulation parameters.

Parameter	Value
Simulation software	NS2
Compared approaches	QHRP, RLOR, MURAO, EE-DBR
Evaluation parameters	PDR, EED, Data integrity, energy tax, average hop count
Simulation environment	$500 \times 500 \times 500$ m^3
Total number of nodes	50-600
Primary energy capacity	1000 J
Mac layer standard	IEEE 802.11
Antenna	Omni-Antenna
Simulation rounds	100
Transmission radius	100 m
Movement model	Random walk
Velocity of nodes	$0-3$ m/s
The size of packets	50 Byte
Packet generation rate	0.1 Packet/s
Signal frequency	25 KHz
Acoustic speed	1500 m/s
Runtime	5000 s
Discount factor	0.5
P_0	2
P_r	0.1

Here, n_{PK_d} and n_{PK_s} are corresponding to the number of packets delivered to the sink node and the number of packets sent by the source node.

- **End-to-end delay:** Eq. (36) obtains EED, i.e. the average time spent to transmit a data packet from the source node to the sink [36].

$$EED = \frac{\sum_{PK_i \in P = \{PK_1, \ldots, PK_n\}} \left(t_D^{PK_i} - t_S^{PK_i} \right)}{n_{PK_d}} \tag{36}$$

Here, n_{PK_d} denotes the number of packets received to the sink, PK_i is ith received packet, P indicates the set of delivered packets. Also, $t_D^{PK_i}$ and $t_S^{PK_i}$ mean the receiving and sending times related to PK_i, respectively.

- **Data integrity:** It is responsible for evaluating the content of the packets transferred from the source node to the sink. It determines whether the content of these packets has changed or not. A data packet contains various data segments that may be lost due to the void area problem or network collision. As a result, data integrity is damaged and data segments must be re-transferred. Hence, QHRP evaluates data integrity by comparing data packets posted by the source node with those delivered to the sink [25].
- **Consumed energy:** Eq. (37) obtains the energy consumed by a sensor node to deliver a data packet to the sink [23].

$$Energy\ tax = \frac{E_{Consumed}}{N \times n_{PK_d}} \tag{37}$$

so that $E_{Consumed}$ denotes the sum of the energy needed to send and receive the packet, and the idle mode. N is the number of nodes, and n_{PK_d} the number of packets received by the sink.

- **The number of hops:** It represents the average number of nodes needed to build a path between the source node and the sink. The ideal mode is that packets are always sent to the sink through the shortest route [25].

5.3. Simulation results

This section evaluates the performance of the four routing approaches, namely QHRP, RLOR, MURAO, and EE-DBR in terms of the five factors introduced in Section 5.2.

5.3.1. Packet delivery rate (PDR)

Fig. 11 performs a comparison between QHRP, RLOR, MURAO, and EE-DBR in terms of PDR. This evaluation shows the high PDR of QHRP, which is 9.068%, 18.73%, and 29.03% higher than that of RLOR, MURAO, and EE-DBR, respectively. As shown in Fig. 11, when there is a scattered network with less than 300 nodes, QHRP, RLOR, MURAO, and EE-DBR have weak performance and low PDR, but when there is a dense network with more than 300 nodes, all four routing approaches are stable and experience high PDR. This is because the scattered networks are damaged by two problems, namely the void area problem and unstable communication paths. As a result, many data packets are lost. However, QHRP is very successful in improving PDR because it is a hierarchical routing method, introduces a Q-learning-based routing tree technique, and forms a stable and energy-efficient tree between nodes in the network to minimize packet loss rate. Moreover, QHRP calculates a reward function based on remaining energy, strategic depth, the size of the state set, and successful transmission probability. This function affects directly the selection of parent nodes

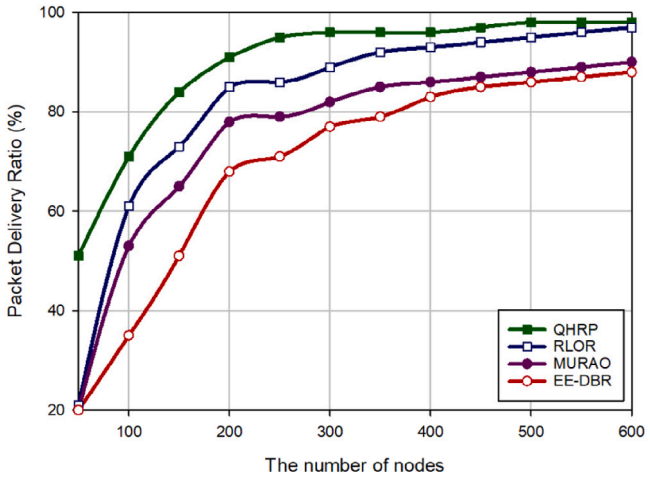

Fig. 11. Comparison of packet delivery rate based on the number of nodes.

in the routing tree and chooses a sensor node with high energy and high successful transmission probability as a parent node. In addition, QHRP attempts to solve the void area problem in the tree creation process so that the void node can also get out of the void area and select a parent node in the routing tree. This will significantly improve PDR.

5.3.2. End-to-end delay (EED)

Fig. 12 performs a comparison between QHRP, RLOR, MURAO, and EE-DBR in terms of EED. This evaluation shows the low delay of QHRP, which is 14.03%, 9.03%, and 18.66% better than that of RLOR, MURAO, and EE-DBR, respectively. As shown in Fig. 12, when there is a scattered network with less than 300 nodes, QHRP, RLOR, MURAO, and EE-DBR experience high delay in the data forwarding process, but when there is a dense network with more than 300 nodes, delay in these four approaches gradually decreases. This is because the scattered networks are more exposed to the void area issue. Thus, QHRP must constantly execute a time-consuming Q-learning-based recovery process to get out of the void area. This increases delay in the scattered networks. On the other hand, scattered networks include more unstable communication routes than dense networks because the nodes are far from each other, and there are fewer options to choose the next forwarder in the routing process. This can lead to more route failures and thus increase delay in the data forwarding process. According to Fig. 12, QHRP is very successful in reducing delay in the routing process because QHRP provides a new concept called strategic depth that combines the depth information and the number of hops to the sink. According to this new concept, each node prefers to remove neighboring nodes with more strategic depth from its state set because these sensor nodes are farther away from the sink and have more hops to the sink. Filtering these sensor nodes based on strategic depth ensures that the data reaches the sink quickly, and the data forwarding process experiences lower delay. Furthermore, QHRP includes a Q-learning-based tree creation process whose state set is a subset of neighboring nodes selected using two filtering steps. This effectively increases the convergence speed of the Q-learning algorithm and reduces delay caused by the tree construction process. In general, a hierarchical routing method effectively reduces delay in the data forwarding process. For example, according to Fig. 12, MURAO is very successful in reducing delay due to the use of a cluster-based hierarchical routing technique because it controls the energy consumption of sensor nodes and creates stable paths in the network.

5.3.3. Data integrity

Fig. 13 shows a comparison between QHRP, RLOR, MURAO, and EE-DBR in terms of data integrity. This evaluation shows that QHRP guarantees data integrity, which is 9.84%, 20.69%, and 28.48% better than in RLOR, MURAO, and EE-DBR, respectively. As shown in Fig. 13, when there is a scattered network with less than 300 nodes, QHRP, RLOR, MURAO, and EE-DBR cannot guarantee data integrity well. However, when there is a dense network with more than 300 nodes, all four methods guarantee data integrity. This is because the scattered networks are vulnerable to the void area problem, which results in the loss of some data segments and reduces data integrity. According to Fig. 13, QHRP guarantees data integrity well because this hierarchical routing approach involves a Q-learning-based tree creation. This routing tree attempts to balance traffic load on sensor nodes and increase the stability of paths to the sink. On the other hand, the tree creation algorithm selects parent nodes from the sensor nodes with high successful transmission probability. As a result, the packet loss caused by network congestion, packet collisions, and failed paths is significantly reduced. This improves data integrity. In addition, when building a routing tree, QHRP attempts to solve the void area problem efficiently to reduce data loss and improve data integrity.

5.3.4. Energy consumption

Fig. 14 shows a comparison between QHRP, RLOR, MURAO, and EE-DBR in terms of energy consumption. This evaluation illustrates the low energy consumption in QHRP, which is 15.61%, 27.27%, and 36.66% better than that in RLOR, MURAO, and

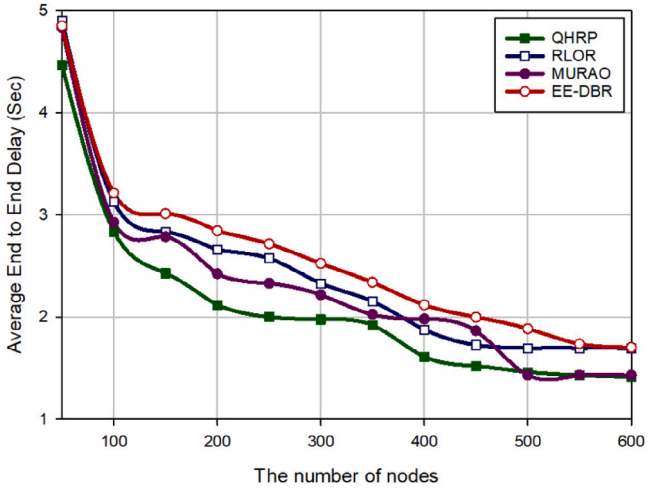

Fig. 12. Comparison of end-to-end delay based on the number of nodes.

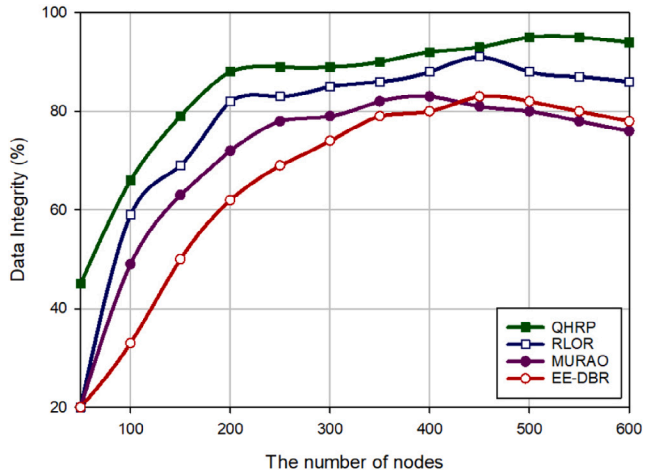

Fig. 13. Comparison of data integrity based on the number of nodes.

EE-DBR, respectively. As shown in Fig. 14, when there is a scattered network with less than 300 nodes, energy consumption in QHRP, RLOR, MURAO, and EE-DBR is very high, but when there is a dense network with more than 300 nodes, energy consumption in different approaches is lower. This is because the scattered networks are vulnerable to the void area issue that causes the failed routing paths. However, solving this problem and recovering the cut-off paths are time-consuming processes that require a lot of energy. However, in dense networks, the void area problem occurs fewer and causes less damage to UASNs. Thus, communication routes are more stable. As a result, sensor nodes consume less energy in the data forwarding process. In general, Fig. 14 shows the successful performance of QHRP in terms of energy consumption because the proposed hierarchical routing technique involves a decentralized Q-learning-based tree creation algorithm to balance the energy consumption of sensor nodes and increase the stability of the created paths. In this algorithm, the reward function is calculated based on remaining energy, strategic depth, the size of the state set, and successful transmission probability. According to this reward function, high-energy nodes have more priority to be selected as parent nodes. As a result, the routing tree can uniformly distribute the traffic load between the network nodes and manage their energy consumption. On the other hand, designing a recovery mechanism to solve the void area issue and modify communication routes increases the stability of the created paths and reduces route failures. This results in the optimized energy optimization in UASN.

5.3.5. The number of hops

Fig. 15 shows a comparison between QHRP, RLOR, MURAO, and EE-DBR in terms of the number of hops in communication paths. This evaluation shows that QHRP is successful in finding the shortest path between the source node and the sink and is 10.31%, 11.44%, and 20.15% better than RLOR, MURAO, and EE-DBR, respectively. As shown in Fig. 15, when there is a scattered

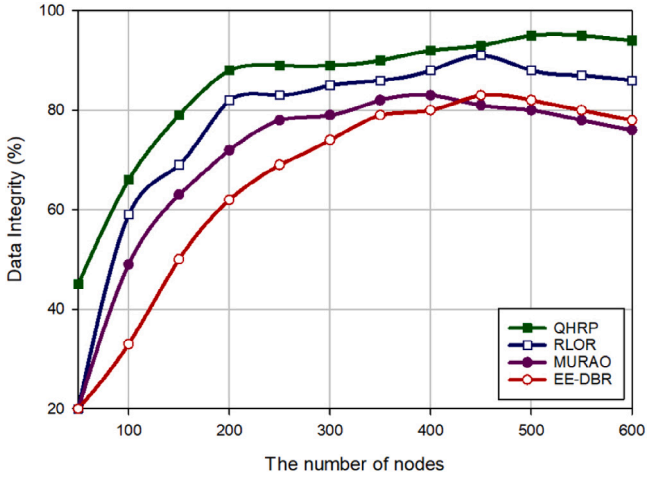

Fig. 14. Comparison of energy consumption based on the number of nodes.

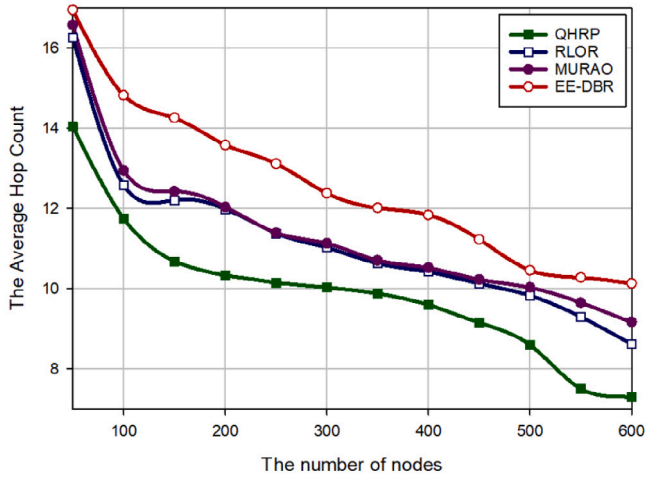

Fig. 15. Comparison of the number of hops based on the number of nodes.

network with less than 300 nodes, QHRP, RLOR, MURAO, and EE-DBR have a weak performance in finding the shortest paths. This is because sensor nodes are farther away from each other. Thus, each sensor node has fewer options for choosing its parent node in the routing tree and may not have an optimal choice. As a result, it is difficult to select the shortest route to the sink. However, when there is a dense network with more than 300 nodes, all four routing methods are more successful in finding the shortest route. This is because the probability of the void area issue in dense networks will be less. As a result, sensor nodes use the recovery mechanism fewer and can find shorter paths to the sink. In general, Fig. 15 shows the successful performance of QHRP for reducing the number of hops because the proposed routing approach uses a new concept called strategic depth (a combination of the depth of the node and the number of hops) to filter the state set and design the reward function. According to this new concept, each node prefers to filter neighboring nodes with more strategic depth because these sensor nodes are farther away from the sink and have more hops to the sink. Filtering these sensor nodes based on strategic depth guarantees that nodes with less depth and less hops are located in the state set and play the role of potential parent nodes in the routing tree.

6. Conclusion

This paper introduced a Q-learning-based hierarchical routing protocol (QHRP) in UASNs. This approach includes the Q-learning-based tree creation algorithm to balance the energy consumption of sensor nodes and increase the stability of paths in the network. In general, QHRP includes three steps: information exchange, Q-learning-based tree creation, and recovery process. In the first step, the information exchange process and the format of exchanged packets, namely hello, data packet, and ACK packet, were introduced. Secondly, QHRP explains a decentralized Q-learning tree creation process, which defines a filtered state set using two filtering steps so that each node can be selected as a set of potential parent nodes in the routing tree. The first filtering set attempts to create a tree

topology between sensor nodes. The second filtering also introduces a new concept called strategic depth, which is a combination of the depth information of nodes and the number of hops to minimize the number of hops and delay in the routing tree. In the third step, the void area problem is solved and a recovery process is presented to modify the state set and the reward function related to the void node so that this node can select its parent node in the routing tree. Finally, the simulation results of QHRP are compared to those of RLOR, MURAO, and EE-DBR. These results show that QHRP improves PDR, delay, data integrity, energy consumption, and the number of hops by 9.068%, 9.03%, 9.84%, 15.61%, and 10.31%, respectively. In future research directions, deep reinforcement learning techniques (DRL) and evolutionary algorithms will be used to design a precise state set and reduce delay in the data transfer process.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge Deanship of Graduate Studies and Scientific Research, Taif University for funding this work.

Data availability

No data was used for the research described in the article.

References

[1] Gupta S, Singh NP. Underwater wireless sensor networks: a review of routing protocols, taxonomy, and future directions. J Supercomput 2024;80(4):5163–96. http://dx.doi.org/10.1007/s11227-023-05646-w.
[2] Ayaz M, Baig I, Abdullah A, Faye I. A survey on routing techniques in underwater wireless sensor networks. J Netw Comput Appl 2011;34(6):1908–27. http://dx.doi.org/10.1016/j.jnca.2011.06.009.
[3] Islam KY, Ahmad I, Habibi D, Waqar A. A survey on energy efficiency in underwater wireless communications. J Netw Comput Appl 2022;198:103295. http://dx.doi.org/10.1016/j.jnca.2021.103295.
[4] Su X, Ren Y, Cai Z, Liang Y, Guo L. A Q-learning-based routing approach for energy efficient information transmission in wireless sensor network. IEEE Trans Netw Serv Manag 2022;20(2):1949–61. http://dx.doi.org/10.1109/TNSM.2022.3218017.
[5] Boukerche A, Sun P. Design of algorithms and protocols for underwater acoustic wireless sensor networks. ACM Comput Surv 2020;53(6):1–34. http://dx.doi.org/10.1145/3421763.
[6] Signori A, Campagnaro F, Nissen I, Zorzi M. Channel-based trust model for security in underwater acoustic networks. IEEE Internet Things J 2022;9(20):20479–91. http://dx.doi.org/10.1109/JIOT.2022.3176374.
[7] Han D, Du X, Liu X, Tian X. Fuzzy C-means clustering and improved arithmetic optimization algorithm-based layering cooperative routing protocol for UASNs. IEEE Sensors J 2024. http://dx.doi.org/10.1109/JSEN.2024.3413793.
[8] Moussa N, Nurellari E, Azbeg K, Boulouz A, Afdel K, Koutti L, et al. A reinforcement learning based routing protocol for software-defined networking enabled wireless sensor network forest fire detection. Future Gener Comput Syst 2023;149:478–93. http://dx.doi.org/10.1016/j.future.2023.08.006.
[9] Felemban E, Shaikh FK, Qureshi UM, Sheikh AA, Qaisar SB. Underwater sensor network applications: A comprehensive survey. Int J Distrib Sens Networks 2015;11(11):896832. http://dx.doi.org/10.1155/2015/896832.
[10] Su Y, Xu Y, Pang Z, Kang Y, Fan R. HCAR: A hybrid-coding-aware routing protocol for underwater acoustic sensor networks. IEEE Internet Things J 2023;10(12):10790–801. http://dx.doi.org/10.1109/JIOT.2023.3240827.
[11] Fan R, Jin Z, Yang W, Yang S, Su Y. A time-varying acoustic channel-aware topology control mechanism for cooperative underwater sonar detection network. Ad Hoc Netw 2023;149:103228. http://dx.doi.org/10.1016/j.adhoc.2023.103228.
[12] Zhu Z, Zhou Y, Wang R, Tong F. Internet of underwater things infrastructure: A shared underwater acoustic communication layer scheme for real-world underwater acoustic experiments. IEEE Trans Aerosp Electron Syst 2023;59(5):6991–7003. http://dx.doi.org/10.1109/TAES.2023.3281531.
[13] Nandyala CS, Kim HW, Cho HS. QTAR: A Q-learning-based topology-aware routing protocol for underwater wireless sensor networks. Comput Netw 2023;222:109562. http://dx.doi.org/10.1016/j.comnet.2023.109562.
[14] Khan ZA, Karim OA, Abbas S, Javaid N, Zikria YB, Tariq U. Q-learning based energy-efficient and void avoidance routing protocol for underwater acoustic sensor networks. Comput Netw 2021;197:108309. http://dx.doi.org/10.1016/j.comnet.2021.108309.
[15] Hosseinzadeh M, Yoo J, Ali S, Lansky J, Mildeova S, Yousefpoor MS, et al. A cluster-based trusted routing method using fire hawk optimizer (FHO) in wireless sensor networks (WSNs). Sci Rep 2023;13(1):13046. http://dx.doi.org/10.1038/s41598-023-40273-8.
[16] Pradeep S, Bapu TBBR, Rajendran R, Anitha R. Energy efficient region based source distributed routing algorithm for sink mobility in underwater sensor network. Expert Syst Appl 2023;233:120941. http://dx.doi.org/10.1016/j.eswa.2023.120941.
[17] Hosseinzadeh M, Tanveer J, Rahmani AM, Aurangzeb K, Yousefpoor E, Yousefpoor MS, et al. A Q-learning-based smart clustering routing method in flying Ad Hoc networks. J King Saud Univ- Comput Inf Sci 2024;36(1):101894. http://dx.doi.org/10.1016/j.jksuci.2023.101894.
[18] Hosseinzadeh M, Ali S, Ionescu-Feleaga L, Ionescu BS, Yousefpoor MS, Yousefpoor E, et al. A novel Q-learning-based routing scheme using an intelligent filtering algorithm for flying ad hoc networks (FANETs). J King Saud Univ- Comput Inf Sci 2023;35(10):101817. http://dx.doi.org/10.1016/j.jksuci.2023.101817.
[19] Wang C, Shen X, Wang H, Xie W, Zhang H, Mei H. Multi-agent reinforcement learning-based routing protocol for underwater wireless sensor networks with value of information. IEEE Sensors J 2023. http://dx.doi.org/10.1109/JSEN.2023.3345947.
[20] Sun Y, Zheng M, Han X, Li S, Yin J. Adaptive clustering routing protocol for underwater sensor networks. Ad Hoc Netw 2022;136:102953. http://dx.doi.org/10.1016/j.adhoc.2022.102953.
[21] Tian W, Zhao Y, Hou R, Dong M, Ota K, Zeng D, et al. A centralized control-based clustering scheme for energy efficiency in underwater acoustic sensor networks. IEEE Trans Green Commun Netw 2023;7(2):668–79. http://dx.doi.org/10.1109/TGCN.2023.3249208.

[22] Yuan Y, Liu M, Zhuo X, Wei Y, Tu X, Qu F. A Q-learning-based hierarchical routing protocol with unequal clustering for underwater acoustic sensor networks. IEEE Sensors J 2023;23(6):6312–25. http://dx.doi.org/10.1109/JSEN.2022.3232614.
[23] Wang C, Shen X, Wang H, Zhang H, Mei H. Reinforcement learning-based opportunistic routing protocol using depth information for energy-efficient underwater wireless sensor networks. IEEE Sensors J 2023;23(15):17771–83. http://dx.doi.org/10.1109/JSEN.2023.3285751.
[24] Han D, Du X, Liu X, Tian X. FCLR: Fuzzy control-based layering routing protocol for underwater acoustic networks. IEEE Sensors J 2022;22(23):23590–602. http://dx.doi.org/10.1109/JSEN.2022.3218136.
[25] Zhang Y, Zhang Z, Chen L, Wang X. Reinforcement learning-based opportunistic routing protocol for underwater acoustic sensor networks. IEEE Trans Veh Technol 2021;70(3):2756–70. http://dx.doi.org/10.1109/TVT.2021.3058282.
[26] Diao B, Xu Y, An Z, Wang F, Li C. Improving both energy and time efficiency of depth-based routing for underwater sensor networks. Int J Distrib Sens Networks 2015;11(10):781932. http://dx.doi.org/10.1155/2015/781932.
[27] Hu T, Fei Y. MURAO: A multi-level routing protocol for acoustic-optical hybrid underwater wireless sensor networks. In: 2012 9th annual IEEE communications society conference on sensor, mesh and Ad Hoc communications and networks. IEEE; 2012, p. 218–26. http://dx.doi.org/10.1109/SECON.2012.6275781.
[28] Su W, Chen K, Lin J, Lin Y. An efficient routing access method based on multi-agent reinforcement learning in UWSNs. Wirel Netw 2022;28(1):225–39. http://dx.doi.org/10.1007/s11276-021-02838-1.
[29] Stojanovic M. On the relationship between capacity and distance in an underwater acoustic communication channel. ACM SIGMOBILE Mob Comput Commun Rev 2007;11(4):34–43. http://dx.doi.org/10.1145/1347364.1347373.
[30] Freitag L, Grund M, Singh S, Partan J, Koski P, Ball K. The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms. In: Proceedings of OCEANS 2005 MTS/IEEE. IEEE; 2005, p. 1086–92. http://dx.doi.org/10.1109/OCEANS.2005.1639901.
[31] Geng X, Zhang B. Deep Q-network-based intelligent routing protocol for underwater acoustic sensor network. IEEE Sensors J 2023;23(4):3936–43. http://dx.doi.org/10.1109/JSEN.2023.3234112.
[32] Zhao Z, Liu C, Guang X, Li K. MLRS-RL: An energy-efficient multilevel routing strategy based on reinforcement learning in multimodal UWSNs. IEEE Internet Things J 2023;10(13):11708–23. http://dx.doi.org/10.1109/JIOT.2023.3243730.
[33] Su W, Chen K, Lin J, Lin Y. An efficient routing access method based on multi-agent reinforcement learning in UWSNs. Wirel Netw 2022;28(1):225–39. http://dx.doi.org/10.1007/s11276-021-02838-1.
[34] Zhang Y, Zhang Z, Chen L, Wang X. Reinforcement learning-based opportunistic routing protocol for underwater acoustic sensor networks. IEEE Trans Veh Technol 2021;70(3):2756–70. http://dx.doi.org/10.1109/TVT.2021.3058282.
[35] Xie P, Zhou Z, Peng Z, Yan H, Hu T, Cui JH, et al. Aqua-Sim: An NS-2 based simulator for underwater sensor networks. In: OCEANS 2009. IEEE; 2009, p. 1–7. http://dx.doi.org/10.23919/OCEANS.2009.5422081.
[36] Shen Z, Yin H, Jing L, Liang Y, Wang J. A cooperative routing protocol based on Q-learning for underwater optical-acoustic hybrid wireless sensor networks. IEEE Sensors J 2021;22(1):1041–50. http://dx.doi.org/10.1109/JSEN.2021.3128594.