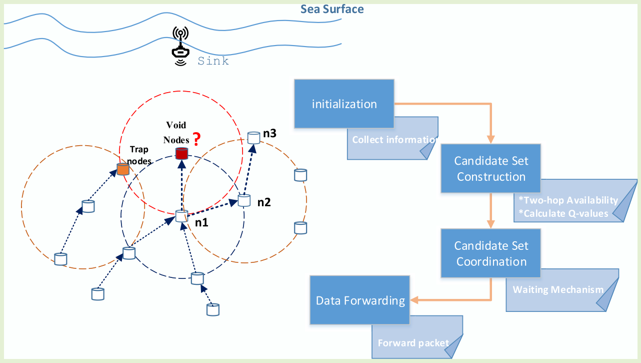
A Reinforcement-Learning-Based Opportunistic Routing Protocol for Energy-Efficient and Void-Avoided UASNs

Rongxin Zhu, Qihang Jiang, Xiangdang Huang, Deshun Li, and Qiuling Yang



***Abstract*—Underwater acoustic sensor networks (UASNs) have been used for various scenarios, such as marine exploration, and development of marine resources. However, due to the challenging underwater environment, UASNs suffers from high propagation delay, low transmission reliability and energy limitation, which pose significant obstacles to the delivery efficiency of data packets. What's more, the sparse topology, the failure of node or link, and other factors can cause void holes, resulting in the considerable package retransmission and the low network reliability of UASNs. To this end, this paper proposed a reinforcement-learning-based opportunistic routing protocol (ROEVA) to reduce energy consumption as well as to improve the transmission reliability which also addresses the issue of void routing in underwater acoustic sensor networks. To seek optimal routing rules, a reward function based on reinforcement learning is proposed, where factors such as energy, delay, link quality, and depth information are all taken into account for appropriate routing decisions. Before forwarding the data packet, a two-hop availability checking function is defined, which can identify trap nodes and avoid routing holes. In addition, aiming to reduce packet redundancy and collisions, a waiting mechanism derived from opportunistic routing is proposed. According to the calculated Q-values, the waiting mechanism determines the priority list for packet forwarding. Evaluation results demonstrate that the proposed ROEVA protocol outperforms HHVBF, RCAR, QELAR and GEDAR in terms of energy efficiency, packet delivery ratio (PDR), average hop count, and end-to-end delay, which are analyzed by varying the number of nodes from 100 to 500.**

***Index Terms*—Underwater acoustic sensor networks (UASNs), reinforcement learning, routing protocol, opportunistic routing, void hole.**

I. Introduction

UNDERWATER Acoustic Sensor Networks (UASNs) have become a prominent topic in recent years due to its potential for marine exploration and monitoring [1]. The concept of the Internet of Underwater Things (IoUT) came up in 2010s, which was inspired by the rapid development of IoT [2]. As a fundamental technology of IoUT, underwater sensor networks have emerged as a promising solution for observing underwater environments. UASNs have been widely applied to various fields such as catastrophe monitoring, oceanographic data collection, navigation assistance, and ocean sampling [3]. Thus, one of the major challenges for those underwater applications is routing and forwarding data packets from the source node to the sink [4]. However, owing to the harsh underwater environment and high deployment cost, deploying UASNs is much more challenging and expensive than terrestrial wireless sensor networks (WSNs) [5]–[7].

Since batteries are used to power sensor nodes, they are hard to be replaced or recharged in the underwater environment. In this context, energy efficiency should also be highly emphasized for UASNs due to the energy limitations as well as the higher communication energy cost than radio communications [5]. Moreover, compared with high-frequency radio waves and optical waves, the acoustic communication channel is relatively more suitable for underwater sensor networks.

Manuscript received March 24, 2022; revised April 26, 2022; accepted May 6, 2022. Date of publication May 18, 2022; date of current version July 1, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61862020 and Grant 62162021; in part by the Key Project of Hainan Province under Grant ZDYF2020199; in part by the Hainan Provincial Natural Science Foundation of China under Grant 2019RC096 and Grant 620RC560; in part by Scientific Research Set Up Fund of Hainan University under Grant KYQD(ZR)1877; in part by the Scientific Research Project of Hainan Province under Grant Hnky2022-4. The associate editor coordinating the review of this article and approving it for publication was Dr. Prosanta Gope. *(Corresponding author: Deshun Li.)*

Rongxin Zhu is with the School of Cyberspace Security, Hainan University, Haikou 570228, China, and also with the School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China (e-mail: rongxin.zhu@hainanu.edu.cn).

Qihang Jiang, Xiangdang Huang, Deshun Li, and Qiuling Yang are with the School of Computer Science and Technology and the State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou 570228, China (e-mail: jqh@hainanu.edu.cn; 990623@hainanu.edu.cn; lideshun@hainanu.edu.cn; 990709@hainanu.edu.cn).

Digital Object Identifier 10.1109/JSEN.2022.3175994

Although acoustic communication can make a long-distance propagation, the sound speed in underwater environments is approximately 1500 m/s, which is almost five orders of magnitude lower than the speed of radio channels [8], thus the low sound speed results in long propagation delay and severe Doppler distortions in communications. In addition, owing to the substantial signal loss and multipath impact in the underwater acoustic channel, the high packet loss rate makes the reliable transmission of data packet more difficult [9]. Therefore, an energy-efficient and reliable routing protocol is urgently needed.

In terms of transmission energy efficiency in UASNs, many existing routing protocols use the shortest path algorithms to select routing path for the purpose of reducing end-to-end delay and energy consumption. However, this causes hot spots of some shortest paths to run out of energy quickly, thereby reducing the lifetime of the entire network. In recent years, numerous sophisticated algorithms have been developed to improve energy efficiency. These routing protocols take residual energy into consideration in the selection of forwarding nodes, thus nodes with more remaining energy are more likely to be selected as forwarding nodes. Obviously, additional network metrics such as latency, network lifetime, energy usage, package delivery ratio, and globally optimal pathways are not taken into account by these algorithms. When nodes with more residual energy are selected, the number of hops for some packets delivering from the source node to the sink node is greatly increased, resulting in a significant increase in end-to-end delay. Furthermore, during route planning, information exchange requires a significant amount of energy. In addition, due to the greedy algorithm in determination of the next hop, some shortest routing pathways may not be the globally optimal. However, the energy consumption, end-to-end delay, and packet delivery ratio are the important metrics in UASNs. Therefore, comprehensive routing protocols should consider all of these indicators. To increase energy efficiency and network longevity, Zhou *et al.* [10] proposed a Q-learning-based location-free routing (QLFR) protocol that incorporates both energy and location when computing the Q-value.

To improve package delivery ratio in wireless networks, opportunistic routing (OR) has been developed. In this way, packet forwarding is enhanced by making use of simultaneous packet reception of neighboring nodes and utilizing the collaboration among the forwarding node's neighbors [11]. However, in most cases, it is impossible to apply directly terrestrial opportunistic routing protocols in underwater without considering the specific characteristics of UASNs. Moreover, forwarding set selection and prioritization are also affected by characteristics such as energy consumption, packet loss rate, and low bandwidth in the underwater environment. Rahman *et al.* [12] proposed a Energy Efficient Cooperative Opportunistic Routing (EECOR) protocol, which applies fuzzy rules to choose the optimal forwarder. Nevertheless, in some underwater scenarios, such as sparse networks with low density, the greedy hop-by-hop forwarding method may cause data packets to fall into void areas, where a forwarding node cannot find a suitable next hop. In addition, the movement of underwater sensor nodes with water flow may create void areas in the current network topology [13]. To avoid being trapped in void areas, the routing protocol tries to bypass or resume forwarding the data packet, otherwise it should be discarded. Ghoreyshi *et al.* [14] proposed a soft-state routing protocol which inherently excludes all those routes leading to a void area. As a result, there is no need for the protocol to enter recovery mode.

In this paper, we proposed a reinforcement-learning-based opportunistic routing protocol for energy-efficient and void-avoided underwater acoustic sensor networks (ROEVA). In ROEVA, a reward function is used to find the optimal route by balancing many factors of network performance in the harsh underwater environment. By considering both the instant rewards and the long-term rewards at a discount, ROEVA makes routing decisions globally optimal. Furthermore, two-hop availability checking method is used to identify void areas, and those nodes which are unable to forward package to the sink can be filtered in advance. In addition, the waiting mechanism based on the calculated Q-values for OR is applied to select a set of qualified forwarders to increase the package delivery ratio and reduce retransmissions.

The main contributions of this paper are as follows:

(1) We select the optimal forwarders based on the reinforcement learning technique. Thus, we design a reward function by considering the residual energy, depth information, end-end delay, and PDR, which makes the routing protocols more energy-efficient, reliable, and extensible.

(2) We deploy a specific two-hop availability checking function before selecting forwarders, which can identify trap nodes and avoid void areas. By solving the void routing problems, ROEVA improves the PDR and communication efficiency of the underwater network.

(3) We adopt OR paradigm to improve the reliability of transmission, and a waiting mechanism based on Q-value is designed. The priority of a forwarding node is determined by the waiting time of candidate nodes based on the calculated Q-value, which reduces redundant transmissions and packet conflicts through the broadcast characteristics of OR.

The remainder of this paper is organized as follows. Classical routing protocols are provided in Section II. The basic concept of the reinforcement-learning and related models is introduced in Section III. Then, in Section IV, the ROEVA protocol is described in detail. The evaluations and conclusions of this paper are presented in Section V and VI.

II. Related Work

In this section, we briefly present some existing well-known routing protocol for UASNs. A comparison of the protocols is listed in Table I.

Due to the harsh underwater environment and energy limitations, many routing protocols have been proposed to solve the energy efficiency problem. Literature [15] proposes a vector-based forwarding (VBF) routing protocol, that is, in which the source node and the sink node form a vector. As defined by

TABLE I
COMPARISON OF DIFFERENT PROTOCOLS

Protocols	Features	Challenges
VBF [15]	Robust and scalable	Not for sparse networks
DBR [16]	Reduce latency and improve PDR	More energy consumption
GEDAR [17]	Reduce latency and conflicts	Not energy-efficient
SEEORVA [18]	Secure and avoid void areas	Long end-to-end delay
VAPR [21]	Avoid void areas and retransmissions	Extra energy overhead
QELAR [24]	QL-based, extend lifetime	Long network latency
RCAR [25]	Avoid congestion	Low packet delivery ratio

VBF, only those nodes within the routing pipe can receive the packets broadcast by the source node and forward the packets from the source node to the receiver. The VBF protocol saves energy by limiting the number of forwarding nodes and direction to the sink. However, due to the fixed pipe radius and forwarding direction, the sensor nodes in the pipe are unavailable in a sparse network, affecting the performance of VBF. To avoid using geographic information, a depth-based routing protocol (DBR) is proposed in [16]. The source node forwards data packets to the sink nodes with lower depth by using a greedy strategy. Based on multiple sinks, the DBR protocol improves the packet delivery ratio and reduces the latency between sensor nodes. The DBR, on the other hand, ignores residual energy and thus the entire network lifetime.

Many UASN routing protocols use the OR paradigm to ensure communication reliability and efficiency. [17] proposes a geographic and Opportunistic Routing-based Routing (GEDAR), which enables depth adjustments of sensor nodes. The node uses a greedy method to forward data packets to the lowest depth forwarder, resulting in fewer hops between the data packet and the sink node. However, the greedy strategy makes some nodes consume serious energy. In [18], a secure and energy-efficient opportunistic routing protocol with void avoidance (SEEORVA) is proposed. SEEORVA adopts an opportunistic routing strategy for reliable data delivery and giving priority in the forwarding process to nodes with energy over a certain threshold, which can extend the network lifetime and improve energy efficiency. Moreover, in some scenarios, since the distribution of nodes are uneven, there are no available neighbor nodes in the communication range for the data forwarding, resulting in void routing [19]. The void areas in opportunistic routing have been investigated in both wireless networks and UASNs [20]. In [21], the authors propose a void-aware pressure routing (VAPR) protocol. Data packet is forwarded to the sink node along a directional path determined by beacon messages from sink nodes. When sensor nodes receive a beacon message, the VAPR protocol updates its forwarding direction according to the location of sender, then it detects void nodes by sending beacons. If the network contains a void node, extra overhead, such as position information in VBF and periodic beacons in VAPR, is required to circumvent the void node.

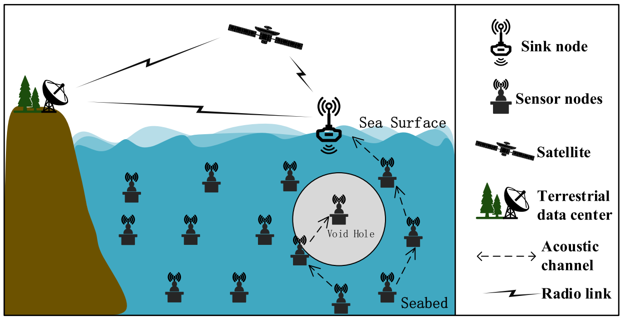


Fig. 1. Network model.

Although traditional routing protocols have improved network performance, the underwater environment is relatively harsh, and routing protocols must still deal with many constraints [22]. With the development of artificial intelligence, machine learning algorithms have been applied to underwater sensor network which improves the overall performance of the network [23]. In [24], a Q-learning-based adaptive routing (QELAR) protocol was proposed to realize energy-saving and extend the network lifetime in underwater sensor networks. The QELAR's reward function considers residual energy and makes routing decisions based on the calculated reward value. Nodes with higher residual energy are more likely to be chosen as forwarders, resulting in a longer network lifetime. However, in order to determine the next forwarding node, each node needs to learn environment information through packet exchanging, resulting in high energy consumption. [25] proposed a reinforcement-learning-based routing congestion avoidance (RCAR) protocol. In order to improve energy efficiency and reduce end-to-end latency, the RCAR makes routing decisions based on congestion and energy. However, they do not consider void holes and cannot guarantee the reliability of transmission.

III. NETWORK SYSTEM MODEL AND ASSUMPTIONS

In this section, we mainly introduce network architecture, underwater acoustic model, and Reinforcement-Learning technique which is the major technique adopted by ROEVA.

A. System Model

Figure 1 depicts our underwater network architecture. By deploying underwater sensor nodes, sensed data is collected from its surrounding area and sent to sink node through multi-hop forwarding. Sink nodes are deployed on the sea surface, making it convenient for charging or replacing batteries, allowing for energy conservation. On the surface, sink nodes are used to receive data packets from underwater sensor nodes through acoustic channels. Additionally, the sink nodes relay the data packets to terrestrial data center and satellites for data analysis via electromagnetic waves. Figure 1 also shows that routing holes may occur because the forwarding node does not have any neighbors with a depth shallower than itself to advance data packet towards the sink node.

Some assumptions:

(1) The surface sink node can use GPS to determine its location, and other sensor nodes are aware of the sink's

location. Sensor node can get its location by the existing service [26]–[28].

(2) Underwater sensor nodes can save their recent communication records in its local storage [24].

(3) All sensor nodes are in the uniform of transmission radius and initial energy.

B. Underwater Acoustic Model

The acoustic signal attenuation model [29] is used to calculate the PDR, which is denoted as $p(d, m)$, transmitting m bits over the distance d. In an obstacle-free propagation path, the attenuation factor of the acoustic signal is [6]:

$$A(d, f) = d^2 a(f)^d \tag{1}$$

where f is the signal frequency and S is the spreading factor which is set to 1.5 in actuality. The absorption factor $a(f)$ indicated by the Thorp equation is:

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

The average signal-to-noise ratio (SNR) with frequency f and distance d is:

$$\Gamma(d, f) = \frac{\mathrm{P}_{tra}/A(d, f)}{N(f)B} \tag{3}$$

where P_{tra} is the power for transmission and B is the channel bandwidth. The probability of bit error between nodes over distance d can be evaluated as [22]:

$$p_e(d) \approx \frac{1}{4SNR}. \tag{4}$$

Therefore, $p(d, m)$ successfully transmits m bits between any two separated nodes over the distance d:

$$p(d, m) = (1 - p_e(d))^m \tag{5}$$

C. Reinforcement-Learning Technique

Reinforcement learning is a branch of machine learning algorithms that achieve specific goals through the interaction between the agent and the environment [30]. Q-Learning is an unsupervised and value-based reinforcement learning algorithm. It is not necessary to have prior knowledge of the surroundings [31]. By taking action, the agent moves from one state to another and obtains the corresponding reward. The information obtained from environmental feedback is learned iteratively, thereby converging to the optimal policy ultimately.

Figure 2 describes the node with a tuple (s_i, a_i, r_i). s_i, a_i, and r_i denote the state, the action and direct reward of node i respectively.

The state of the node is busy or idle. When node i handles a packet, s_i will be set to busy. Action is very important for making forwarding decisions. In UASNs, the actions of each node are composed of its neighbors chosen as the next hop. Reward is the reflection of the quality of action taken by agent.

The agent selects action a_i under strategy π, then entering into next state s_j from state s_i. The performance of action taken by agent can be evaluated according to the calculated Q-value. The reward function $Q_\pi(s_i, a_i)$ contains two parts: the direct reward and discounted long-term rewards, defined as follows:

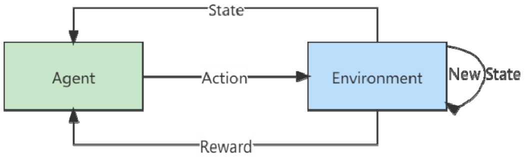


Fig. 2. Framework of Q-learning.

$$Q^\pi(s_i, a_i) = r_i + \gamma \sum_{s_j \in X} P^{a_i}_{s_i s_j} Q^\pi(s_j, a) \tag{6}$$

where r_i represents the direct reward and $\gamma \in (0, 1)$ represents the discount of the future reward in the total reward. $P^{a_i}_{s_i s_j}$ is the probability that agent in state s_i entering into state s_j. According to the Bellman equation, at least one optimal policy $\pi*$ [32] can be calculated out. Under the optimal strategy, the optimal value can be described as:

$$\begin{aligned} V^*(s) &= \max_a(Q^*(s, a)) \\ Q^*(s_i, a_j) &= r_i + \gamma \sum_{s_j \in X} P^{a_i}_{s_i s_j} V^*(s_j) \end{aligned} \tag{7}$$

$Q^*(s_i, a_j)$ is the expected reward, which is achieved by taking action a_j under the optimal policy at the state s_i. Furthermore, the optimal action a_i^* to obtain the largest Q is described as:

$$a_i^* = \arg\max_{a_i \in A(s_i)} Q(s_i, a_i) \tag{8}$$

The Q-learning framework of ROEVA is shown in Figure 2. The agent designs reward function with the factors of residual energy, end-end delay, packet delivery ratio, and depth information, which will be introduced in section IV-C.

IV. ROEVA PROTOCOL

In this section, we describe the details of ROEVA protocol, including the ROEVA protocol mechanism, the reward function of different parts for making routing decision, the packet format, and information interaction.

A. The Protocol Overview

The proposed ROEVA protocol is mainly used for finding the optimal routing for high energy-efficient and reliable communication in UASNs. The process of the ROEVA protocol is depicted in Figure 3. First, sensor nodes monitor channel conditions and update their local information. Before applying the Q-Learning technique to selecting candidate forwarders, ROEVA performs two-hop availability checking to identify trap nodes and void nodes which can help bypass these nodes and address the void area problem. Afterwards, the sender calculates the Q-values of candidate forwarders by Q-Learning. Candidate set for next hop is constructed by jointly considering the energy, delay, PDR, and depth of nodes. In addition, to avoid retransmissions and conflicts, ROEVA uses a waiting mechanism based on Q-value to realize opportunistic forwarding among multiple relay candidates.

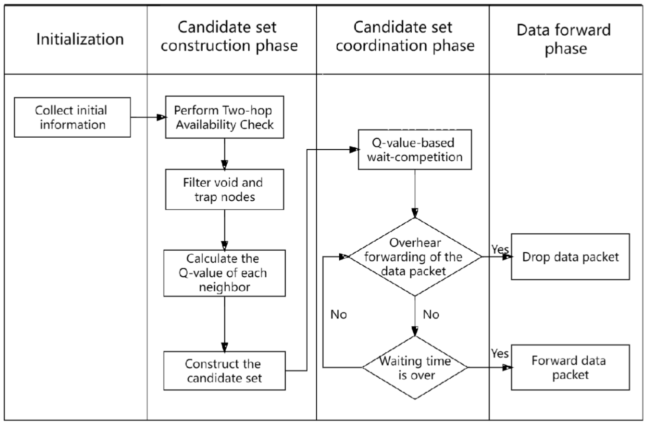

Initialization
Candidate set construction phase
Candidate set coordination phase
Data forward phase
Collect initial information
Perform Two-hop Availability Check
Filter void and trap nodes
Calculate the Q-value of each neighbor
Construct the candidate set
Q-value-based wait-competition
Overhear forwarding of the data packet
Yes
Drop data packet
No
No
Waiting time is over
Yes
Forward data packet

Fig. 3. Flowchart of ROEVA protocol.

TABLE II
LIST OF SYMBOLS

Symbol	Meaning
R_0	the constant cost
n_i	sequence of the sensor node
$Neighbor_i = \{n_1,n_2,\ldots, n_m\}$	neighbor set of node n_i (Neighbor$_i$ contains total m neighbor nodes)
$F_i(F_i \subseteq Neighbor_i)$	the candidate set of node n_i
Num_i	the number of node n_i's neighbor node
$\mathrm{Neighbor}_{is}$	the selected neighbor set of node n_i that satisfy the two-hop routing
$neighbor_{ij}$	The *j*th neighbor of node n_i
$E_{n_i}^{res}$	The residual energy of node n_i
$E_{n_i}^{ini}$	The initial energy of node n_i
T_i	The calculated waiting and holding time before forwarding data packet of node n_i
$P_b^{n_j}$	The number of buffered packets in node n_j
γ	Discount factor
$\alpha_1,\alpha_2,\alpha_3,\beta_1,\beta_2$	Sensitivity factor
T_{delay}	The predefined maximum delay
E_r,E_s	Energy for receiving and sending packets

Moreover, data packets in ROEVA protocol are transmitted from the seabed to the sink with multiple nodes' participation based on the OR paradigm. The waiting time of candidate nodes is related to the calculated Q-value, the larger Q-value means the higher the forwarding priority and the shorter waiting time. Especially, the waiting time of the candidate node with the highest priority is set to zero to reduce the transmission delay. Then the sender packages the candidate list including nodes number and waiting time into packet header. Sensor node retrieves packet header when receiving a data packet. If the current node is in the candidate list, it will hold the packet for a period of time according to the waiting time. Otherwise, the packet will be dropped. During the holding time, if the node overhears the forwarding of this packet by other nodes, it will also stop forwarding the packet. The notations in ROEVA protocol are listed in Table II.

B. ROEVA Protocol Mechanism

The ROEVA protocol contains four steps: initialization, candidate set construction, candidate set coordination, and data forwarding.

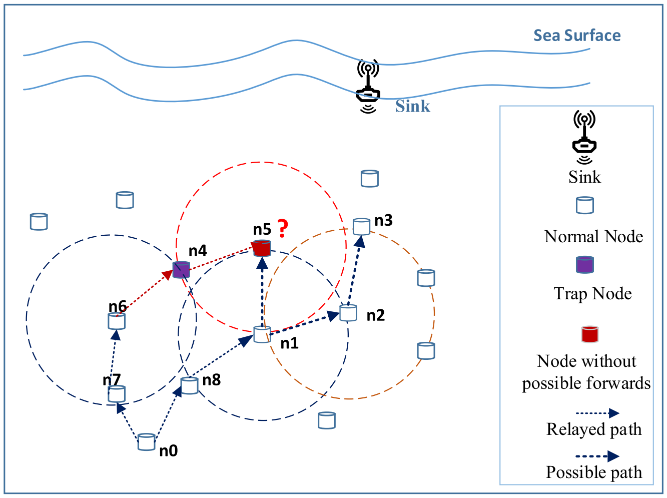


Fig. 4. Trap node and void node.

1) Initialization: At this stage, the initializations such as the neighbor tables, Q-tables, and node energy are set. The residual energy, locations, and other related information are obtained from physical layer. The sensor nodes exchange information with one-hop neighbors by broadcasting and receiving data packets. Each node has its own local neighbor table, storing information of neighbor nodes and Q-values for making routing decisions. In this way, each node can acquire the flow of information in the overall network instead of its neighbor nodes.

2) Candidate Set Construction: In this phase, ROEVA performs a two-hop routing availability checking method before selecting forwarders, thus identifying void nodes and trap nodes. The sender checks the reachability of adjacent nodes according to the neighbor information stored in the neighbor table, including the depth of nodes and the symbol of the trap or void node. In addition, the two-hop checking method can also help to identify the trap nodes which forward the packet towards the void node. Except for the void node, the sender checks the neighbor table to see if there are any other adjacent nodes with a higher depth than its own. If the answer is yes, the current node is ordinary; otherwise, it is a trap node. Then the sender updates its neighbor table with the checking result and broadcasts a packet to neighbor nodes.

For example, as shown in Figure 4, when data packets are forwarded to node n1 from n8, as n1 is the source node at the current time. Node n1 broadcasts a packet, and all the nodes in its transmission range receive the packet, where n2 and n5 are in the transmission range of nodes n1. According to the neighbor table, node n1 finds its corresponding adjacent nodes n2, and n5. Then, our protocol queries the adjacent information of nodes n2 and n5. The result shows that n5 has no adjacent nodes above it in transmission range, hence n5 is a void node; the adjacent node of n2 is n3 which meets the requirements. Therefore, void node n5 is excluded from the candidate adjacent nodes. In another case, when data packets are forwarded to node n4 from n6, n6 is the source node at the moment. In the transmission range of n4, n5 is the only adjacent node above n4. By performing two-hop routing

Algorithm 1 Two-Hop Availability Checking Algorithm

```
Procedure TwoHopCheck (source node $n_i$)
Get $Neighbor_i$ of $n_i$ and
Get the locations of $Neighbor_i$ from Neighbor table;
// Init the number of Neighbor $n_i$
$Num_i = 0$;
For $n_j \in Neighbor_i$ do
  // exclude neighbor which is a void node
  Query
  Get $Neighbor_j$ of $n_j$ from Neighbor table;
   // Init the number of Neighbor $n_j$
   $Num_j = 0$;
   For neighbor$_{ij} \in Neighbor_j$ do
     If ($neighbor_{ij}$ is above the location of $n_j$)
       $Num_j = Num_j + 1$;
     End if
   End for
   If ($Num_j > 0$)
     $Num_i = Num_i + 1$;
     Save $n_j$ in $Neighbor_{is}$;
   Else
Set node $n_j$ with the type of void node;
End If
End for
If ($Num_i == 1$&& that is a Void Node)
 Set node $n_i$ with the type of trap node;
End If
Return Neighbor$_{is}$;
End Procedure
```

availability checking method, n6 finds that the adjacent node n5 of n4 is a void node while referring to the neighbor table. Therefore, n6 updates its neighbor table and marks n4 as a trap node. Other adjacent nodes update their neighbor table after receiving the broadcast package from n6. The procedure of the proposed two-hop routing checking is described as Algorithm 1.

Generally, according to the forwarding policy of ROEVA, the node broadcasts the data packet to adjacent neighbors. The neighbor nodes that satisfy the two-hop routing can be chosen to forward the packet. They are qualified neighbor nodes that construct the candidate set F_i.

The nodes in the candidate set receive data packet, but they do not know how many other nodes receive the same packet. If we allow all of them to forward their successfully received packets, it will result in a broadcast storm in UASNs, which will not only waste the energy of sensor nodes but also occupy the bandwidth of acoustic channel. Owing to the limited energy and bandwidth in UASNs, routing protocol with high energy efficiency, more bandwidth, and a high delivery ratio is urgently required. Therefore, the sender should determine the forwarding priority list and package it into the packet header.

3) Coordination: At this stage, the sender determines the priority of candidate nodes by the calculated the Q-value. First, the sender calculates the Q-value of each node in F_i according to the neighbor table and physical information. Then, the waiting time based on the Q-value of each node, namely the priority of forwarding is computed. The waiting time of candidate nodes needs to be set reasonably. If the waiting time is too short, low-priority nodes cannot be suppressed and the data packet will be forwarded before the waiting time expires, resulting in data redundancy. Otherwise, the delay will be too long. Therefore, the sender calculates the node's waiting time through its buffered neighbor table, packet header, and calculated Q-value. The large Q-value means the high priority of the nodes, thus the node with a shorter waiting time participates in the forwarding. The sensor node with the highest Q-value is more likely to be selected as the next hop. The waiting time based on the Q-value is calculated as follows:

$$T_j = \begin{cases} 0, & Q = Q_{\max} \\ [1 - \dfrac{Q - Q_{\max}}{Q_{\max}}] * T_{dealy}, & Q \neq Q_{\max} \end{cases} \tag{9}$$

where T_{delay} is the maximal delay that candidates can hear the transmission of other nodes with a higher priority before relaying the packet. Q is the Q-value of node $\mathrm{n_j}$ and Q_{max} is the maximum value of all the candidates Q-value. Especially, when the Q-value is equal to the maximum value, T_j is set to zero, which means the node has the highest priority to forward data packets immediately to reduce end-to-end delay. In the case of more than one node's Q-value equal to the maximum value, a node among the multiple nodes with max Q-value at random sets T_j to zero while other nodes are calculated normally. Waiting time is used to suppress forwarding multiple copies of the same packet, thus other candidate forwarders can improve the package delivery when something goes wrong with the most proper node because of the dynamic underwater environment. Afterwards, the sender node will package the set of candidate nodes and their waiting time into data packet header.

4) Forwarding Phase: When a node receives a data packet, it checks the package header information which is carried by the packet. If the current node is one of candidate sets, it sets the waiting timer which is based on the received packet header.

An example is shown in Figure 5, node n2 constructs coordination set, where $F_1 = \{\text{n1, n3, n5}\}$. After nodes n1, n3 and n5 receive the data packet from n2, they start their own timers. According to the successfully received packet, it can be found that the candidate forwarder list contains n1, n3, and n5. and the waiting time of them is 0.511s, 0s, and 0.339s, respectively. Therefore, the timer of n3 ends for the first and starts the forwarding. Other candidate nodes n2 and n5 drop the data packet after hearing the transmission has been conducted before its timer ends. In this way, redundant packets and extra energy consumption can be avoided.

Afterwards, the data forwarding keeps on until the data packet reaches the sink node. In this way, a routing path is built completely. The consequent data packets from the same source node are directly forwarded along the completed routing path. Once the transmission fails, reinforcement learning algorithm will work and converge to other optimal routing paths. The data forwarding process of ROEVA protocol is shown in Algorithm 2.

Algorithm 2 Forwarding Data Packets

```
Procedure forwardingData(node $n_i$)
  Initialize $V(s)$;
  Get $E_{res}^{n_{ij}}, P_b^{n_{ij}}$ and locations of $Neighbor_i$;
  If ($n_i$ != sink node)
    $Neighbor_{is}$ ←TwoHopCheck($n_i$);
  For $n_{ij}$ in $Neighbor_{is}$ do
    Calculate the direct reward $R_{ij}$;
    Calculate $Q^*(s_i, a_i)$;
    Calculate $T_j$;
      Set a timer for $n_{ij}$ based on $T_j$;
      // coordination
      While the timer is not expired
        If overhear the packet has been forwarded
          Drop the data package;
        End if
      End while
      Update the rate of packet loss $P_{s_i s_i}^{a_i}$;
      Update $V(s)$ with the max $Q^*(s_i, a_i)$;
      Packet transmission;
    End for
  End if
End procedure
```

C. Design of Reward Function

The goal of the routing protocol based on Q-Learning is to obtain high energy efficiency and transmission reliability. We introduce the design of the reward function in detail in this section because it is such an important aspect of Q-Learning. The ROEVA protocol introduces multiple metrics into the reward function, including residual energy, end-end delay, depth, and link quality between current and next hops, to evaluate the plausibility of action execution which reflects the quality of routing decisions. Node n_i forwards a data packet while n_j is in its forwarders. $R_{s_i s_j}^{a_j}$ denotes the reward for taking action a_j in the successful transmission, defined as follows:

$$R_{s_i s_j}^{a_j} = -\mathrm{R}_0 - \beta_1 * c(en) + \beta_2 * c(env) \tag{10}$$

where $\beta_1, \beta_2 \in (0, 1)$.

The reward function considers the following cost: constant cost, energy-related cost, and environment-related cost. The constant cost R_0 is due to the factor that the communication occupies the channel bandwidth, and the constant cost accumulates with the increase of the number of hops. If there is only a constant cost in the reward function, it will cause the node to only choose the shortest path as the protocol converges. However, the shortest path may not be the optimal path considering the unbalanced energy consumption and transmission reliability. Thus, other aspects such as residual energy, delay, package delivery, and depth are taken into consideration.

c(en) is the energy-related cost which considers the residual energy and initial energy. The successful transmission of c(en) is defined as:

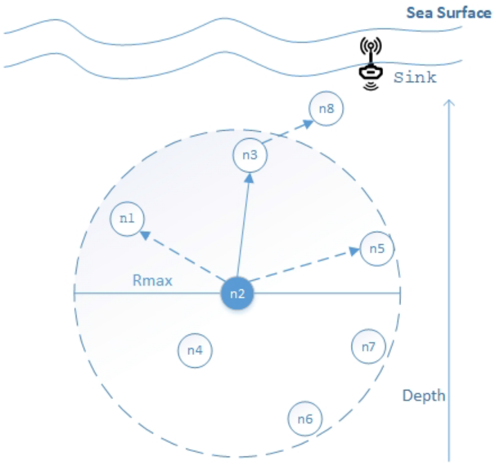


Fig. 5. Waiting mechanism.

$$\begin{aligned} c(en) &= \mathrm{c}_i(\mathrm{en}) + \mathrm{c}_j(\mathrm{en}) \\ &= (1 - \frac{E_{res}^{n_j} - E_r}{E_{ini}}) + (1 - \frac{E_{res}^{i} - E_s}{E_{ini}}) \end{aligned} \tag{11}$$

The residual energy of the node is compared with its initial energy, and more residual energy means a lower cost. Selecting nodes with more residual energy makes a balance the distribution of energy, thereby prolonging the network lifetime in UASNs.

The last part c(env) is the environment-related cost, and the environment means not only physical environment but also communication environment. c(env) specifically includes delay-related cost, link-related cost and depth-related cost, defined as:

$$c(env) = -\alpha_1 * c(delay) + \alpha_2 * c(pdr) + \alpha_3 * c(dep) \tag{12}$$

where $\alpha_1, \alpha_2, \alpha_3 \in (0, 1)$.

c(delay) reflects the condition of sensor network. A large number of packets in the buffer of the nodes results in high end-end delay. c(delay) is defined below:

$$c\,(delay) = 1 - \frac{1}{p_b^{n_j} + 1} \tag{13}$$

where $P_b^{n_j}$ denotes the number of buffered packets in neighbor node n_j. When there are more packets in the buffer of neighbor node, the estimated waiting time for a data packet to be successfully transmitted from that neighbor node to the next-hop is longer which makes the data packets waiting in the queue for a longer time. Therefore, in this case, the cost of c(delay) is higher.

c(pdr) is the packet delivery-related cost, namely PDR, which reflects the link quality of transmission in UASNs. PDR is defined as the number of packets received divided by the number of packets transmitted. ROEVA estimates the packet delivery ratio through the acoustic signal attenuation model, which is denoted as $p(d_j, m)$:

$$c(pdr) = p(d_j, m) \tag{14}$$

The PDR is important to evaluate the reliability of routing. The node with the high delivery cost is considered to be more reliable in packet transmission, thereby the node has a greater possibility to be selected as forwarder.

c(dep) is related to the depth of sensor nodes, defined as follows:

$$c(dep) = \frac{d(n_i, n_j)}{dep_{\max}} \tag{15}$$

where $d(n_i, n_j)$ is the depth between n_i and its adjacent node n_j. dep_{max} represents the transmission range. Clearly, when the depth of the candidate node n_j is less than the current node, namely $d_j - d_i > 0$, c(dep) is above zero and considered to be a reward. Otherwise, $d_j - d_i < 0$, it means c(dep) is below zero and regarded as a punishment.

Since the proposed ROEVA mainly addresses the energy problem, both c(en) and c(env) are in the range of (0,1) by definition, which enables us to balance $\beta 1$ and $\beta 2$ in (10) by tuning the weights only. Other factors in c(env) such as c(delay), c(link), and c(dep) can be further tuned for various demands, hence the factors of $\alpha_1, \alpha_2, \alpha_3$ are set to 0.33 simply for the same weight.

In the other case, it means the failure of transmission. After the packet retransmission reaches the limitation, the next forwarder still does not receive the packet, resulting in the time consumption. The retransmission of data packets increases delay and energy consumption, which increases the cost for failed transmission. The reward function for failure is defined as follows:

$$\begin{aligned} R^{a_j}_{s_i s_i} &= -\mathrm{R}_0 - \beta_1 * c(en) + \beta_2 * c(env) \\ c(env) &= -\alpha_1 * c(delay) + \alpha_3 * c(dep) \end{aligned} \tag{16}$$

where

$$\beta_1, \beta_2, \alpha_{1-3} \in (0,1) c(en) = \mathrm{c}_i(\mathrm{en}) = 1 - \frac{E^i_{res} - E_s * N_{\max}}{E_{ini}}$$

$$c\,(delay) = 1 - \frac{1}{p_b^{n_j} + 1}$$

$$c(dep) = \frac{d(n_i, n_j)}{dep_{\max}}$$

According to the above, the direct reward of transmission in both success and failure is defined as follows:

$$r_i(a_j) = P^{a_j}_{s_i s_j} R^{a_j}_{s_i s_j} + P^{a_j}_{s_i s_i} R^{a_j}_{s_i s_i} \tag{17}$$

In order to estimate the channel state and state transition probability, sensor nodes record recent packet transmissions. The count of packets lost is denoted as λ while the total number of packet transmissions is n. Thus, loss rate $P^{a_j}_{s_i s_i}$ and successful transmission rate $P^{a_j}_{s_i s_j}$ are respectively expressed as:

$$\begin{aligned} P^{a_j}_{s_i s_i} &= \frac{\lambda}{n} \\ P^{a_j}_{s_i s_j} &= 1 - \frac{\lambda}{n} \end{aligned} \tag{18}$$

Therefore, substituting $P^{a_j}_{s_i s_i}$ and $P^{a_j}_{s_i s_j}$ into reward function, we can update the reward function as follows:

$$Q(s_i, a_j) = r_i(a_j) + \gamma\,((1-\frac{\lambda}{n})Q^*(s_j) + (\frac{\lambda}{n}) * Q^*(s_i)) \tag{19}$$

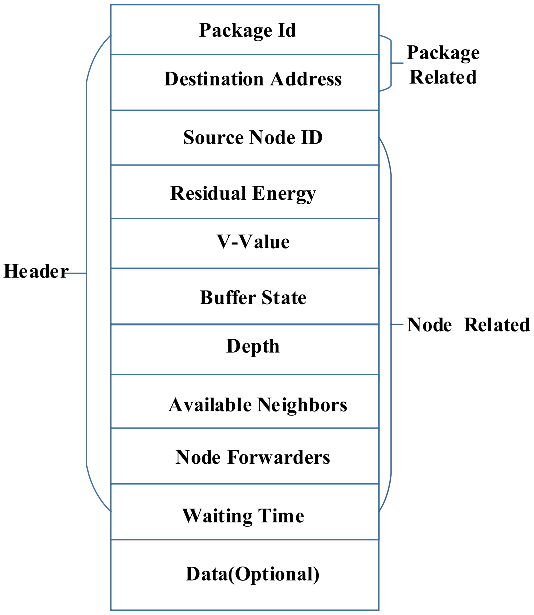


Fig. 6. Packet structure.

The Q-value of ordinary nodes is set to zero initially. Q-value of each node is updated based on the information interaction of the network. Since the Q-values of ordinary nodes are less than 0 after successful forwarding, the sink's Q-value is set to 0. This allows the protocol to converge.

D. Packet Structure

The data packet structure of ROEVA is pictured in Figure 6, which is designed to exchange information in UASNs. The packet header and the data constitute the data packet in ROEVA. The packet header is divided into two parts: packet-related fields and node-related fields. The packet-related fields are the unique ID of the packet and the destination address to which the package should be sent to; The node-related fields include source node ID, residual energy, V value, buffer state, depth, available neighbors, node forwarders, and waiting time list, in which:

1) Source Node ID, the source node ID, namely the current node sending the package.
2) Residual Energy, residual energy of current node.
3) Q-Value, the node's calculated Q-value.
4) Buffer State package, the number of nodes buffered in the current node.
5) Depth, the depth of current node.
6) Available Neighbors, the count of current node's neighbors which are in the transmission range and above the current node.
7) Forwarders, candidate set, and the position of each node in the set can represent the priority of nodes.
8) Waiting Time, the list of suppressing time before forwarding packet corresponding to Forwarders.

Once a sensor node receives a data packet, it retrieves the packet header whether it is in the candidate relay set and the relevant information of the previous node from the packet header and its neighbor table. If so, ROEVA uses the reinforcement-learning technique to calculate the Q-values of its qualified neighbor nodes are based on various related costs, and the packet header is packaged with the required fields in the packet structure, waiting for the data forwarding. Otherwise, the node simply drops the data packet after updating its local information.

The other part of data packet structure is the Data, which is optional. This part may contain data from upper-level protocols when the payload data is not absent. Otherwise, the data packet is just for information exchange.

E. Information Exchange and Overhead Analysis

For information to flow through the network, nodes exchange their local information with neighbors. When a node has packets to forward, the node should select candidate forwarders. Except for its own local information, the calculated forwarding list is also added to the packet header. Because nodes monitor the network traffic, they overhear the data packets and retrieve the packet for data update, even if the packet is not forwarded towards it.

Through the above introduction, the overhead of the ROEVA mainly comes from the information exchanging, storage of routing information, and computation. The first is the information exchanging overhead, in order to reduce the information exchanging overhead, the updating mechanism is designed to work in a cycle time based on the packet generation rate, hence this part of overhead can be neglected. Moreover, the storage of information including packet ID, the source node ID, the forwarders, two-hop neighboring ID, and the destination all can be represented by one byte will be no more than 16 bytes [24], hence the storage overhead can be ignored. Furthermore, the sensor nodes need to compute Q-values when selecting forwarders. Because the computation is not complex, the delay and energy consumption for computation are lower than those in acoustic communication [28]; Thus, the overhead of the computation can be ignored as well.

V. Evaluations

In this section, evaluations on the performance of the ROEVA are conducted based on the Aqua-sim platform [30]. We introduce our simulation settings before evaluations. Afterwards, we assess the performance of ROEVA with different parameters. The metrics of the ROEVA, including PDR, energy consumption and end-to-end delay, are compared with RCAR, QELAR, HHVBF, and GEDAR.

A. Simulation Settings

In the simulations, sensor nodes are deployed in a 4000 m × 4000 m × 5000 m 3D area at random. We deploy the sink node on the sea surface, which not easy to move after being deployed. We choose a source node at the seabed of the network for analysis. The speed of acoustic transmission is set to $v_0 = 1500$; the number of sensor nodes varies between 100 and 500. The parameters of the simulation are shown in Table III [33], [34].

TABLE III
Simulation Parameters

Parameter	Value
Initial energy	1000J
Transmission power	10W
Receiving power	3W
Idle power	30mW
Data packet size	50Bytes
Transmission rate	1kbps
Communication range	1500m
Packet generation rate	0.1packet/s
Simulation time	2000s
$\alpha_1, \alpha_2, \alpha_3$	0.33

Similar to most researches on routing protocols, we use the Carrier Sense Multiple Access (CSMA) MAC protocol as the underlying MAC protocol. If the channel is free, the forwarder can broadcast the packet; otherwise, it backs off and the packet will be dropped after backing off for five times [34]. Furthermore, we evaluate the performance of the ROEVA protocol using the following four quantitative metrics: Average End-to-End Delay represents the average time it costs for sending a data packet from the source node to the sink node. Packet Delivery Ratio is the ratio of the number of data packets successfully received by sink node to that of data packets sent by the source node. The average Hop Count of Delivered Packets is defined as the average hop count on the routing path from source to destination. Energy Consumption is the total energy consumption of all nodes for packet transmission.

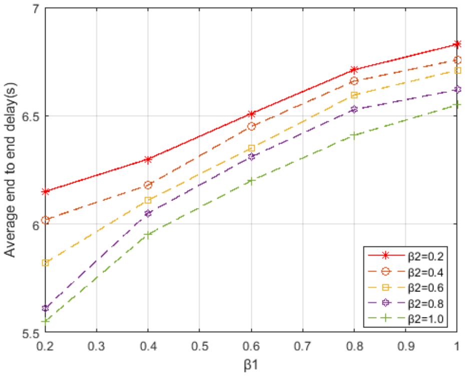


Fig. 7. Average end-to-end delay of $\beta 1$ and $\beta 2$.

B. Performance Evaluation

1) Parameter Analysis: To test different coefficients, 300 nodes with a communication range of 1500m are deployed in the network, and the effects on residual energy variance and average end-to-end delay are investigated. Figure 7 and Figure 8 show the effects of $\beta 1$ (energy cost weight) and $\beta 2$ (environment related sensitivity), respectively. These coefficients affect the reward function, where the value of $\beta 1$ varies between 0.2 and 1.0, and the value of $\beta 2$ varies between 0.2 and 1.

As shown in Figure 8, the residual energy variance increases with the value of $\beta 2$ increasing, because environmental sensitive elements such as packet delivery ratio, node depth, and transmission delay have an impact on routing decisions, the

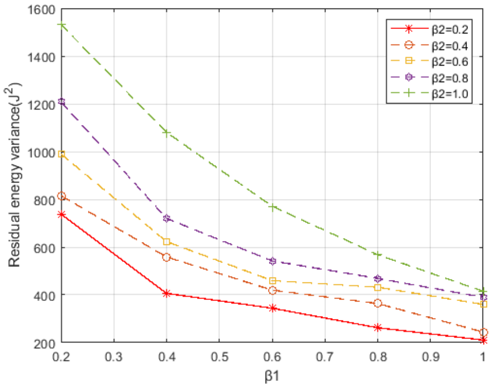


Fig. 8. Residual energy variance of $\beta 1$ and $\beta 2$.

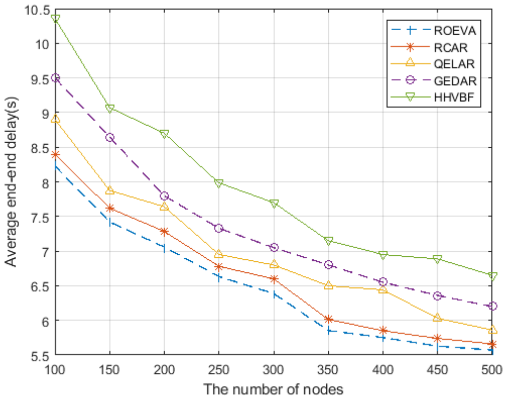


Fig. 9. Comparison of average end-to-end delay.

residual energy variance grows as the value of 2 increases. As a result, a delay-limited routing path cannot guarantee even energy distributions. Correspondingly, Figure 7 shows that the increase of $\beta 2$ encourages the protocol to select the node which considers environmental factors as the forwarder. Therefore, the protocol can not only converge to the path to improve energy efficiency with a minimum number of hops, but also reduce latency. Specifically, when $\beta 2$ and $\beta 1$ are set to 0.8 and 0.2, respectively, the average end-to-end delay is 5.61s, which is about 16% lower than that of $\beta 2 = 0.2$ and $\beta 1 = 0.8$. By comparing Figure 7 and Figure 8, it can be seen that although increasing $\beta 2$ can reduce the end-to-end delay, it also increases the residual energy variance, thereby shortening the network lifetime.

According to Figure 8, the residual energy variance decreases as $\beta 1$ increases. For example, while $\beta 2$ is 0.2, the residual energy variance at $\beta 1 = 0.8$ is 71% lower than at $\beta 1 = 0.2$. This is because energy has a greater influence on the reward function and becomes a more important factor in routing decisions. The larger the value of $\beta 1$, the more likely it is that a node with greater residual energy will be chosen as a forwarder. The energy consumption between nodes is more uniform, thus the network life can be extended. We can also observe in Figure 8 that the residual energy variance increases as $\beta 2$ increases, as link quality, physical depth, and delay take a larger proportion of routing decisions. Similarly, considering only the globally optimal path cannot guarantee a uniform distribution of the residual energy.

Therefore, we can conclude that a larger value of $\beta 1$ distributes energy more uniformly, but this makes the average end-to-end delay higher. While a larger value of $\beta 2$ means lower delay and higher residual energy variance. The values of $\beta 1$ and $\beta 2$ are weighed according to the situation, and different values are selected to satisfy different requirements in the network. Therefore, in the following evaluations, $\beta 1$ and $\beta 2$ are set to be equal, and both are set to 0.5.

2) Comparison With the State-of-the-Arts: Figure 9 shows the average end-to-end delay of ROEVA. Comparing with RCAR, QELAR, HHVBF, and GEDAR. The average end-to-end delay decreases as the number of nodes grows. When the number of deployed nodes becomes larger, packets can be forwarded along shorter routing paths. Therefore, when the number of sensor nodes is 500, the end-to-end delay is relatively minimal. Furthermore, we can also see that the average end-to-end delay of these protocols decreases, when there are more nodes in the network. Because more nodes mean the deployment is denser, the optimal path can be found in the network from the source node to the sink node. In addition, the average end-to-end delay of ROEVA protocol is apparently better than other protocols. When there are 500 nodes in the network, ROEVA's average end-to-end delay is 5.57s, while RCAR, QELAR, GEDAR, and HHVBF are 5.66s, 5.86s, 6.2s, and 6.65s, respectively. The reason is that ROEVA avoids void areas and uses Q-value-based waiting mechanism to coordinate forwarders, which is useful for reducing retransmissions of data packet and avoiding collisions. The average end-to-end delay of RCAR is only second to ROEVA because RCAR avoids congestion and considers multiple factors to make routing decisions. However, it does not take the void areas which results in retransmissions into account. Meanwhile, the average end-to-end delay of HHVBF is the highest among the five protocols. Collisions occur as a result of the hidden terminal problem, which increases the average end-to-end delay. Moreover, the average delay of GEDAR ranks second, because GEDAR uses opportunistic routing to improve the delivery ratio of packet, but it takes time to shift void nodes to other locations, resulting in extra end-to-end delay.

Figure 10 shows the comparisons in PDR for different protocols. We can see that the PDR of all the listed protocols increases with the number of nodes. This is because the sender nodes have more neighbor nodes for forwarding the data packet, which improves the packet delivery rate. Simultaneously, it takes fewer hops for the data packet from the source node to the sink node within the communication range. The PDR of RCAR and QELAR increases because the sender can find a path with fewer hops to make the packet delivery rate higher. GEDAR already takes PDR into consideration for expected packet advance, and more than one node participate in the transmission, thus the PDR of GEDAR in the figure is higher. The HHVBF does not consider the depth between nodes and the packet error rate. Therefore, unnecessary detours and high packet error rates lead to low PDR. We can also observe that the PDR of ROEVA is higher than the other four protocols. For example, when the number of nodes is 500, the PDR of ROEVA reaches 90.6%, which is higher compared

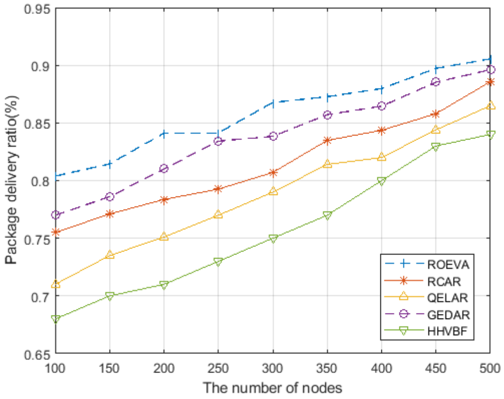


Fig. 10. Comparison of packet delivery ratio.

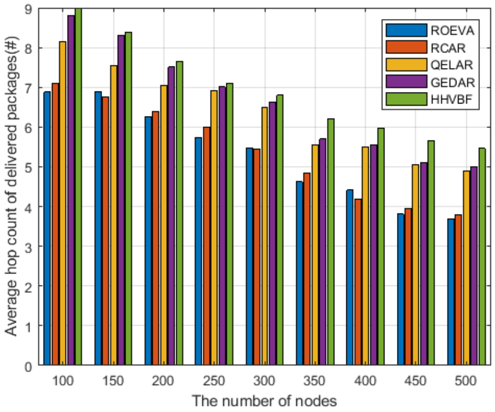


Fig. 11. Comparison of average hop count of delivered packages.

with GEDAR, RCAR, QELAR, and HHVBF. On one hand, the ROEVA performs a two-hop routing availability checking function in ROEVA to avoid getting stuck in void areas and reducing retransmissions. On the other hand, the ROEVA protocol not only considers the PDR for making routing decisions, but also other factors, including depth information and end-to-end delay, which ensure high PDR in global.

Figure 11 shows the average hop count of delivered packets for different protocols. In some scenarios, such as the sparse network, sensor nodes cannot completely cover the shortest routing path between the source node and the sink node, therefore, the average hop count and the ratio of packet delivery should be traded off. More nodes are involved in the data forwarding. Figure 11 shows that as the number of nodes increases, the average hop count of delivered packets decreases, and the results are clearly in line with our expectations. This is because the node density becomes larger, and the packets will be forwarded along optimal routes, and therefore fewer nodes are involved in forwarding. Furthermore, among these protocols, the protocols including ROEVA, RCAR, and QELAR which are based on reinforcement-learning, have lower average hop counts than other protocols, because they use intelligent algorithms to select the optimal path for data forwarding and emphasis different metrics of network in their reward functions. For example, when there are 500 nodes in the network, the average hop count of delivery packet is 3.7, while that of RCAR, QELAR, GEDAR, and HHVBF is 3.9, 4.9, 5.1, and 5.5, respectively. One of the reasons is that ROEVA performs two-hop routing availability checking and has a global view of the network topology by reinforcement learning, it can avoid void areas and choose the optimal routing path, thus reducing the average hop-count of delivered packages and adapting to different scales of network architectures. However, HHVBF is limited to nodes within the pipe radius and not flexible enough to find the path with the fewest hops to the receiver. In addition, GEDAR advances packets based on greedy forwarding strategy, rather than based on hop count, which makes GEDAR selecting routing paths with more hop counts than ROEVA.

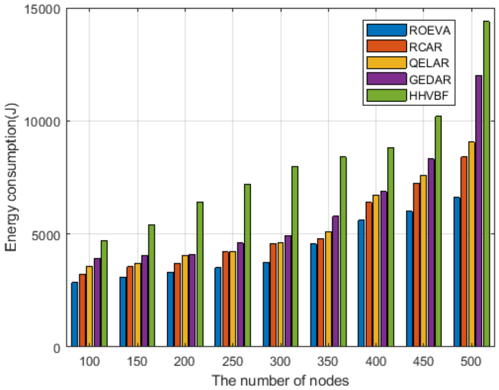


Fig. 12. Comparison of energy consumption.

Because of the harsh underwater environment and high deployment cost, batteries of sensor nodes are hard to be recharged or replaced. Therefore, energy is very important in UASNs. The routing protocol should find an energy-efficient routing path for delivering data packets to the sink node. The comparison in energy consumption is shown in Figure 12. The total energy consumption of a node in the simulation includes packet transmission and packet receiving. It can be seen from the figure that the energy consumption of all these protocols increased with the increase of sensor nodes. The occurrence of this is mainly due to the fact that more nodes are involved in data forwarding and more energy consumption for handling the data packets. We can observe that the ROEVA protocol consumes less energy than other protocols in general. Especially, when there are 500 nodes in the network, the energy consumption of ROEVA is 21.3%, 27%, 44.9% and 54.1% lower than RCAR, QELAR, GEDAR, and HHVBF, respectively. Since ROEVA uses opportunistic routing based on Reinforcement-Learning technology to forward packets and sets different waiting time for candidate nodes which reduce retransmissions, ROEVA consumes significantly less energy compared to HHVBF and GEDAR protocols. In addition, no extra overhead is needed for exchanging information with control packets and the two-hop availability checking method avoids void areas in advance, thereby more energy-efficient compared with RCAR and QELAR.

From Figure 9 to Figure 12, we evaluate ROEVA in average end-to-end delay, PDR, the average hop count of delivered packets, and energy consumption, and compared with RCAR, QELAR, GEDAR and HHVBF protocols in the four evaluation indexes. The simulation results show that the proposed ROEVA has excellent performance among the five protocols and is more balanced in these four aspects.

VI. Conclusion

In this paper, a reinforcement-learning-based opportunistic routing protocol for UASNs was proposed. The proposed protocol finds the optimal routing path to realize energy-efficient and reliable data packet transmission. In order to avoid void areas, the protocol performs two-hop routing availability checking before constructing candidate nodes, thereby identifying trap nodes and void nodes and reducing the retransmissions of routing. Furthermore, the ROEVA protocol improves network performance through applying the reinforcement-learning technique to the OR paradigm. To make a balance among several performance metrics, we design the reward function of Reinforcement-Learning by considering the delay, depth, packet delivery ratio, and residual energy. Moreover, a waiting mechanism for opportunistic routing was proposed, which can reduce collisions and retransmissions as well as to adapt to different scales of network architectures. The simulation results show that the ROEVA protocol embrace an excellent performance in total energy consumption, PDR, the average hop count, and average end-to-end delay, compared with RCAR, QELAR, GEDAR, and HHVBF in different scenarios with various node density.

References

[1] S. Nasir, N. Sharma, M. Jain, and P. Garg, “Underwater optical wireless communications, networking, and localization: A survey,” *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101935.

[2] N. Saeed, M.-S. Alouini, and T. Y. Al-Naffouri, “Toward the internet of underground things: A systematic survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3443–3466, 4th Quart., 2019.

[3] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, “Research challenges and applications for underwater sensor networking,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2006, pp. 228–235.

[4] G. Tuna, O. Arkoc, and K. Gulez, “Continuous monitoring of water quality using portable and low-cost approaches,” *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 6, Jun. 2013, Art. no. 249598.

[5] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, “Underwater Internet of Things in smart ocean: System architecture and open issues,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4297–4307, Jul. 2020.

[6] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, “A survey on MAC protocols for underwater wireless sensor networks,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1433–1447, 3rd Quart., 2014.

[7] S. Rajasoundaran *et al.*, “Secure routing with multi-watchdog construction using deep particle convolutional model for IoT based 5G wireless sensor networks,” *Comput. Commun.*, vol. 187, pp. 71–82, Apr. 2022.

[8] S. Lu, Z. Wang, Z. Wang, and S. Zhou, “Throughput of underwater wireless ad hoc networks with random access: A physical layer perspective,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6257–6268, Nov. 2015.

[9] J. Liu, Z. Wang, J.-H. Cui, S. Zhou, and B. Yang, “A joint time synchronization and localization design for mobile underwater sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 530–543, Mar. 2016.

[10] Y. Zhou, T. Cao, and W. Xiang, “QLFR: A Q-learning-based localization-free routing protocol for underwater sensor networks,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[11] T. Patel and P. Kamboj, “Opportunistic routing in wireless sensor networks: A review,” in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Jun. 2015, pp. 983–987.

[12] M. A. Rahman, Y. Lee, and I. Koo, “EECOR: An energy-efficient cooperative opportunistic routing protocol for underwater acoustic sensor networks,” *IEEE Access*, vol. 5, pp. 14119–14132, 2017.

[13] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, “Void-handling techniques for routing protocols in underwater sensor networks: Survey and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 800–827, 2nd Quart., 2017.

[14] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, “An inherently void avoidance routing protocol for underwater sensor networks,” in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2015, pp. 1–6.

[15] P. Xie, J. H. Cui, and L. Lao, “VBF: Vector-based forwarding protocol for underwater sensor networks,” in *Proc. Int. Conf. Res. Netw.*, Coimbra, Portugal, May 2006, pp. 1216–1221.

[16] H. Yan, Z. J. Shi, and J. H. Cui, “DBR: Depth-based routing for underwater sensor networks,” in *Proc. Int. Conf. Res. Netw.*, May 2008, pp. 72–86.

[17] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro, “Geographic and opportunistic routing for underwater sensor networks,” *IEEE Trans. Comput.*, vol. 65, no. 2, pp. 548–561, Feb. 2016.

[18] V. Menon, D. Midhunchakkaravarthy, S. John, S. Jacob, and A. Mukherjee, “A secure and energy-efficient opportunistic routing protocol with void avoidance for underwater acoustic sensor networks,” *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 4, pp. 2303–2315, Jul. 2020.

[19] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, “An underwater routing protocol with void detection and bypassing capability,” in *Proc. IEEE 31st Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2017, pp. 530–537.

[20] N. Javaid, A. Majid, and A. Sher, “Avoiding void holes and collisions with reliable and interference-aware routing in underwater WSNs,” *Sensors*, vol. 18, pp. 11–13, Sep. 2018.

[21] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, “VAPR: Void-aware pressure routing for underwater sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 895–908, May 2013.

[22] Z. Jin, C. Duan, Q. Yang, and Y. Su, “Q-learning-based opportunistic routing with an on-site architecture in UASNs,” *Ad Hoc Netw.*, vol. 119, Aug. 2021, Art. no. 102553.

[23] H. Luo, K. Wu, R. Ruby, Y. Liang, Z. Guo, and L. M. Ni, “Software-defined architectures and technologies for underwater wireless sensor networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2855–2888, 4th Quart., 2018.

[24] T. Hu and Y. Fei, “QELAR: A Q-learning-based energy-efficient and lifetime-aware routing protocol for underwater sensor networks,” in *Proc. IEEE Int. Perform., Comput. Commun. Conf.*, Dec. 2008, pp. 247–255.

[25] Z. Jin, Q. Zhao, and Y. Su, “RCAR: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks,” *IEEE Sensor J.*, vol. 19, no. 22, pp. 10881–10891, Nov. 2019.

[26] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, “Scalable localization with mobility prediction for underwater sensor networks,” in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 2198–2206.

[27] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, and Z. Lu, “3D underwater sensor network localization,” *IEEE Trans. Mobile Comput.*, vol. 8, no. 12, pp. 1610–1621, Dec. 2009.

[28] Z. Fang, J. Wang, C. Jiang, B. Zhang, C. Qin, and Y. Ren, “QLACO: Q-learning aided ant colony routing protocol for underwater acoustic sensor networks,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[29] L. M. Brekhovskikh, Y. P. Lysanov, and R. T. Beyer, “Fundamentals of ocean acoustics,” *J. Acoust. Soc. Amer.*, vol. 90, no. 6, pp. 3382–3383, Dec. 1991.

[30] A. Gosavi, “Reinforcement learning: A tutorial survey and recent advances,” *INFORMS J. Comput.*, vol. 21, no. 2, pp. 178–192, 2009.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. London, U.K.: MIT Press, 2015.

[32] A. Now? and T. Brys, “A gentle introduction to reinforcement learning,” in *Proc. Int. Conf. Scalable Uncertainty Manage.*, 2016, pp. 18–32.

[33] *The Network Simulator-Ns-3*. Accessed: Jan. 10, 2020. [Online]. Available: http://www.nsnam.org

[34] H. Yan, S. Zhou, Z. J. Shi, and B. Li, “A DSP implementation of OFDM acoustic modem,” in *Proc. 2nd Workshop Underwater Netw.*, Sep. 2007, pp. 89–92.

Rongxin Zhu received the M.E. degree in software engineering from Nanjing University, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree in cyberspace security with Hainan University, Haikou, China. His research interests include security of the underwater sensor networks and intelligent algorithm.

Qihang Jiang received the B.E. degree from the Zhengzhou University of Aeronautics, Henan, China, in 2021. He is currently pursuing the master's degree with Hainan University, Haikou, China. His research interests include protocols design and security of the underwater acoustics sensor networks.

Xiangdang Huang received the B.E. degree from Yanan University, Shaanxi, China, in 2002, and the M.E. degree from the University of Electronic Science and Technology of China, Sichuan, China, in 2012. He is currently an Associate Professor and the Master Tutor with Hainan University, Haikou, China. His research interests include protocols design and security of the underwater acoustics sensor networks.

Deshun Li received the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2017. He is a Lecturer at the School of Computer Science and Technology, Hainan University, Haikou, China. His current research interests include data center networks (DCNs), the Internet of Things (IoT), network security, and graph theory.

Qiuling Yang received the B.E. degree from Shenyang Aerospace University, Shenyang, China, in 2003, the M.E. degree from Guangxi University, Nanning, China, in 2010, and the Ph.D. degree from Tianjin University, Tianjin, China, in 2016. She is currently a Professor and the Doctoral Supervisor with Hainan University, Haikou, China. Her research interests include protocols design and security of the underwater acoustics sensor networks.