IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.DOI

DQELR: An adaptive Deep Q-Network-based energy- and latency-aware routing protocol design for underwater acoustic sensor networks

YISHAN SU[1], RONG FAN[1], XIAOMEI FU[2], AND ZHIGANG JIN[1]

[1]Department of Electrical and Information Engineering, Tianjin University, Tianjin (yishan.su@tju.edu.cn; rongfan@tju.edu.cn; zgjin@tju.edu.cn)
[2]Department of Marine Science and Technology, Tianjin University, Tianjin (fuxiaomei@tju.edu.cn)

Corresponding author: Zhigang Jin (zgjin@tju.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 61701335, 61862020, 61861014, 61571323 and 61571318, in part by the Science and Technology on Underwater Information and Control Laboratory under Grant 614221801050517, in part by the Natural Science Foundation of Tianjin under Grant 17JCQNJC01300 and in part by the Key Research and Development Plan of Hainan under Grant ZDYF2018006.

ABSTRACT Underwater acoustic sensor networks (UASNs) have become a popular research topic, with research challenges focused on underwater communication techniques. By incorporating long end-to-end latency, high energy consumption and dynamic network topology in UASNs, many intelligent routing protocols have been proposed to solve the problem. However, shortcomings still exist, and comprehensive routing protocols are urgently needed. In this paper, we propose an adaptive Deep Q-Network-based energy- and latency-aware routing protocol (DQELR) to prolong network lifetimes in UASNs. In the DQELR, a Deep Q-Network algorithm with both off-policy and on-policy methods is adopted to make globally optimal routing decisions. Based on both the energy and depth states of nodes at different communication stages, nodes with the maximum Q-value can be selected as forwarders adaptively considering both energy and latency. A hybrid of the broadcast and unicast communication mechanisms is also designed to reduce network overhead. In addition, network topology changes can be addressed through an on-policy method that makes a new routing decision when the current route becomes corrupted. With less energy consumption and strict latency limitations, the DQELR can prolong network lifetimes in UASNs. Simulation results show that the DQELR can achieve a superior network lifetime with better latency and energy efficiency performances relative to other general schemes applied in UASNs.

INDEX TERMS Underwater acoustic sensor network, routing protocol, Deep-Q Network.

I. INTRODUCTION

Underwater acoustic sensor networks (UASNs) have recently attracted great interest in terms of increasing scientific research and commercial development activities along with the enhancement of military defense in the oceans [1]. Unlike terrestrial sensor networks, due to the rapid attenuation of radio waves, UASNs communicate through acoustic signals. The propagation speed of acoustic signals in water is approximately 1.5×10^3 m/s, which is five orders of magnitude lower than the speed of radio waves (3×10^8 m/s) [2]. In addition, there exist various uncertain factors in underwater environments, which may change the network topology and affect link connectivity [3]. Moreover, due to the large volume, high energy consumption and difficulty in battery replacement for underwater sensor nodes, limited resources restrict network lifetimes [4], which is defined as the time span from the deployment of a network to the instant when the network is considered nonfunctional [5]. In this paper, we adopt the time when the first sensor node dies to define the network lifetime. Because of the unique characteristics of underwater communication, efficient protocols are urgently required to ensure reliable communication over UASNs.

Over the past few decades, a variety of routing protocols have been proposed for terrestrial wireless sensor networks. However, the challenges of long end-to-end latency, high energy consumption, dynamic network topology and short

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

network lifetime in UASNs mean that the principles of terrestrial sensor networks cannot be directly applied to underwater communication. In terrestrial communication, routing mechanisms are generally divided into three types, including the proactive routing protocol, the passive routing protocol and the geographical routing protocol [6]. In the proactive routing protocol, each node establishes and maintains a routing table, which reports the routing information of that node to all other nodes in the network. The routing table needs to be updated in real time according to the current state of the network. However, a dynamic network topology can lead to considerable network overhead in the establishment and maintenance of routing tables, which may significantly decrease the lifetimes of UASNs with limited resources, as in the Optimized Link State Routing (OLSR) protocol [7]. In the passive routing protocol, the node establishes the route temporarily according to the communication demand, which greatly reduces the network overhead. Compared with active routing, it is more suitable for a network with a dynamic topology. However, the processes of route discovery and establishment are unpredictable, which makes end-to-end latency more volatile and difficult to predict, such as in the ad hoc On-Demand Distance Vector (AODV) routing protocol [8]. Both of the protocols mentioned above rely on information flooding, which may produce large network overhead in UASNs. In the geographical routing protocol, the routing is mainly based on geographical location information from the nodes in the network, such as GFG [9]. Although a global positioning system (GPS) radio receiver can estimate the geographic locations of sensor nodes in terrestrial systems, it cannot work properly in underwater environments.

To address these shortcomings, a great deal of research has been focused on protocol design applicable to UASNs. Typical routing protocols adopt shortest path algorithms to make routing decisions, which can shorten end-to-end latency and reduce energy consumption. However, some hot nodes on frequently used shortest paths can be drained quickly, resulting in a shorter network lifetime. To prolong the network lifetime, some intelligent algorithms have been proposed. These routing protocols select forwarding nodes according to residual energy. As a result, the node with more residual energy is chosen as the next hop. However, these algorithms do not consider end-to-end latency, energy consumption, node mobility and globally optimal paths at the same time as lifetime. To choose the node with more residual energy, some packets deliver from the source to the sink with more hops, which increases end-to-end latency. In addition, information interactions consume significant energy in the process of planning the route. Moreover, node movement is not considered in some protocols. Furthermore, some routing paths will fall into local optima because a greedy algorithm is applied to determine the next hop, which may not be globally optimal. In UASNs, the end-to-end latency, energy consumption, node mobility and network lifetime are all important. The protocols mentioned above prolong network lifetime without considering the other crucial indicators. Therefore, comprehensive routing protocols are urgently needed to improve network communication performance in UASNs.

In this paper, we propose an adaptive Deep Q-Network-based energy- and latency-aware routing protocol (DQELR) that comprehensively considering the crucial indicators mentioned above to prolong network lifetime in UASNs. The Deep Q-Network (DQN) technique [10] combines a deep neural network with Q-learning (QL), which adopts Q-values as the rewards for making decisions in some policy scenarios. While lowering energy consumption and strictly limiting communication latency, the network lifetime can also be extended by selecting forwarding nodes with relatively high residual energy. In order to comprehensively consider the important multi-metrics, including end-to-end latency, energy consumption, globally optimal paths, node mobility and network lifetime in underwater acoustic communication, we adopt Deep Q-Network in DQELR to take all these metrics into consideration. To solve a complex multi-state decision problem with too many states including node information and network topology in UASNs, a DQN technique with saving space and time has better adaptability to make routing decisions. By learning the environment, DQN can give an action-value function (Q-value), which is the reward for taking an action in a given state. In the DQELR, nodes with the residual energy and depth information of themselves and their neighbors can select the next forwarding nodes automatically using the DQN technique, and the network lifetime can be extended both considering end-to-end latency and energy consumption. Furthermore, by considering both the instant rewards and the discounted long-term rewards, the routing decision made by Deep Q-Network can be made globally optimal. In addition, for dynamic network topology, an asynchronous method in Deep Q-Network is designed to update parameter of DQN and make routing decision again with changed topology.

The following are the main contributions to the subject:

- We adopt DQN as the main technique in a routing protocol to prolong the network lifetime in UASNs. To solve a complex multi-state decision problem in UASNs, DQN is more applicable to the routing protocol than is the QL technique in [11]. The DQELR performs well in terms of energy efficiency, end-to-end latency and network lifetime in UASNs.
- In UASNs, information interaction through broadcast among sensor nodes consumes a great deal of energy. Our proposed DQELR protocol adopts a hybrid of the broadcast and unicast communication mechanisms that results in less energy consumption compared to the flood broadcast protocol.
- The DQELR protocol can prolong the network lifetime by making routing decisions according to residual energy under the premise of strictly limiting end-to-end latency.
- In UASNs with dynamic network topologies, the DQELR can use on-policy training to update Q-values

and correct network parameters for new routing decisions, which can guarantee reliable communication.

- The DQELR is extensible. In addition to considering energy and the depth of nodes, other factors such as node density and environmental noise can be incorporated into the reward function to deal with various communication situations and targets.

In conclusion, compared to an existing intelligent routing algorithm based on reinforcement learning, the DQELR performs well in terms of energy efficiency and latency as well as improving the network lifetime by 34-36%.

The remainder of this paper is organized as follows. Firstly, we analyze the existing routing protocols in UASNs in section II. Then, the basic Deep Q-Network technique is introduced in section III. Next, DQELR protocol is described in section IV, and the simulation results are presented with analyzation in section V. Finally, we conclude the paper in section VI.

II. RELATED WORK

In this section, we summarize our literature review on the developments and status of routing protocols in UASNs.

In a hash underwater environment, sensor nodes are powered by batteries that have limited energy and are difficult to replace. Therefore, many routing protocols have been focused on improving energy efficiency to prolong the network lifetime. In [12], Xie, P et al. propose a vector-based forwarding (VBF) routing protocol that uses location information to improve energy efficiency in highly dynamic networks. In VBF, packets are delivered within the range of a routing pipe; nodes that fall within this range will be used more frequently than those outside the range, leading to an imbalance in energy consumption and a shorter network lifetime. To solve this problem, the hop-by-hop vector-based forwarding (HH-VBF) routing protocol [13] and adaptive hop-by-hop vector-based forwarding (AHH-VBF) routing protocol [14] have been presented and have achieved high data delivery, low energy consumption and short latency, although they lead to heavier overhead than does VBF. In [15], Yan, H et al. propose a depth-based routing (DBR) protocol that reduces energy consumption and the number of collisions. Adopting a greedy algorithm, packets are transmitted from the source to the sink in an upward direction based on depth information. However, if there is no sink near the node, the packets carried by that node will be dropped. In addition, packets can only be transmitted from bottom to top, which does not meet the demand of the sink to transmit data download. Based on the DBR protocol, the fuzzy depth-based routing (FDBR) protocol [16] selects the forwarding node according to hop numbers, depth and energy information, which improves the energy efficiency and end-to-end delay performance of the network.

Due to the low propagation speed of acoustic signals in UASNs, high latency is also a challenge. In [17], a low propagation delay multipath routing (MPR) protocol is proposed to reduce latency. In the MPR, data packets are divided into several time slots by the source nodes based on bandwidth and a two-hop transmission scheme is adopted to deliver data packets to the relay nodes. The MPR uses multiple paths during the path construction from the source to the destination to avoid data collisions. The disadvantage of the MPR is that its energy consumption is much higher than that of other protocols due to heavy overhead. In [18], Bzoor, M.A et al. propose an adaptive power controlled routing (APCR) protocol. In the APCR, sensor nodes are divided into different power levels and can adjust their transmission power according to network conditions. Relay nodes can be chosen based on the power level and residual energy of the sensor nodes. Thus, the APCR can achieve a high delivery ratio and reduce delay in both sparse and dense networks. However, there is no limit on the number of relay nodes, which leads to high energy consumption when several nodes forward the same packet.

Because a dynamic network topology affects link connectivity, many researchers have taken node mobility into consideration when designing routing protocols. In [19], a pressure routing HydroCast protocol is proposed. In a hash underwater environment where nodes move continuously, HydroCast takes channel quality into consideration to improve routing performance. However, the energy consumption of the nodes is not considered. In [20], void aware pressure routing (VAPR) is also proposed for dynamic topology underwater networks to improve routing robustness. The relay nodes can be chosen based on sequence numbers, depth information, hop counts and forwarding directions. However, the protocol does not consider the residual energy of the nodes, which is crucial for prolonging the network lifetime.

Although traditional routing protocols can improve network performance well, they are still limited by their capacities to deal with multiple constraints and high calculation complexity [21]. In recent years, many intelligent algorithm-based routing protocols have been proposed for terrestrial wireless sensor networks, but few of those protocols have been applied in UASNs to prolong network lifetimes. Among these intelligent algorithms, fuzzy logic-based routing protocols have been applied in UASNs. The cluster-based routing algorithm (CBRA) protocol [22] relies on a fuzzy logic method similar to low energy adaptive clustering hierarchy (LEACH) protocol that evenly distributes loads in large scale networks. In [23], a distance based reliable and energy efficient (DREE) routing protocol is also proposed based on a fuzzy logic algorithm. The DREE adopts a fuzzy logic system to determine the best routing path with distance, residual energy and link quality as the system inputs; it outperforms traditional routing protocols in terms of network lifetime. However, the end-to-end latency in these algorithms is relatively longer. In [24], Z. M. Zahedi et al. propose a swarm intelligence based fuzzy (SIBF) routing protocol for clustered wireless sensor networks. Unlike the DREE, the distances between the sensor nodes and the cluster head, rather than link quality, are used as the inputs for choosing relay nodes. As a result, the network lifetime is prolonged.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

However, packet delivery, end-to-end latency and energy consumption are not considered. Furthermore, clustering-based routing protocols have also been applied in UASNs. In [25], a novel nature-inspired evolutionary link quality-aware queue-based spectral clustering routing protocol (LR-P) is proposed with better performance than existing routing protocols in terms of data delivery ratio, overall network throughput, end-to-end delay, and energy efficiency. In [26], a genetic algorithm-based routing protocol, namely quality-of-service (QoS) aware evolutionary routing protocol (QER-P) is proposed to achieve low network delay, high packet delivery ratio, and low energy consumption in UWSNs with highly stable and reliable clustering and routing mechanisms. In addition to clustering algorithms, reinforcement learning-based routing protocols are also a popular research topic. In [11], a Q-learning-based adaptive routing (QELAR) protocol is proposed for energy-efficient and lifetime-extended underwater sensor networks. The QELAR makes routing decisions according to a reward function that considers energy. Nodes with higher residual energy can be chosen as subsequent hops, thus prolonging the network lifetime. However, the nodes in the network need to learn the surrounding environment through meta data packet interactivity to determine the next forwarding node, which results in high energy consumption. In addition, the QELAR does not strictly control network latency when choosing the node with more residual energy. Moreover, routing algorithms that rely on instant rewards can be trapped at local optima rather than finding global optima. In [27], a Q-learning-based delay-aware routing algorithm (QDAR) protocol is proposed. Comparing with QELAR, propagation delay is considered for the purpose of restricting latency to prolong the network lifetime. However, in the QDAR, the source-initiated query phase and interest phase, owing to their need to send routing path information to the source, will result in high energy consumption and a long propagation delay, which weakens the network performance.

III. DEEP Q-NETWORK TECHNIQUE

In this section, we briefly introduce the major technique adopted by DQELR from Q-learning to Deep Q-Network.

A. Q-LEARNING TECHNIQUE

Q-learning is one of the main reinforcement learning (RL) techniques used in machine learning. RL is a method concerned with optimizing the actions of autonomous agents in their environment to maximize rewards. QL has the same goal as RL but does not need an environmental model and can handle problems with random transitions and rewards without requiring adaptations.

The QL process can be seen as a Markov decision process (MDP). The current state and the selected action of the agent determine a fixed state transition probability distribution, the next state, and an immediate reward. Fig. 1 shows the basic framework of Q-learning. In QL, the agent selects an action under the environment in the current state. Then, the environment feeds back a reward signal to the agent after

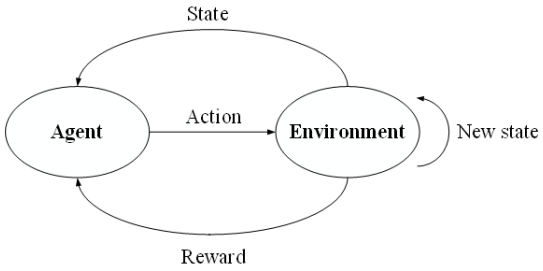

FIGURE 1. The basic framework of Q-learning.

accepting the action, with the current state updating to the new state. According to the reward signal, the agent then selects and executes the next action. Generally, the environment is random, which means that the next state is also random. Therefore, the set of states and actions, together with the rules for changing states, constitute a Markov decision process. An episode in this process forms a limited sequence of states, actions, and rewards in (S, A, R) :

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, ..., s_{n-1}, a_{n-1}, r_n, s_n. \quad (1)$$

where s_i denotes the state, a_i denotes the action, r_{i+1} denotes the reward obtained after performing the action a_i, and S, A, R denote the set of states, actions and rewards, respectively. The episode ends when the final state moves to s_n. A MDP is based on the Markov assumption that the probability of the next state s_{i+1} depends on the current state s_i and action a_i, rather than the previous state and action.

The learning goal of the agent is to maximize the cumulative value of future rewards. For a given run of the MDP, the total reward for an episode, considering both immediate and future rewards, can be calculated by the following formula:

$$R = r_1 + r_2 + r_3 + ... + r_n. \quad (2)$$

However, because of the randomness of the environment, the reward will change after the next same action. As time goes on, the error accumulates. Therefore, a factor $\gamma \in [0, 1]$, which represents the extent to which time affects rewards, is used to discount future rewards. The total reward can be written as

$$R_i = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + ... + \gamma^{n-i} r_n. \quad (3)$$

In QL, the true value of an action a_i in a state s_i under a given policy π is expressed as a Q-value. According to Eq. (3), the Q-value can be derived as follows:

$$Q_\pi(s_i, a_i) = R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_*(s_{i+1}, a_{i+1}). \quad (4)$$

where $R(s_i, a_i)$ represents the reward of action a_i in state s_i, $P(s_i, a_i, s_{i+1})$ represents the probability of switching to state s_{i+1} after action a_i in state s_i, and $Q_*(s_{i+1}, a_{i+1}) =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

$\max Q_\pi(s_{i+1}, a_{i+1})$ represents the optimal Q-value of action a_{i+1} in next state s_{i+1} . Then, update Q-value by the following formula:

$$Q_\pi(s_i, a_i) \leftarrow Q_\pi(s_i, a_i) + \alpha \times \left[R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_*(s_{i+1}, a_{i+1}) - Q_\pi(s_i, a_i) \right]. \tag{5}$$

where $\alpha \in (0, 1]$ is learning rate. From Eq. (5), the optimal action a_i can be obtained as follows:

$$a_i = arg\,max\ Q_\pi(s_i, a_i). \tag{6}$$

Therefore, the optimal policy can be derived from the optimal action.

B. DEEP Q-NETWORK TECHNIQUE

QL uses a table to store Q-values, but it does not work well when there are too many states. The reason is that not only does the establishment of a Q-table take up a great deal of space, but online search based on the Q-table also takes up a great deal of time. Differing from QL, a Deep Q-Network can use parameter fitting to construct a function to predict Q-values through a combination of QL and neural networks, which saves space and time. The DQN is essentially a neural network in which, for any given state, the Q-network can output a vector of action Q-values.

The key of the DQN is to train the neural network. First, we construct a loss function. Using the mean square deviation to define the target function, we obtain the following loss function:

$$L(w) = E\left[(TargetQ - Q_\pi(s_i, a_i, w))^2 \right]). \tag{7}$$

where

$$TargetQ = R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_*(s_{i+1}, a_{i+1}, w).$$

w is the network parameter and the Q-value to be updated from QL is used as $TargetQ$. Q_π is a predicted Q-value.

Then, we calculate the gradient of the parameter w about the loss function:

$$\frac{\partial L(w)}{\partial w} = E\left[(TargetQ - Q_\pi(s_i, a_i, w)) \frac{\partial Q_\pi(s_i, a_i, w)}{\partial w} \right]. \tag{8}$$

where $w = w + \eta \frac{\partial L(w)}{\partial w}$ and $\eta \in [0, 1]$.The process of network training consists of updating w until the loss function converges. Let $P(s_i, a_i, s_{i+1}) = 1$ for the sake of simplicity. However, the DQN replaces the function of estimating Q-values in QL with a deep neural network, which leads to the instability of the algorithm. The main reason is that there is a strong correlation between continuous state and action inputs; thus, small updates to the Q-value of a signal action will change the entire network and affect the Q-value of each action in other states, which will affect the distribution of the sampling data. In [28], Mnih et al. propose two important methods that use a target network and experience replay to solve this problem. In experience replay, each step of the experience is stored in a database and sampled subject to a uniform distribution. In the target network, the parameters are synchronized every few steps with those of the online network. These two methods improve the performance of the DTN algorithm.

IV. DEEP Q-NETWORK-BASED DQELR PROTOCOL

In this section, we describe the DQN-based DQELR protocol, including the entire DQELR protocol mechanism, the routing decision algorithm, and a consideration of dynamic network topology.

A. PROTOCOL OVERVIEW

Based on DQN technique, the DQELR protocol applied in UASNs can be explained as follows. In a UASN, source nodes are deployed underwater to send collected data packets to sink nodes on the surface through relay nodes. Each packet in the network comprises an agent; the current information of the sensor node, such as its residual energy, depth and neighboring nodes, comprises the current state; and the forwarding of a packet from one node to the next node in the current state comprises an action. After the node sends the packet to one of its neighbors, it can receive a reward, with the current state of each node updating to a new state. Based on the reward, the agent can make a routing decision under the policy π . In the DQELR protocol, each node can obtain the information necessary to calculate all the Q-values through broadcast with its neighbor nodes; thus, the agent can make an optimal routing decision directly when it needs to send a packet, which reduces the energy consumption substantially.

B. DQELR PROTOCOL MECHANISM

In this section, we introduce the entire DQELR protocol mechanism and design the packet format for the DQELR.

1) The DQELR protocol mechanism

We first briefly introduce the DQELR, which provides a complete execution rule for data packets transmitted from the source to the sink. In the DQELR, a hybrid of the broadcast and unicast communication mechanisms is adopted in which broadcast communication is used for information updating among sensor nodes and unicast communication is used for data transmission. When making routing decisions, we adopt the DQN algorithm with both the off-policy and on-policy methods applied under different network conditions. Here, off-policy and on-policy mean training the neural network offline and online, respectively, In addition, in the case of a dynamic topology in a UASN, the DQELR adopts an on-policy method to make a new routing decision when the current route is corrupted.

Assumptions in UASN we deployed are made as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

1. Sensor nodes in the acoustic sensor network we deployed can get their own residual energy and depth information.

2. The topology of the acoustic sensor network is changeable but remains relatively stable in the short time.

3. The computational delay of Q-values and power consumption of computation of the sensor nodes are much lower than those involved in acoustic communication [29]. Therefore, it can be neglected.

Then, we describe the mechanism of the DQELR in detail. Before deploying sensor nodes underwater, the states of the nodes and the network topology are assumed to be known, including the residual energy, deployment depth rather than three-dimensional geographic information, and the neighbors of nodes with the same information. At this stage, the DQN algorithm adopts an off-policy method to train the neural network to make routing decisions. Next, the sensor networks are deployed underwater according to the assumed states of the nodes and the network topology in the off-policy stage. When the DQN algorithm runs underwater, the network topology and the states of the nodes can be changed and an on-policy method is adopted to train the neural network to make routing decisions based on the full procedure stored in the experience pool in the off-policy stage. Because the nodes are transmitting data packets continuously, the residual energy of the nodes will be reduced and the depth and neighbors of the nodes can be changed owing to water flow. The nodes need to obtain this information as inputs for the neural network to produce the next optimal forwarder. However, it is costly for a node to obtain the information before sending data packets. Thus, we adopt a broadcast communication mechanism to allow interaction between nodes and their neighbors. When a node needs to send a data packet, with information and calculated Q-values of all the neighbors held already, it can directly deliver the packet to the next optimal forwarder using the unicast communication mechanism with the maximum Q-value. Because the DQN is stable and the topology of UASNs remains relatively stable in the short term, information updating among nodes can be done periodically according to the actual conditions of the underwater environment to reduce network overhead. In this paper, we update information through broadcasting based on the packet generation rate. In addition, when the chosen optimal forwarder moves out of the communication range because of the changeable topology, a node with a suboptimal Q-value can be chosen as the forwarder; when the chosen optimal forwarder moves into the communication range, the node with the optimal Q-value can still be selected as the forwarder. Then, the nodes with changed neighbors adopt an on-policy method to update the Q-values and correct the network parameters to make a new routing decision based on the changed topology the next time they send a packet.

2) Packet format

There are two types of packet formats for updating information and data transmission, as shown in Fig. 2.

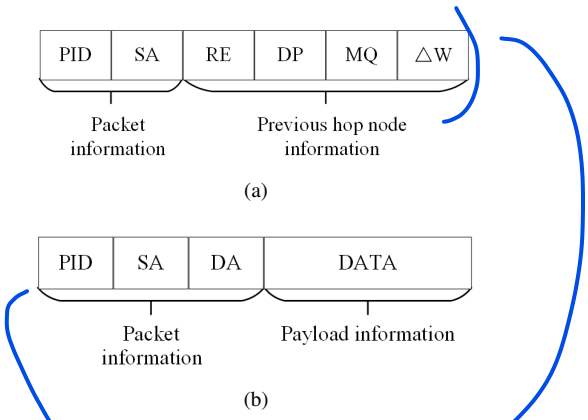

FIGURE 2. (a) Packet format for updating information; (b) Packet format for data transmission.

For updating information, as in Fig. 2(a), the packet format consists of the packet ID (PID) and source address (SA), the residual energy (RE), the depth (DP), the maximum Q-value (MQ) with the optimal next forwarder ID and the parameter variation Δw of the neural network, where Δw represents the gradient of the parameter w about the loss function. In the packet information format, the PID gives the unique identity of the packet with the type embedded, and the SA labels the address of the node that produces packets. The information in these parts is permanent. In the previous hop node information format, RE, DP, MQ and Δw are needed for neural network training and parameter updating, and these parts of the information can be changed in the course of packet transmission.

For data transmission, as in Fig. 2(b), the destination address (DA) is added to the packet information format to label the destination to which the packet should be delivered. In this format, data payload information (DATA) replaces the previous hop node information. Because the optimal next forwarder of each node has been calculated in the information updating stage, the packet only needs to carry the packet information and data information in the data transmission stage. It can be seen that this information needs to be carried by the packet in DQELR is the same as that of other routing protocols, and the transmission efficiency of DQELR is no lower than that of the other routing protocols.

To make a routing decision, a sensor node needs to store the information defined in Fig. 3, consisting of its own and neighbor information. The information of the sensor node itself includes the residual energy, the depth, the Q-values with all the neighbors and Δw . The neighbor information includes the updated information from all the neighbors in the data interaction stage.

In addition, the sensor nodes need to calculate Q-values to make routing decisions. According to the DQN algorithm, the computational content can be derived as follows. In the neural network, there are four parameter matrices w_1, w_2, w_3, w_4 and two bias matrices b_1, b_2 used to construct the Q-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

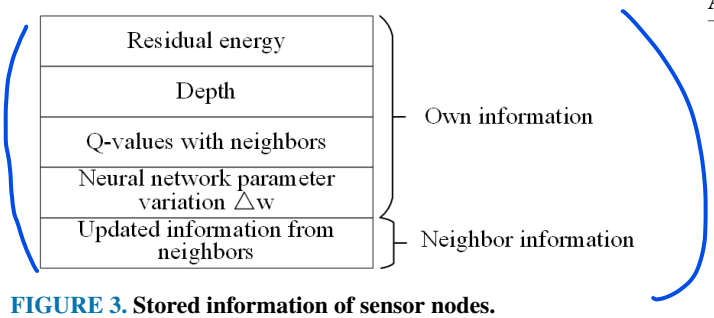

FIGURE 3. Stored information of sensor nodes.

value function. With the input information x_1, x_1 of the node and its neighbor, the Q-value under the action of forwarding a packet from the node to its neighbor can be calculated through the following formula:

$$Q = f(xW+b) = f([x_1, x_2]\,[w_1 \times w_2 \times w_3 \times w_4] + [b_1, b_2]). \tag{9}$$

The sensor nodes need to store the parameters and the calculation rule.

By the above description, we know that the overhead of the DQELR arises mainly from the interaction and storage of the information held by the node itself and its neighbors, including the residual energy, the depth, the Q-values with all the neighbors, Δw and the updated information from the neighbors. Each piece of information can be represented by a 16-bit word and the storage overhead can be ignored [11]. However, the interaction overhead is still existing although it is small. In order to reduce the overhead, the information updating stage which causes the overhead is designed to occur periodically according to the packet generation rate instead of occur in real time. Because the DQN is stable and the topology of UASNs remains relatively stable in the short term, information updating designed periodically still enables nodes to make correct routing decisions and reduces the overhead significantly. Thus, this part of overhead is negligible. Furthermore, the ID of the packet, the source, the forwarder and the destination can each be represented by a byte. All these corresponding overheads will be no more than 10 bytes [11]. In addition, the sensor nodes need to calculate Q-values when updating information. Because the updating does not occur in real time, the computation is simple and the computational delay and power consumption are much lower than those involved in acoustic communication [29]; thus, the computational overhead can also be ignored.

C. DQN-BASED ROUTING DECISION ALGORITHM

In this section, we describe the DQN-based routing decision algorithm, including its neural network structure, reward and loss function, off-policy training and on-policy training. Algorithm 1 provides the process of the DQN-based routing decision.

Algorithm 1 DQN-based routing decision algorithm

Step 1. Feature extraction of inputs.
Step 2. Initialize network parameters, rewards and Q-values.
Step 3. Off-policy training.
Adopted before deploying sensor nodes underwater: nodes with initial states can choose the forwarders with the maximum Q-values.
Step 4. On-policy training.
Adopted after deploying sensor nodes underwater: a hybrid of broadcast and unicast communication mechanisms is used; nodes with changeable states can choose the forwarders with the maximum Q-values.
If topology change
  For current update time **do**
    Nodes with neighbors reduced: choose the forwarders with the second maximum Q-values (and so on for other special cases); nodes with neighbors increased: choose the forwarders according to the existing maximum Q-values.
  End for
  For next update time **do**
  Update Q-values and correct network parameters to make new routing decisions with changed topology.
  End for
End if

1) Neural network model

We first determine the input of the neural network as in Fig. 4. Each packet in the network serves as an agent, and the agent at any sensor node can obtain Q-values of all its neighbors and choose the optimal one among them as the forwarder. Before the agent makes a routing decision, the initial data obtained through sampling from the neighbors need to be preprocessed as the input for the neural network. This process is called feature extraction. In this algorithm, when a node is calculating Q-values with all its neighbors, on the premise of having obtained the residual energies and depths of the neighbors, it first deals with the residual energy and depth in the form of r_{sen} and r_{dep} , respectively, as shown in Eqs. (10) and (11):

$$r_{sen} = \frac{e_{own}}{e_{\max}}. \tag{10}$$

$$r_{dep} = \frac{d_{own}}{d_{\max}}. \tag{11}$$

where e_{own} and d_{own} are the residual energy and depth of a neighbor node, respectively, and e_{max} and d_{max} are the maximum residual energy and maximum depth of all the neighbors, respectively. For example, sensor node A has three neighbors B, C, and D with residual energies and depths of 1000 J, 800 J, 800 J and 500 m, 500 m, 800 m respectively. According to the data processing rule, the maximum residual energy is 1000 J; the maximum depth is 800 m; and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

information B, C, D carry can be expressed as 1, 0.8, 0.8 and 0.625, 0.625, 1 respectively. This set of information can be represented by a tuple; for example, the information from A to B can be written as (A, 1.0, 0.625, B).

Then, a Multi-Layer Perception (MLP) model is constructed as the neural network for DQN, where the input of the MLP model is the tuple described above, which represents a set of states and actions. The MLP neural network model is shown in Fig. 4.

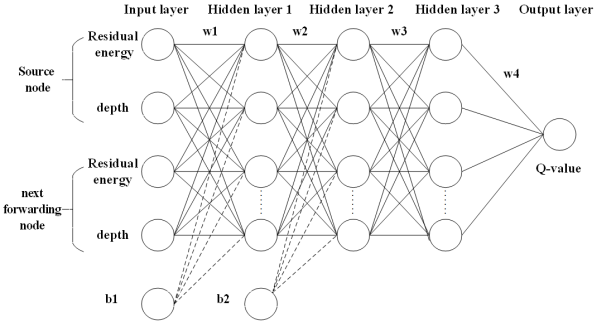

FIGURE 4. MLP neural network model.

In the MLP model, the input layer consists of four nodes, including the residual energy and depth of the source node sending the packet and the next node forwarding the packet, respectively. Hidden layer 1 has 300 nodes with bias b1, hidden layer 2 has 150 nodes with bias b2, hidden layer 3 has 15 nodes and the output layer is a Q-value. In addition, there is a weight parameter between every two layers. By way of the full connectedness of each layer, a four-dimensional input vector can be mapped and compressed into a one-dimensional output vector, which represents the Q-value corresponding to the set of input states and actions. There are three commonly used activation functions to better fit a neural network; these are the tanh, sigmoid and RELU functions. The function images and expressions are shown in Fig. 5. In the first two layers of the MLP model, we add a tanh activation function to each output layer, namely, the hyperbolic tangent function. The function value of tanh is always between 1 to -1. The value changes less when the input is farther from the origin and changes larger when the input is closer from the origin, which has a positive effect on feature processing. The sigmoid and RELU functions are not sensitive to negative values, so the tanh function is the ideal activation function to fit the network.

2) Reward and loss function

Eq. (7) provides the loss function of DQN, where a reward function $R(s_i, a_i)$ is produced at each set consisting of a state and an action. In this algorithm, when a node transmits the packet to the sink, the reward is set to 100; otherwise, the reward is set to

$$R(s_i, a_i) = c + \alpha r_{sen} + \beta r_{dep}. \tag{12}$$

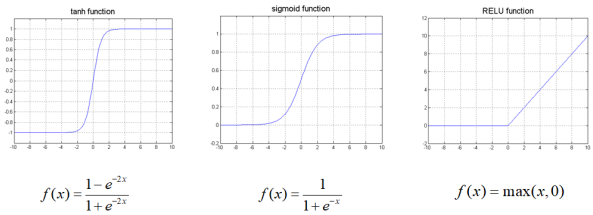

FIGURE 5. Tanh, sigmoid and RELU activation functions of the neural network.

where α and β denote the parameters of residual energy and depth, respectively, the value of c is equal to the sum of the values of α and β, which is much less than 100. Then, according to the loss function in Eq. (7), the neural network is iterated and converges through training. Because the reward of the sink with 100 is greater than those of the other nodes between 0 and c, the reward of 100 plays the main role when returning the reward; thus, the node prefers to choose a route closer to the sink. As a result, the number of detours is reduced when avoiding nodes with low residual energy, and nodes with high residual energy and high depth still serve as the selection criteria for each routing decision.

3) Off-policy training

Before deploying the sensor nodes underwater, we adopt an off-policy method to train the neural network. First, in order to achieve rapid convergence of neural network, the network parameters previously defined are initialized according to experience: γ is set to 0.9, w_1, w_2, w_3, w_4 are randomly initialized with normal distributions, two bias matrices b_1, b_2 are set to 1. With approximately 1300 tuples of states and actions among 80 sensor nodes and one sink, which will be introduced in the next section, 1300 rewards and Q-values are also initialized. The loss function $L(w)$ to fit is in Eq. (7) and the method of gradient descent is used in Eq. (8) to fit the loss function. For each training time of the neural network with initialized parameters, a Q-value is randomly selected from the 1300 Q-values and the optimal route is obtained under the initial network topology and the states of the sensor nodes, including the residual energies and depths.

Because DQN is developed from QL with the off-policy method, this method can also be extended to the DQN. However, the Q-values used to calculate the target Q and predicted Q in QL are the same and the new Q-value can be updated immediately when training the network, which is not suitable for the DQN. In the DQN, because a continuous process is associated with each node, if the network parameters are updated immediately, the Q-value of the current tuple will not be applied to the next tuple, which does not meet the basic requirements of a neural network with noncorrelation and leads to the nonconvergence of the parameters. Therefore, the loss and new Q-values need to be stored in an experience pool to be updated at an appropriate time.

4) On-policy training

When sensor nodes are deployed underwater, an on-policy method is adopted to train the neural network based on the full procedure stored in the experience pool in the off-policy stage. Because the energy consumption of each sensor node is far lower than the initial energy we set in the later section, online training can be performed periodically with a low computational overhead. At this stage, we use an asynchronous method in the one-step DQN in [30] to train the network. The main idea of this asynchronous method is that when the loss value $L(w)$ of a tuple is obtained, we do not update the network parameter w immediately, but store the loss gradient $\frac{\partial L(w)}{\partial w}$ and update w after a certain period through $w = w + \eta\frac{\partial L(w)}{\partial w}$, which meets the requirement of noncorrelation among the inputs of a neural network. With the asynchronous method, multi-nodes can be multithreaded to complete the communication task. For example, suppose there are several nodes ready to send packets and a series of loss gradient storage serves as a thread of them. These nodes will hold the gradients for a certain period rather than updating immediately because the communication among nodes does not occur in real time. If one of the nodes modifies the network parameters immediately, it is bound to affect the other nodes, but the modification cannot be transferred to other nodes in a timely manner. Thus, it is not advisable to modify the parameters independently. The best method is to store the loss gradient for a period and update concentratively according to the experience playback. The asynchronous strategy can unify the network parameters and speed up network convergence, and [30] has proved through experimentation that it results in faster convergence in the one-step DQN model.

As described in the previous section, in the on-policy training stage, broadcast communication is adopted for information updating and unicast communication is adopted for data transmission, which reduces the network overhead significantly. Because of the stability of the DQN, a slight lag in the network parameter updating can be ignored.

D. DISCUSSION FOR DYNAMIC NETWORK TOPOLOGY

In this section, we analyze the case of the topology changing in the network. A case of topology change is shown in Fig. 6, in which node E moves out of the communication range of D and is added to that of node B.

In the current scenario, D is ready to send the packet and E is the optimal forwarder of D with a maximum Q-value Q_{DE} between D and E in the last stage. Because E moves out of the communication range of D, Q_{DE} cannot be inherited and D will choose the neighbor with the second highest Q-value as the forwarder. Now, the maximum Q-value of D is updated and written as Q_D . Although E appears to be within the communication range of B, B has not obtained the information of E and is unable to select E as the forwarder. After the network updates the information once again through a broadcast, the Q-values of E are updated to $Q_{EB} = R_{EB} + \gamma Q_B$ and

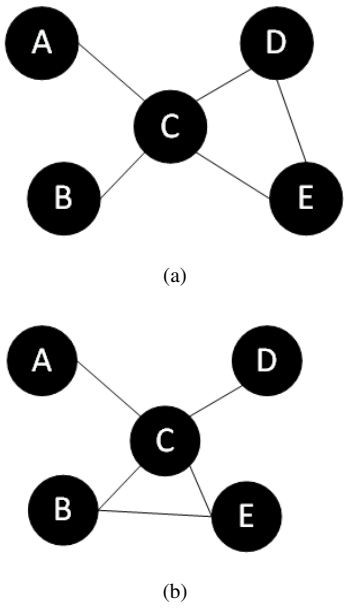

FIGURE 6. (a) topology stable; (b) topology changed.

$Q_{EC} = R_{EC} + \gamma Q_C$, where R_{EB} and R_{EC} are the rewards obtained through information interaction and Q_B and Q_C are the updated maximum Q-values of B and C, respectively. Therefore, DQN can solve the problems caused by nodes moving out of or into to the existing communication range at the current time and update the changeable information in the next broadcast cycle. Because the variability in the modified network parameters between two update cycles is extremely small, the slight lag can be ignored.

V. SIMULATION AND PERFORMANCE EVALUATIONS OF THE DQELR

In this section, numerical experiments are presented to evaluate the performance of the DQELR through an underwater sensor network simulation platform based on NS2, namely Aqua-sim [31]. First, the training cost of the neural network is analyzed. Then, the DQELR performances with various network parameters are explored. Finally, the performances of three routing protocols, namely DQELR, QELAR and VBF, are compared in terms of packet delivery ratio, end-to-end latency, energy efficiency and network lifetime. Packet delivery ratio is the ratio of the number of packets successfully received by the sink to the number of packets sent by the source nodes. End-to-end latency is the time it takes for the packet to be transmitted correctly from the source to the sink. Energy efficiency is defined as the ratio of the network lifetime to the total energy consumption,which represents the network survival time under specific energy consumption. Network lifetime is defined as the time when the first sensor node dies.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

A. NETWORK FRAMEWORK AND SIMULATION SETTINGS

In our simulation scenario, 80 sensor nodes are uniformly deployed at the red points in a $5000 \times 4000 \times 4500\ m^3$ three-dimensional mid size network and one sink is deployed on the center of the water surface at the green point with coordinates (2500, 2000, 4500), as shown in Fig. 7. Each node in the network can act as the source node to generate data packets following an independent Poisson process and aims to transmit the packet to the sink. The rest of the specific parameter values are shown in Table 1 [32], [33] .

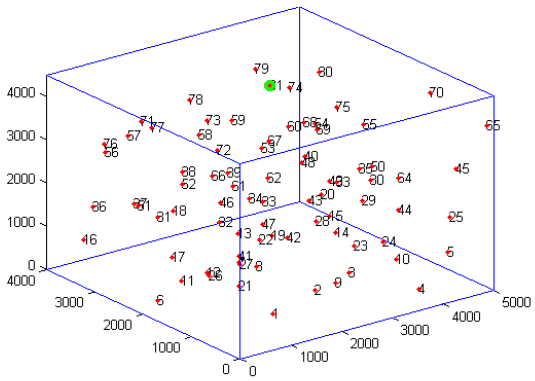

FIGURE 7. Node deployment in the UASNs.

TABLE 1. Experimental parameters.

Experimental paraments	values
Transmission range	2000 m
Initial energy	10000 J
Transmission power	10 w
Receiving power	1 w
Idle power	30 mw
Data packet size	100 bytes
Transmission rate	10 kbps
Packet generation rate λ	0.01 $\sim$ 0.1 packets/s
Experiment runs	80 trials

B. TRAINING COST ANALYSIS

Fig. 8 shows the convergence of the loss value in Eq. (7) as the number of training rounds increases. When performing off-policy training of the neural network, it takes approximately 16,000 training rounds to make loss value converge because the neural network is still in the learning state, after which the agent in the UASN can make a routing decision. After the sensor nodes are deployed underwater, on-policy training is adopted. It takes approximately 1,000 training rounds to make loss value converge when an information updating task is performed through broadcast. This is because on-policy training is based on the full procedure stored in the experience pool at the off-policy stage and achieves faster convergence compared to off-policy training, which leads to less computational overhead for the sensor nodes and a shorter latency for the network in energy and latency sensitive UASNs.

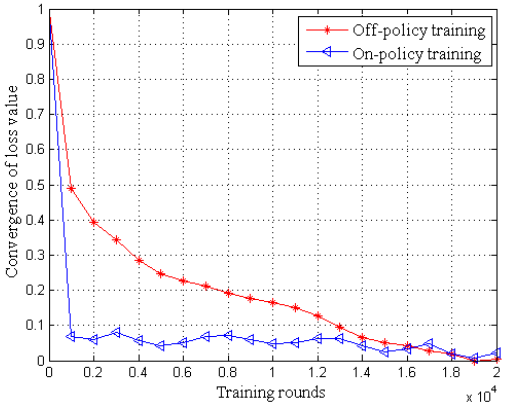

FIGURE 8. Convergence of loss value as the number of training rounds varies.

C. PERFORMANCE EVALUATION OF THE DQELR

In this section, we simulate and analyze the performances of the DQELR with various network parameters and provide a situation in which the forwarding node is selected or replaced in a data packet transmission task from the source to the sink. Fig. 9 shows a case in which node 60 is the source and node 74 is the forwarder most frequently used in several delivery paths from node 60 to the sink 81, along with the usage situation of node 74 under various residual energy percentages of itself.

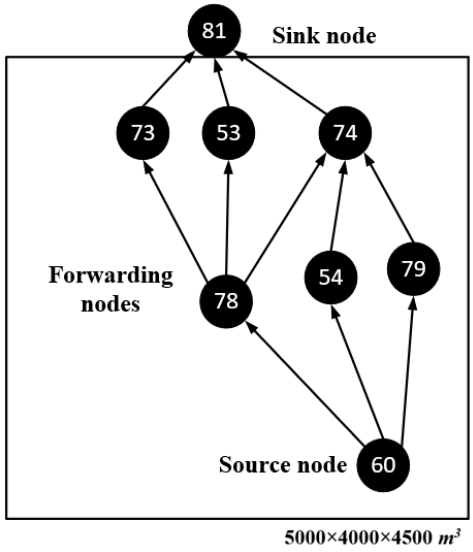

FIGURE 9. A particular path diagram.

In Fig. 10, 0 means that node 74 is not used and 1 means that it is used. As the energy level of node 74 goes from full to exhausted, the delivery path is always changing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

forwarding node. Initially, node 74 is used as the forwarding node. Then, as the residual energy goes down, other forwarders are adopted to replace node 74. Later, as the residual energies of the other nodes go down, node 74 is used as the forwarding node again. Node 74 is always in the process of state transition between forwarding and not forwarding the packet. Because of this mechanism, the time when the energy of node 74 is exhausted can be prolonged, which extends the network lifetime.

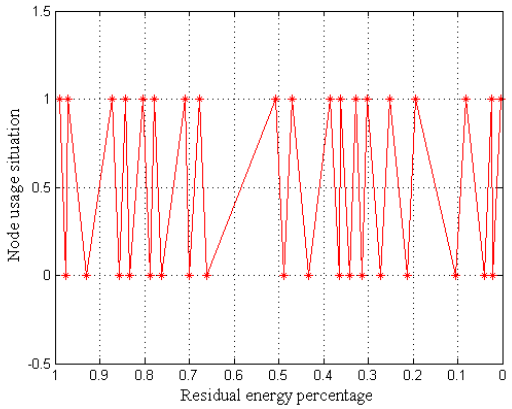

FIGURE 10. Usage situation of node 74 under various residual energy percentages.

Fig. 11 shows the relationship between the end-to-end latency in delivering packets from the source to the sink and the parameters of residual energy α and depth β, defined in Eq. (12), for delivering 1,500 packets. The values of α and β are set from 0 to 1.2 when the trends of end-to-end latency in Fig. 11 have been obvious and stable, and these values are also set from 0 to 1.2 when the trends of relevant performances in Fig. 12, Fig. 13 and Fig. 14 have been obvious and stable. We can observe from Fig. 11 that as α and β change, the end-to-end latency is concentrated in the range of 3.5 s to 3.7 s, except when α and β are both 0. This is because without a constraint on the parameters, the DQN network will choose the forwarder according to the initial maximum reward of the sink with the value of 100 defined in its reward function; thus, the shortest path from the sink will be selected at this time with minimum end-to-end latency. With other values of α and β , there is almost no difference in terms of the end-to-end latency. The reason is consistent with the design of the reward function in Section IV; the reward of the sink is greater than that of the other nodes, and thus the agent still prefers to choose a route closer to the sink. At the same time, there are multiple shortest paths to the sink, nodes with relatively high residual energy can be selected as forwarders in these shortest paths, and the energy of the nodes can be consumed evenly. As a result, the number of detours is reduced when avoiding nodes with low residual energy, even when the energy parameter α is large. It can be proven that this routing mechanism can prolong the network lifetime and effectively control the latency.

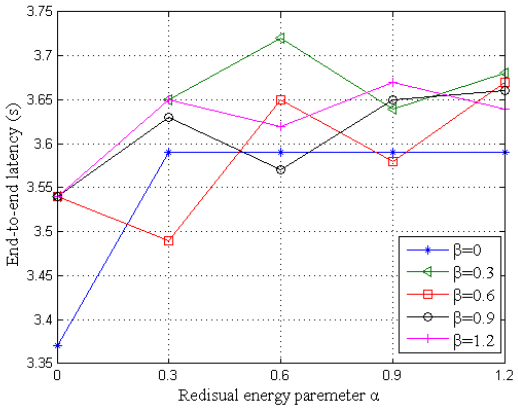

FIGURE 11. End-to-end latency with varying residual energy parameter α and depth β for delivering 1,500 packets.

Fig. 12 shows the relationship between the total energy consumption and the parameters of residual energy α and depth parameter β for delivering 1,500 packets. We can observe from Fig. 12 that as α and β change, the total energy consumption is concentrated in the range of 62,500 J to 65,000 J, except when α and β are both 0. The reason can be seen from Fig. 11: without a constraint on the parameters, the end-to-end latency is minimal and the nodes participating in delivering data packets are correspondingly minimal; thus, the energy consumption is also the least. With other values of α and β, there is almost no difference in terms of the end-to-end latency; thus, there is almost no difference in the total energy consumption. It can be proven that this routing mechanism can prolong the network lifetime and effective control the energy consumption.

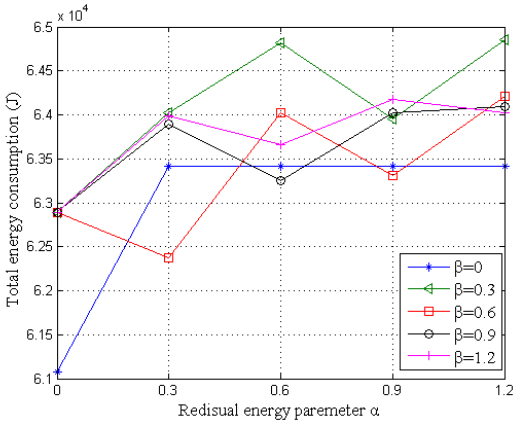

FIGURE 12. Total energy consumption with varying residual energy parameter α and depth parameter β for delivering 1,500 packets.

Fig. 13 shows the relationship between the variance of energy consumption and the parameters of residual energy α and depth parameter β for delivering 1,500 packets. We can observe from Fig. 13 that as α increases and β decreases,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

the variance of energy consumption decreases gradually. The reason is that when the influence of energy is greater, agents will select a forwarder with more energy instead of selecting a node with less energy that is frequently used as the forwarder. As a result, the nodes in the sensor network are used more uniformly; thus, the variance of energy consumption decreases.

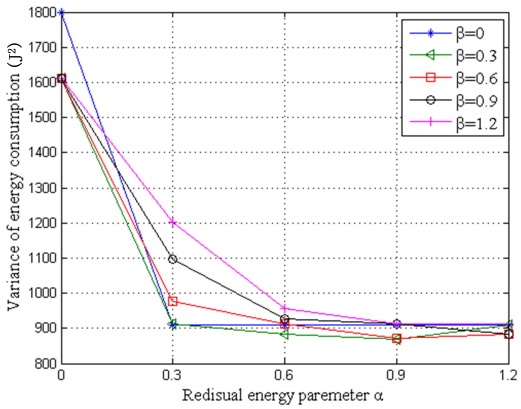

FIGURE 13. Variance of energy consumption with varying residual energy parameter α and depth parameter β for delivering 1,500 packets.

Fig. 14 shows the relationship between the network lifetime and the parameters of residual energy α and depth parameter β. We can observe from Fig. 14 that as α increases and β decreases, the network lifetime increases. When β is equal to 0, the network can send most of the 6,400 packets. This is because the larger the values of α, agents will select the forwarder with more energy and replace the forwarder more frequently, and the energy of the nodes can be consumed evenly, thus maximizing the network lifetime. When α and β are both 0, the network can only send approximately 1,500 packets, and the network lifetime is only 23% of its maximum lifetime when sending 6,400 packets. This is because, without a constraint on the parameters, agents will choose several shortest paths from the sink. A small number of nodes on these paths will be frequently used, so their energy will be exhausted quickly, resulting in a shorter network lifetime. The transmission mechanism here is similar to that of the VBF protocol.

D. PERFORMANCE COMPARISONS AMONG DQELR, QELAR AND VBF

In this section, considering that the representative VBF routing protocol in UASN has good performance in some aspects, and that the DQELR adopts the same type of intelligent routing algorithm based on reinforcement learning as the QELAR routing protocol, we compare the performances of the DQELR, QELAR and VBF in terms of several aspects, including the packet delivery ratio, end-to-end latency, energy efficiency and network lifetime. The performances of the DQELR are presented with α set to 1.2 and β set to 0.

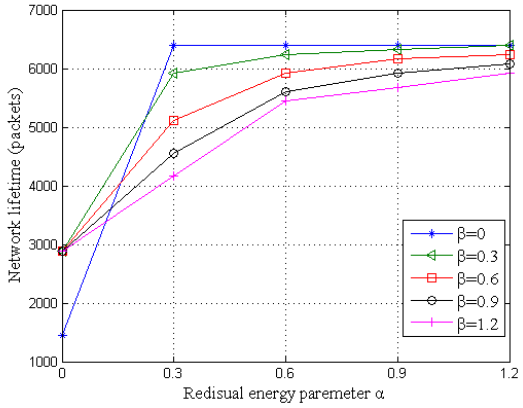

FIGURE 14. Network lifetime with varying residual energy parameter α and depth parameter β.

Fig. 15 shows the relationship between the packet delivery ratio and generation rate among the DQELR, QELAR and VBF with three routing pipe radiuses of 500 m, 1000 m and 2000 m. As the packet generation rate increases, the packet delivery ratio under each scheme decreases. This is because all the nodes in an underwater sensor network can be source nodes. When the packet generation rate is greater, the congestion in the network increases, and the packet collision rate increases as well. In the DQELR mechanism, all the nodes can participate in data transmission because there is no limitation on the routing pipe; thus, they can achieve the maximum packet delivery ratio with a small packet generation rate. With an increase in the package generation rate, DQELR can still achieve a package delivery ratio similar to that of the VBF2000 . However, with the multicast communication method, the packet delivery ratio of the QELAR is always lower than that of the DQELR.

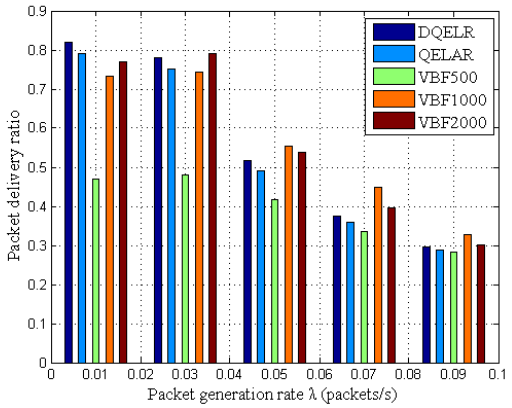

FIGURE 15. Packet delivery ratio comparison among the DQELR, QELAR and VBF with packet generation rate λ.

Fig. 16 shows the relationship between the end-to-end latency and packet generation rate among the DQELR, QELAR and VBF with three routing pipe radiuses of 500 m,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2891590, IEEE Access

1000 m and 2000 m. The packet generation rate is a selection of three representative values from Fig. 15. As the packet generation rate increases, the end-to-end latency under each scheme increases. This is because with the increase of packet collision rate, the probability of data transmission failure will increase and more packets will be retransmitted, leading to the increase of end-to-end latency for successful packets transmission. Furthermore, because more packets need to be delivered, all of these five mechanisms will detour to avoid packet collisions or to consume energy more uniformly, which can also increase the end-to-end latency. In these five mechanisms, the VBF has the lowest latency, the QELAR has the highest latency, and the DQELR has a latency between those of the QELAR and the VBF. This is because the VBF selects forwarding nodes within the routing pipe radius and the delivery path is relatively short. As the radius of the routing pipe increases, the number of nodes participating in forwarding packets increases and thus the end-to-end latency of the VBF increases. To consume energy more uniformly, the QELAR selects forwarding nodes with more residual energy as much as possible, which can result in even spreading throughout the entire network, thus causing detours and increasing the end-to-end latency. The DQELR has detour limitations when selecting forwarding nodes with more residual energy, which leads to a lower end-to-end latency than that of the QELAR but a relatively higher latency than that of the VBF, which only finds the shortest path without considering the residual energy.

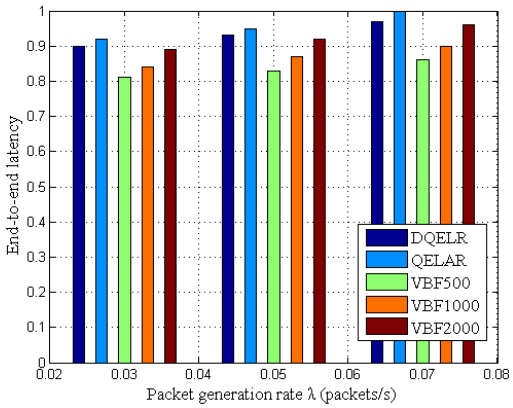

FIGURE 16. End-to-end latency comparison among the DQELR, QELAR and VBF with packet generation rate λ.

Fig. 17 and Fig. 18 show the relationship between the energy efficiency and packet generation rate, and the network lifetime and packet generation rate, respectively, among the DQELR, QELAR and VBF with three routing pipe radiuses of 500 m, 1000 m and 2000 m. As the packet generation rate increases, the energy efficiency and the network lifetime under each scheme decreases. This is because more energy will be consumed to deliver more packets, thus the network lifetime will be shortened, and leading to the decrease of energy efficiency. In these five mechanisms, the DQELR has the highest energy efficiency and network lifetime, followed by the QELAR, and the VBF is the lowest. The DQELR improves the network lifetime by approximately 34-36% compared to the QELAR. Because the DQELR is designed with a method to consume energy more uniformly and uses a hybrid of the broadcast and uncast communication mechanisms to save energy, it shows great advantages in terms of energy efficiency and network lifetime. However, the QELAR and the VBF adopt a broadcast communication method with high energy consumption even when the nodes have no packet transmission task, which leads to low energy efficiency. The QELAR also has no detour limitation when selecting forwarding nodes with more residual energy, and the energy of the nodes in the VBF can be exhausted quickly because of frequent use. These characteristics make the network lifetime and energy efficiency of the QELAR and VBF lower than that of the DQELR.

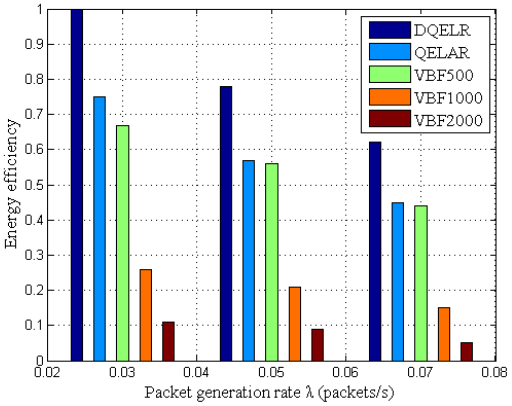

FIGURE 17. Energy efficiency comparison among the DQELR, QELAR and VBF with packet generation rate λ.

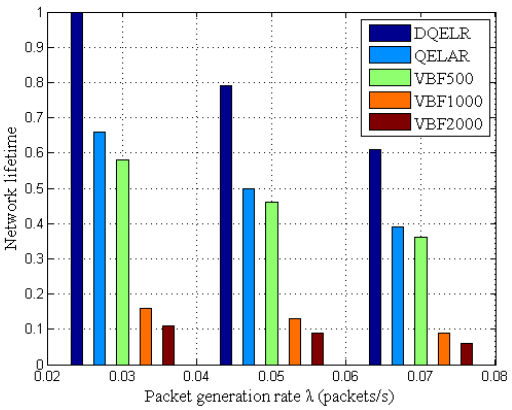

FIGURE 18. Network lifetime comparison among the DQELR, QELAR and VBF with packet generation rate λ.

VI. CONCLUSION

In this paper, an energy- and latency-aware routing protocol, the DQELR, is proposed to prolong network lifetimes in UASNs. The DQELR adopts a DQN algorithm with both off-policy and on-policy methods to make routing decisions adaptively in different network conditions, and a hybrid of the broadcast and unicast communication mechanisms is designed to save network overhead. With less energy consumption and a strict latency limitation, the network lifetime can be extended by selecting forwarding nodes with the maximum Q-value. In addition, the DQELR can cope with network topology changes by adopting an on-policy method to make a new routing decision when the current route is corrupted.

Numerical experiments are presented to evaluate the performance of the DQELR with various parameters, and comparisons among the DQELR, QELAR and VBF are provided for several performance indicators. According to the simulation results, the DQELR can achieve a long network lifetime with a relatively high residual energy parameter α and low depth parameter β, while the end-to-end latency and energy consumption of the network show almost no difference with respect to these parameters. Compared to the QELAR and VBF, the DQELR performs best, with the longest network lifetime, highest energy efficiency and a slightly longer latency than that of the VBF. The experimental results show that the DQELR improves the network lifetime by approximately 34-36% compared to the QELAR. In conclusion, the DQELR can achieve a superior network lifetime with better latency and energy efficiency performances relative to other general schemes applied in UASNs.

REFERENCES

[1] Y. Pan, R. Diamant, and J. Liu, "Underwater acoustic sensor networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, Aug. 2016.

[2] P. Jiang, X. Wang, and L. Jiang, "Node deployment algorithm based on connected tree for underwater sensor networks," *Sensors*, vol. 15, no. 7, pp. 16 763–16 785, Jul. 2015.

[3] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "Vapr: Void-aware pressure routing for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 895–908, May 2013.

[4] M. Shah, Z. Wadud, A. Sher, M. Ashraf, Z. A. Khan, and N. Javaid, "Position adjustment–based location error–resilient geo-opportunistic routing for void hole avoidance in underwater sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 21, p. e4772, Oct. 2018.

[5] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communications Letters*, vol. 9, no. 11, pp. 976–978, Nov. 2005.

[6] J. Jiang and G. Han, "Routing protocols for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 58–63, Jan. 2018.

[7] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," Tech. Rep., Oct. 2003.

[8] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, Feb 1999, pp. 90–100.

[9] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless networks*, vol. 7, no. 6, pp. 609–616, Nov. 2001.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, Dec. 2013.

[11] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796–809, Feb. 2010.

[12] P. Xie, J.-H. Cui, and L. Lao, "Vbf: vector-based forwarding protocol for underwater sensor networks," in *International conference on research in networking*. Springer, May 2006, pp. 1216–1221.

[13] N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini, "Improving the robustness of location-based routing for underwater sensor networks," in *Oceans 2007-Europe*. IEEE, Jun. 2007, pp. 1–6.

[14] H. Yu, N. Yao, and J. Liu, "An adaptive routing protocol in underwater sparse acoustic sensor networks," *Ad Hoc Networks*, vol. 34, pp. 121–143, Nov. 2015.

[15] H. Yan, Z. J. Shi, and J.-H. Cui, "Dbr: depth-based routing for underwater sensor networks," in *International conference on research in networking*. Springer, May 2008, pp. 72–86.

[16] R. Mohammadi, R. Javidan, and A. Jalili, "Fuzzy depth based routing protocol for underwater acoustic wireless sensor networks," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 7, no. 1, pp. 81–86, Jan. 2015.

[17] Y.-S. Chen, T.-Y. Juang, Y.-W. Lin, and I.-C. Tsai, "A low propagation delay multi-path routing protocol for underwater sensor networks," *Internet Technol*, vol. 11, no. 2, pp. 153–165, Mar. 2010.

[18] M. Al-Bzoor, Y. Zhu, J. Liu, A. Reda, J.-H. Cui, and S. Rajasekaran, "Adaptive power controlled routing for underwater sensor networks," *International Journal of Sensor Networks*, vol. 18, no. 3/4, pp. 549–560, Aug. 2012.

[19] U. Lee, P. Wang, Y. Noh, L. F. Vieira, M. Gerla, and J.-H. Cui, "Pressure routing for underwater sensor networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, Mar. 2010, pp. 1–9.

[20] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "Vapr: Void-aware pressure routing for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 895–908, May 2013.

[21] N. Li, J.-F. Martínez, J. M. Meneses Chaus, and M. Eckert, "A survey on underwater acoustic sensor network routing protocols," *Sensors*, vol. 16, no. 3, p. 414, Mar. 2016.

[22] R. Banerjee and C. K. Bhattacharyya, "Cluster based routing algorithm with evenly load distribution for large scale networks," in *Computer Communication and Informatics (ICCCI), 2014 International Conference on*. IEEE, Jan. 2014, pp. 1–6.

[23] M. Tariq, M. S. Latiff, M. Ayaz, Y. Coulibaly, and N. Al-Areqi, "Distance based reliable and energy efficient (dree) routing protocol for underwater acoustic sensor networks," *Journal of networks*, vol. 10, no. 5, p. 311, May 2015.

[24] Z. M. Zahedi, R. Akbari, M. Shokouhifar, F. Safaei, and A. Jalali, "Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks," *Expert Systems with Applications*, vol. 55, pp. 313–328, Aug. 2016.

[25] M. Faheem, G. Tuna, and V. C. Gungor, "Lrp: Link quality-aware queue-based spectral clustering routing protocol for underwater acoustic sensor networks," *International Journal of Communication Systems*, vol. 30, no. 12, p. e3257, Nov. 2017.

[26] M. Faheem, G. Tuna, and V. C. Gungor, "Qerp: Quality-of-service (qos) aware evolutionary routing protocol for underwater wireless sensor networks," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2066–2073, Sep. 2018.

[27] Z. Jin, Y. Ma, Y. Su, S. Li, and X. Fu, "A q-learning-based delay-aware routing algorithm to extend the lifetime of underwater sensor networks," *Sensors*, vol. 17, no. 7, p. 1660, Jul. 2017.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, Feb. 2015.

[29] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: An acoustic communications and navigation system for multiple platforms," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, Sep. 2005, pp. 1086–1092.

[30] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, Feb. 2016, pp. 1928–1937.

[31] Z. Rahman, F. Hashim, M. Othman, and M. F. A. Rasid, "Reliable and energy efficient routing protocol (reep) for underwater wireless sensor networks (uwsns)," in *2015 IEEE 12th Malaysia International Conference on Communications (MICC)*, Nov 2015, pp. 24–29.

[32] H. Yan, L. Wan, S. Zhou, Z. Shi, J. H. Cui, J. Huang, and H. Zhou, "Dsp based receiver implementation for ofdm acoustic modems ąî," *Physical Communication*, vol. 5, no. 1, pp. 22–32, Sep 2012.

[33] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey, "An integrated, underwater optical /acoustic communications system," in *OCEANS'10 IEEE SYDNEY*, May 2010, pp. 1–6.

YISHAN SU received his B.E. degree, M.E. degree and Ph. D degree from Tianjin University, in 2008, 2010 and 2015 respectively. He was a visiting professor in University of Connecticut, CT, USA in 2012-2013. He is currently working as an assistant professor in Tianjin University, Tianjin, China.

His research interests include protocols design and implementation in underwater sensor networks (UWSNs) and delay/disruption tolerant networks (DTN).

RONG FAN received her B.E. degree in information engineering from China University of Mining and Technology, jiangsu, China, in 2013. She is currently working toward the M.S. degree with the Department of Information and Communication Engineering, Tianjin University, Tianjin, China.

Her research interests include protocols design in underwater sensor networks (UWSNs) and delay/disruption tolerant networks (DTN).

XIAOMEI FU received the M.S. and the Ph.D degree from Tianjin University, China in 2000 and 2006, respectively. She was an academic visitor of Imperial college of London, UK from 2009 to 2010. She is currently a professor of Tianjin University, China.

Her research interests include wireless network communication, physical layer security, underwater communication networks.

ZHIGANG JIN received his B.E. degree from Hebei University of Technology, Tianjin, China, in 1993, M.E. degree from Tianjin University, Tianjin, China, in 1996 and Ph.D. degree from Tianjin University, Tianjin, China, in 1999. He was a visiting professor in Ottawa University, Ottawa, Canada, in 2002. He is currently a professor in Tianjin University, Tianjin, China.

His research interests focus on the performance evaluation of traffic and networks, the management and security of the computer networks, and the wireless networks.