Q-Learning-based Energy-Efficient Custom Cooperative Routing Protocol for Underwater Wireless Sensor Network

Payal Karmakar[1], Rajshekhar Rakshit[2], Alak Roy[3*], Paresh Kumar Palai[4]
[1,3,4]Department of Information Technology, Tripura University, Suryamaninagar, Tripura, India.
[2]Department of Computer Science and Engineering, Indian Institute of Technology Madras, India
[*]E-Mail: alakroy@tripurauniv.ac.in
[*]Corresponding Author.

Abstract—Underwater Wireless Sensor Networks (UWSNs) are currently a pivotal focus in academic and industrial domains due to their diverse applications, such as disaster prevention, military security, environmental monitoring, data collection, scientific research, and industrial usage. The underwater area is very dense, and hence, exploring such a denser environment is difficult in the first place. To make this exploration easy, underwater sensors are used that can collect information from underwater and forward it to the base station where these data can be used for various purposes. The problem with UWSNs is that they have a very limited amount of energy, so optimizing the energy usage of the sensors will be beneficial. To deal with this, this paper proposed a Q-Learning-based Energy-Efficient Custom Cooperative Routing (QEECCR) protocol that uses a Q-learning technique to optimize the routing based on the energy levels of the sensors. The algorithm selects a node based on the Q-value of the node for forwarding data to the base station. The proposed routing protocol is compared with the QCMR routing protocol, and results showed that it consumes less energy compared to the QCMR. With an increasing number of nodes under the water, designing a manual routing for low energy consumption becomes hard. This proposed protocol can remove human intervention and can find the routing path with less time and with higher accuracy.

Keywords: Underwater Wireless Sensor Network, Routing Protocols, Location-based Routing Protocols, Location Free Routing Protocols, Cooperative Routing Protocols

INTRODUCTION

Underwater Wireless Sensor Networks (UWSNs) are the most important research area in academics and industry. It has a variety of essential applications, such as disaster prevention, military and security, environmental monitoring, data collection, scientific, industrial, etc. In recent times, there has been a boom in Artificial Intelligence (AI). This also impacted the UWSNs. The use of AI in UWSNs has been proven to improve the performance of the network by creating an efficient routing path that can adapt itself to the ocean. Underwater networks contain many sensors that can communicate with each other and receive and forward information through multi-hop to the destination. Due to bad conditions of underwater, it is still a difficult task to achieve a good level of optimization. The sensors have a limited amount of energy; if the sensor tries to send data to the base station directly, then it may use a large amount of energy, and bit error may also increase due to the large distance. Underwater sensor nodes are powered by batteries, so optimizing the power consumption of the sensor nodes is a challenging task. One of the most important tasks of communication is routing, routing is a type of high computation task, and routing also decides the path for

https://doi.org/10.22232/stj.2025.13.01.20

the data to travel; if the routing makes a longer path, then; as a result, energy consumption will be large, that's why optimizing routing can reduce the energy consumption of the whole network in a significant way.

Many routing protocols for underwater sensors have been designed to solve the problem of high-bit error and repeated transmission. In (Xie P *et al.* 2006), a routing protocol has been proposed that uses locations of source node, sink node, and relay nodes to optimize the energy in a dynamic underwater environment; this algorithm is called Vector-based routing (VBF). In this algorithm, each node calculates the density of nodes of its neighboring nodes based on local information, and then the algorithm chooses the next node. The VBF doesn't take the forwarder's energy information into consideration. To solve this problem, another algorithm called Lifetime-extended Vector-based forwarding routing (LE-VBF) has been proposed in (Xiao X *et al.* 2012). LE-VBF considers the position and the energy information of the nodes to perform routing to optimize energy usage. An energy-efficient routing algorithm was developed for nodes that don't change their location (Gul H *et al.* 2021); EERBCR algorithm performs well where nodes don't have to change their locations. The main objective is to optimize the energy usage of UWSNs so that the nodes can work for longer periods of time with the provided energy. In this paper, power optimization is done by increasing the efficiency of the routing algorithm for the UWSNs. Q-learning is a type of algorithm that interacts with the environment and takes action based on the feedback it gets from the environment. In this case, the environment is an underwater sensor. Interaction with these sensors from time to time may give different results, and thus, the routing path will change every time the environment changes; this dynamic nature of this algorithm is faster and can remove manual interventions. With the increasing number of nodes, manual intervening in low energy consumption path becomes hard; with Q-learning, the algorithm can design the whole path efficiently in less time without any human interventions.

This paper proposes an algorithm for Cooperative routing using a Q-learning-based approach, earlier using the cooperative communication approach, the node which is going to transfer data chooses the node with minimum energy consumption to act as a cooperative node, but due to this, sometimes the energy consumption of the whole network increases because of local optimum. The proposed algorithm that takes distance and energy into consideration to performing routing. The Q-learning algorithm provides a Q-value to every node, and based on this Q-value, the algorithm chooses the node that can be used to forward data. After the node is selected, the proposed algorithm uses cooperative communication to perform the data transmission. The algorithm chooses a node that may increase the total path. Optimization of this problem may reduce the total energy consumption of the network. The proposed algorithm discusses the way to reduce the total distance traveled by the data and thus reduce the energy consumption of the whole underwater sensor network.

The remaining portion of this paper is organized as follows. Section 2 represents the background details of the Q-Learning algorithm and presents related work to cooperative-based routing protocols. Section 3 represents the proposed algorithm and the working principle of the algorithm. Section 4 represents the experimental results, and Section 5 gives the conclusion of the paper with future directions.

RELATED WORK

Q-learning is a type of reinforcement learning. In this, the algorithm creates a Q-table after interacting with the environment. Based on the interaction, the environment gives a reward, which can be either positive or negative. If the reward is positive, that means the algorithm has selected the right path, and if negative it means the algorithm selected the wrong path. When the algorithm trains the algorithm enough number of times the agent tends to get positive results always. Once the Q-table is ready, the algorithm can use this table to perform routing.

Reinforcement Learning is a type of algorithm that trains the agent to interact with the environment. The main task of the agent is to maximize the feedback reward it gets from the environment. Reward can be either positive or negative. In Q-Learning (Watkins CJ and Dayan P 1992) the algorithm mainly works with three things. First is Environment, second is Agent, and the third one is Reward.

In the Q-Learning algorithm, the agent selects a random action on the current state, and the environment feed-backs as reward based on the current state and action that the agent has performed.

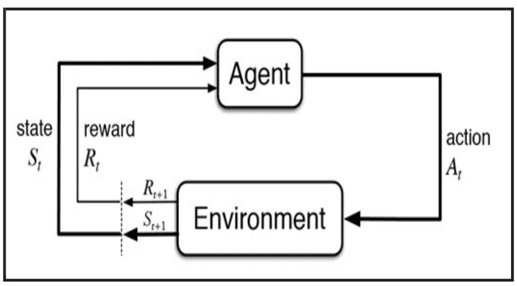

Fig. 1: Q-learning Approach

$$Q(s, a) = R(s, a) + \gamma * \max (Q(s', a')) \qquad (1)$$

In this equation (1), the 's' denotes the current state, 'a' denotes the current action's denotes the next state, and a' denotes the set of all the actions that the next state 's' can take. Q(s,a) denotes the Q-value of state 's' on action 'a'. R(s,a) denotes the reward got after taking action a on state s. max(Q(s',a')) represents the maximum Q- value that is possible when the algorithm takes action a' on state s'. γ here is called as discount factor, its value ranges from $0 < \gamma < 1$. If the value of γ is larger, then it means that the agent will try to learn new things instead of using what it has already learned.

The advent of AI has led to the integration of various intelligent algorithms, includ- ing the ant colony algorithm (ACA), artificial fish-swarm algorithm (AFSA), simulated annealing algorithm (SAA), and Q-learning algorithm, into the design of routing for UWSNs. ACA emulates the foraging behavior of ants to discover optimal routes, while it has the drawback of prolonged search times and susceptibility to local optima (Chen Y *et al.* 2021).

In recent research, a Q-learning-based adaptive routing protocol (QELAR) was introduced to enhance the longevity of UWSNs. QELAR employs a reward mechanism based on energy consumption, allowing an AI agent trained with this protocol to autonomously select routes with minimal energy usage (Hu T and Fei Y 2010).

In (Nasir H *et al.* 2014), the authors proposed Collaborative Depth Based Routing (CoDBR), which is an enhanced version of DBR. All sensor nodes communicate their depth details with their neighbors. The source node first discovers its neighbors and adds them to the neighbor's collection. As the destination for the next hop, the source node chooses the node with the minor depth from the list of neighbors.

In (Rahman MA *et al.* 2017), the authors proposed an Energy-Efficient Cooperative Opportunistic Routing Protocol (EECOR) protocol that addresses the issue of energy consumption. The source chooses the optimum path for the packet transmission using a fuzzy logic-based relay selection mechanism, sends it to the destination, and uses holding time for each forwarder to avoid packet collisions. This protocol shows better energy usage, packet delivery, and end-to-end latency compared with DBR, FDBR, VBF, and HH-VBF.

In (Watkins CJ and Dayan P 1992) proposed a Q-Learning-based multi-hop cooperative routing (QMCR) protocol for UWSNs. QMCR protocol uses different kinds of communication methods to transfer data from the source node to the destination node. The main part of the algorithm is the Q-table, as every decision is made based on the Q-values. It uses the modular approach of programming. Algorithms that are discussed below are made as functions that take arguments as input and return the output, and these outputs are further used to take actions.

QMCR main task is to design the optimal routing path with the lowest transmission energy. In Q-Learning, the main task is to design the reward table. Once the reward table is ready, the algorithm can find the routing path which is optimal. In the QMCR, Algorithm 1 takes the negative distance between two nodes as the reward. In QMCR, let's say the distance between node 2 and node 5 is 3km and the distance between node 2 and node 3 is 2km, then the reward of taking action the 5 in state 2 is -3, and the reward of taking action 3 in state 2 is -2km. After the training of the agent is finished, the value of Q(2, 3) > Q(2, 5) and thus the agent will choose action 3 in state 2. The algorithm will select the best routing path based on the energy consumption.

Whenever cooperative communication is possible to forward data forward, the algorithm can use the nodes present between the transmitting node and receiving node to act as cooperative node. The node having less transmitting energy consumption is chosen as cooperative node, but it may happen that the node selected has less energy consumption but the distance that the data have to travel is much larger, and thus in return, the algorithm will use more energy.

PROPOSED APPROACH

This section proposes a Q-Learning-based Energy-Efficient Custom Cooperative Routing Protocol (QEECCR) Networks for Underwater Wireless Sensor Network that uses less energy as compared to QCMR. The algorithm can be optimized by making sure that the algorithm only takes the path that has minimum energy. The following steps are given to optimize the cooperative communication:

1) Initialize the environment: Distribute the nodes in the environment and create the Q-table and reward table for the algorithm to perform routing. Create an array routing path which will be used to keep the track of the path followed. The 1st item inside the array is the source node.

2) Performing direct communication: The algorithm iterates over the nodes list till the destination node is not in the routing path array. Inside the loop, the algo- rithm will take the node that has the maximum reward value, and then the distance between this new node and the last node present in routing path is measured. If the distance ≤ 2.5, then transfer

the data to the node and add the new node to the routing path array.

3) Performing custom-cooperative communication: If 2.5 ≤ distance ≤ 4 then get all the nodes that are present between the transmitting node and the receiving node. Track the energy consumption that will occur if the algo- rithm directly sends the data, then get the node with min energy consumption. If the new node energy ≤ previous node energy then use it as a cooperative node.

In this way, the algorithm can be optimized. This algorithm checks that if the energy consumption of a node is increased in the future due to a larger distance, then don't use this node as a cooperative node.

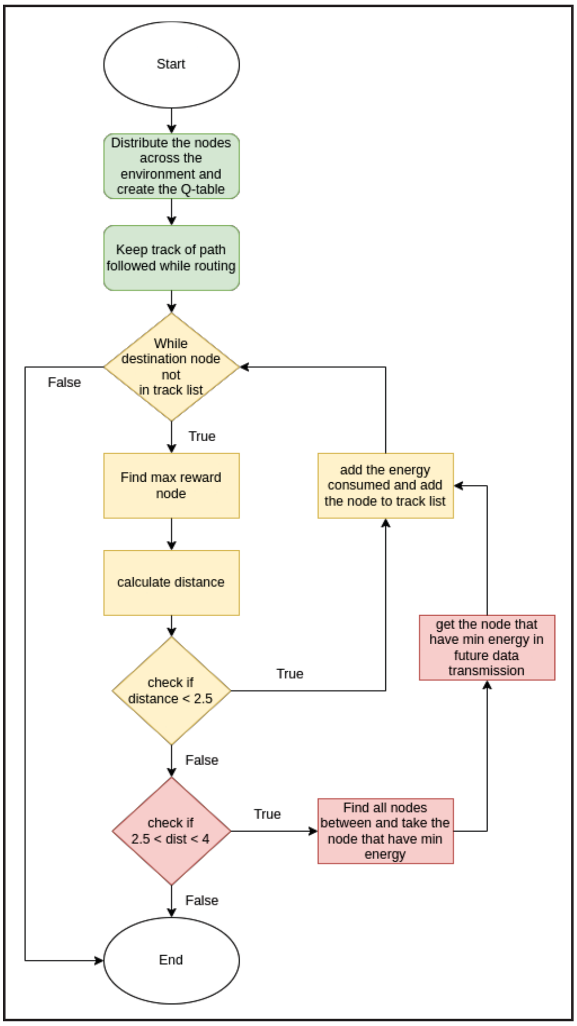

Fig. 2: Flowchart of Q-Learning-based Energy-Efficient Custom Cooperative Routing Protocol

Figure 2 represents the proposed flowchart of the custom-cooperative communi- cation as a whole. The green highlighted part denotes the first step of the proposed algorithm where the nodes are distributed and create the Q-table for the routing algo- rithm. The yellow part denotes the second step of the proposed algorithm, where the algorithm keeps track of those nodes that have distance ≤ 2.5 and add the energy levels to the track list. The red part denotes the third step of the proposed algorithm. Till step two, the algorithm is similar to cooperative communication; the third step is the crucial part, where the optimization of the routing algorithm takes place. If the distance between the transmitting node and destination node is less than 2.5km, the node directly sends the data to the receiving node. If the distance between the same two nodes is greater than 2.5km and less than 4km, then find all the nodes between these two nodes and track the energy consumption of all the between nodes and take that node as a cooperative node that has minimum energy in future communication. This algorithm gives more accurate result than cooperative communication. The out- put and results are discussed in the next section. Algorithm 1 discusses the approach of custom-cooperative routing. In this, cooperative nodes are used for transferring data to the destination node but in a way that the total energy consumption of the network is minimum.

EXPERIMENTAL RESULTS

The experiment is performed in Python language. Libraries that are used in this are numpy and matplotlib. Google Colab environment is used to perform the experiments. For Simpler use of the code, an Object-oriented approach is used.

As shown in Figure 3, a total of 18 nodes are arranged, where each circle represents a sensor node, and the number represents the index of nodes. The source node S is node 14 and the destination node D is node 7. In order for the transmission to proceed smoothly, the 18 nodes are located in a rectangular area with around 14 km length and 5 km width. The environment parameters that are used for this experiment are given below: 1) Number of nodes = 18, 2) Number of iterations = 1500, 3) Discount factor = 0.8, 4) Value of k = 1.5, 5) Value of r1 = 2.5, 6) Value of r2 = 4, 7) Range of x-axis = 14, 8) Range of y-axis = 5.

We have considered two case scenarios as an example. In both case we compared QCMR cooperative and non-cooperative routing with the proposed custom cooperative routing. We compared the total energy consumed in both these cases. We found that custom cooperative routing gives less energy consumption than QCMR cooperative and non-cooperative routing.

Algorithm 1: Algorithm of Q-Learning-based Energy-Efficient Custom Cooperative Routing Protocol

Algorithm 1 Algorithm of Q-Learning-based Energy-Efficient Custom Cooperative Routing Protocol

```
function Custom Cooperative Routing (Q table)
    routing path ← [source node]
    final energy consumed ← 0
    while destination node ∈/ routing path do
        prev node ← last item of routing path
        Q table[:, prev node] ← −1000
        node loc ← index of prev node where max reward
        distanceij ← euclidean distance(env.node locations[prev node], env.node locations[node loc])
        if distanceij ≤ env.r1 then
            energy ← cooperative energy consumption(distanceij )
            routing path ← node loc
        else
            if distanceij ≥ env.r1 ∧ distanceij ≤ env.r2 then
                energyprev ← cooperative energy consumption(distanceij )
            end if
        end if
        nodes ← nodes between prev node and node loc
        min E ← MAX ENERGY POSSIBLE
        if min E < energyprev then
            energy ← cooperative energy consumption(distanceij )
            routing path ← node loc
            min index ← 0
            for i ← 0 to nodes do
                curr node ← nodes[i]
                distance ← euclidean distance(env.node locations[prev node], env.node locations[curr nod
                energy ← attenuation energy(distance)
                if energy ≤ min E then
                    min E ← energy
                    min index ← i
                end if
            end for
            distancecj ← euclidean distance(env.node locations[nodes[min index]], env.node locations[n
            energy ← cooperative energy consumed(distanceij , distancecj )
            routing path ← nodes[min index]
        else
            energy ← energyprev
            routing path ← node loc
        end if
        final energy consumed += energy
    end while
    return routing path
end function
```

4.1 Case 1

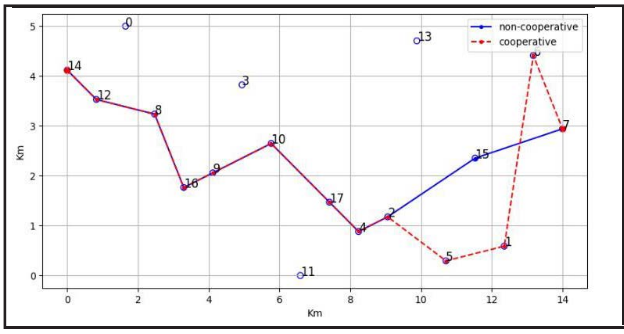

Fig. 3: Cooperative and Non-cooperative Routing in QCMR Protocol (Case 1)

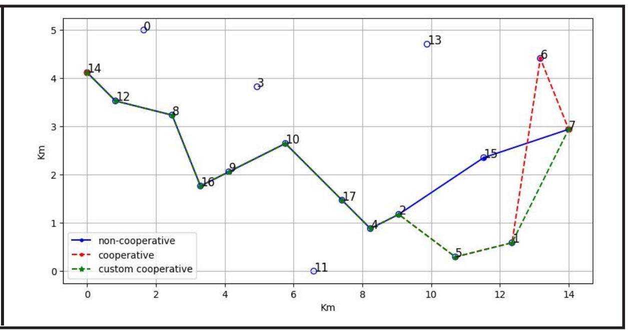

Fig. 4: Custom cooperative routing in QEECCR with respect to QCMR protocol (Case 1)

Table 1: Comparison Table for Case 1

Algorithm	Routing Path	Total Energy Consumed (Joule)
Non-Cooperative	14-12-8-16-9-10-17-4-2-15-7	7155993.045759567
Cooperative	14-12-8-16-9-10-17-4-2-5-1-6-7	6772851.631869233
Custom-Cooperative	14-12-8-16-9-10-17-4-2-5-1-7	6100890.370335663

4.2 Case 2

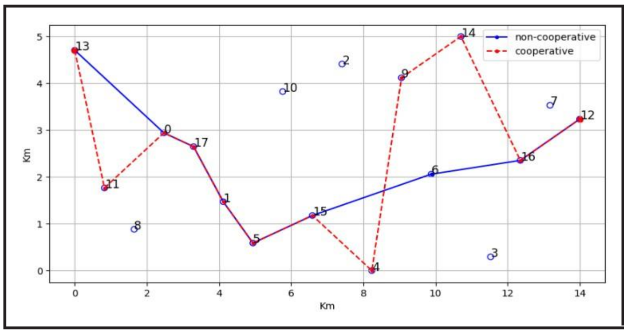

Fig. 5: Cooperative and Non-cooperative Routing in QCMR Protocol (Case 2)

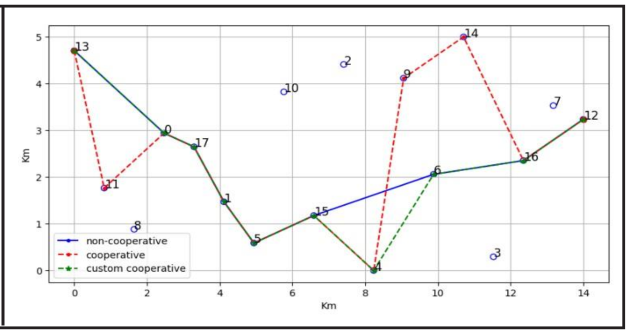

Fig. 6: Custom cooperative routing in QEECCR with respect to QCMR protocol (Case 2)

Table 2. Comparison Table for Case 2

Algorithm	Routing Path	Total Energy Consumed (Joule)
Non-Cooperative	13-0-17-1-5-15-6-16-12	6281496.264072517
Cooperative	13-11-0-17-1-5-15-4-9-14-16-12	8231940.665120615
Custom-Cooperative	13-0-17-1-5-15-4-6-16-12	7173696.5802509375

In the Figure 3, from node 1, data can directly be sent data to node 7. But using cooperative routing, data are sent from node 1 to node 6 and then to node 7, this increases the energy consumption. The energy consumption of non-cooperative approach found out to be 7155993.045759567 Joule, and the energy consumption of cooperative approach is found out to be 6772851.631869233 Joule.

In the Figure 4, the green line denotes the path that was produced by the pro-posed custom cooperative algorithm. Here, node 1 directly sends the data to node 7, which consumes less energy than the cooperative routing protocol. Here, the energy consumption of non-cooperative and cooperative approaches is the same as above, but the energy consumption of the custom-cooperative approach was found out to be 6100890.370335663 Joule as shown in Table 1.

In the Figure 5, from node 4, data can directly be send data to node 6. But using cooperative routing, data are first sent to node 9 and due to this the data travelled a larger distance and thus increase the energy consumption. The energy consumption of non-cooperative approach found out to be 6281496.264072517 Joule, and the energy consumption of cooperative approach is found out to be 8231940.665120615 Joule.

In figure 6, the green line denotes the path that was produced by the proposed algorithm. Here, node 4 directly sends the data to node 6. Using this, the data travel a lesser distance compared to cooperative approach and thus consume less energy than the cooperative routing protocol. Here, the energy consumption of non-cooperative and cooperative approach is same as above but the energy consumption of the custom-cooperative approach found out to be 7173696.5802509375 Joule as shown in Table 2.

In this case, it can be observed that the non-cooperative communication is consuming less energy compared to other two, but there are disadvantages of using non-cooperative approach, the data can be lost in non-cooperative approach, this is the reason of using cooperative approach. It can be observed that the energy consumption of cooperative communication is larger than the energy consumption of custom-cooperative communication.

CONCLUSIONS

In this paper, an optimized Q-learning-based custom cooperative routing protocol has been proposed that performs better than the QMCR cooperative routing protocol. Q-Learning is a type of reinforcement learning algorithm that trains the agent to learn from interacting with the environment. The critical part of the whole algorithm is the design of the Q-table; every decision that the nodes will take while transmitting is decided by the Q-table. The algorithm performs better if the value of the Q-table is accurate. For accurate Q-values of the table, the agent has to be trained with more number of iterations until the value of the Q-table gets stabilized. Non-cooperative algorithm consumes more energy than the cooperative approach, and the cooperative approach consumes more energy than the custom-cooperative approach. Optimization in the routing is crucial because one of the key features of communication is routing which is computationally expensive. If routing does not select the shortest distance path, then as a result, energy consumption of the network may increase, and thus energy will be consumed faster. This Q-learning approach for cooperative communication maximizes the overall performance of the communication.

REFERENCES

Xie P, Cui JH, Lao L (2006) VBF: Vector-based forwarding protocol for underwater sensor networks. In: NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems: 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, May 15-19, 2006. Proceedings 5, Springer, pp. 1216–1221

Xiao X, Ji XP, Yang G, Cong YP (2012) LE-VBF: Lifetime-extended vector-based forwarding routing. In: 2012 International Conference on Computer Science and Service System, IEEE, pp. 1201–1203

Gul H, Ullah G, Khan M, Khan Y (2021) EERBCR: Energy-efficient regional based cooperative routing protocol for underwater sensor networks with sink mobility. Journal of Ambient Intelligence and Humanized Computing, pp 1–13

Watkins CJ, Dayan P (1992) Q-learning, Machine learning 8, pp. 279–292

Chen Y, Zheng K, Fang X, Wan L, Xu X (2021) QMCR: A Q-learning-based multi- hop cooperative routing protocol for underwater acoustic sensor networks. China Communications 18(8), pp 224–236

Hu T, Fei Y (2010) QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. IEEE transactions on mobile computing 9(6), pp.796–809

Nasir H, Javaid N, Ashraf H, Manzoor S, Khan Z A, Qasim U, Sher M (2014) CODBR: Cooperative depth based routing for underwater wireless sensor networks. In: 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, IEEE, pp. 52–57

Rahman MA, Lee Y, Koo I (2017) EECOR: An energy-efficient cooperative oppor- tunistic routing protocol for underwater acoustic sensor networks. IEEE Access, vol. 5, pp. 14119–14132