MAC Protocol for Underwater Acoustic Networks Based on Deep Reinforcement Learning

Xuan Geng

xuangeng@shmtu.edu.cn

College of Information Engineering,
Shanghai Maritime University
Shanghai, 201306, P.R. China

Y. Rosa Zheng

yrz218@lehigh.edu

Dept. of Electrical & Computer Engineering,
Lehigh University
Bethlehem, PA. 18015, U.S.A

ABSTRACT

This paper introduces a Medium Access Control (MAC) protocol for Underwater Acoustic Network (UAN) where one of the transmitter nodes equipped with a Deep Reinforcement Learning (DRL) agent learns the communication environment and adapts its transmission policy to maximize the network throughput. In contrast to a radio frequency (RF) wireless network where the propagation delay is ignored, the UAN experiences significant propagation delays in both transmission from source to sink and feedback from sink to source. When co-existing with a Time Division Multiple Access (TDMA) node or a slotted Aloha node, the DRL node learns to take advantage of the difference in propagation delays and occupy the spare time slots of the network to achieve minimum collision in the UAN environment. Computer simulation results demonstrate the performance gain of using the DRL agent in both synchronous and asynchronous timing models.

CCS CONCEPTS

• **Networks** → **Network protocol design**; *Link-layer protocols*.

KEYWORDS

Deep Reinforcement Learning, Medium access control, Underwater Acoustic Networks.

ACM Reference Format:

Xuan Geng and Y. Rosa Zheng. 2019. MAC Protocol for Underwater Acoustic Networks Based on Deep Reinforcement Learning. In *WUWNET'19: International Conference on Underwater Networks & Systems (WUWNET'19)*, October 23–25, 2019, Atlanta, GA, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3366486.3366542>

1 INTRODUCTION

A Medium access control (MAC) protocol manages the resources in a wireless network so that multiple communication nodes can effectively share the common wireless channels. Design of MAC protocols in Underwater Acoustic Networks (UAN) is particularly challenging because of three key aspects: 1. the UAN has very limited spectrum which spans only up to a few hundred kilo-Hertz; 2.

acoustic waves suffer from huge propagation delays as the nominal sound speed in water is 1500 m/s in contrast to the electromagnetic wave speed of 3×10^8 m/s; 3. the low data rate and low reliability in the physical layer also put stringent constraints on the MAC design. Since the spectrum is limited in UAN, most of the existing MAC protocols are time shared.

Several classic time shared MAC protocols for UAN were investigated in the past years [1–4]. The classic Aloha and slotted Aloha protocols have been used [1] in UAN. The authors in [2] proposed an ordered carrier sensing multiple access (CSMA) combining the concepts of round-robin scheduling and CSMA. To solve the hidden nodes problem in the CSMA, a protocol named multiple access collision avoidance (MACA) was presented in [3] where the RTS (Request To Sent) / CTS (Clear To Send) packets were employed. Due to the different packet delays, the collision avoidance is ineffective in MACA. To overcome this problem, the protocol known as Floor Acquisition Multiple Access (FAMA) was designed in [4] where the packet length was increased to cover the longest delay in the network to avoid collision.

In recent years, intelligent algorithms have been introduced into MAC protocol design of radio frequency (RF) networks, in which the nodes capture the characteristics of surrounding environment and learn to maximize the network performance. A reinforcement learning (RL)-MAC was introduced in [5], which adjusts the frame active time and duty cycle by a Q -learning process. The authors in [6] proposed an Aloha- Q method where the time slots in one frame are chosen optimally also by a Q -learning process. The work in [7] applied deep reinforcement learning (DRL) to capture the free time slots of the heterogeneous network to achieve optimal throughput. Recently, the Artificial Intelligence (AI) also makes its way into UAN MAC design. In [8], the author proposed a slotted CSMA based RL to extend the lifetime of underwater wireless sensor networks (UWSN). The work in [9] evaluated the performance of the Aloha- Q for UAN and proposed asynchronous operation to improve channel utilization.

Considering underwater environment property, we propose a MAC protocol using DRL in time slotted system. We adopt the DRL approach for RF wireless networks in [7] to UAN by incorporating the propagation delay of each node in the network. The collision test in UAN is introduced, which takes propagation delays and attenuation into account. Three UAN time models are considered: one synchronous and two types of asynchronous models, which lead to different collision scenarios and thus determine the learning outcomes. Computer simulation results show that the proposed DRL method can take advantage of the difference in propagation delays among the nodes and maximize the network throughput in both

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WUWNET'19, October 23–25, 2019, Atlanta, GA, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7740-9/19/10...\$15.00

<https://doi.org/10.1145/3366486.3366542>

synchronous model and asynchronous time model-I, but is ineffective in asynchronous time model-II due to limitations in learning the random start time. This is a topic for future improvement.

2 UAN MEDIUM ACCESS MODEL

Consider a UAN with multiple source nodes transmitting packets to a central sink. At least one agent node uses the DRL-based MAC protocol, and other nodes are assumed to use either time division multiple access (TDMA) or slotted Aloha protocols. The source nodes are randomly scattered around the sink node in the 3D underwater space, where the maximum distance between a source node and the sink node is D_{max} .

Due to slow propagation of sound in water, the propagation delay of a transmission may be comparable to one packet duration and collision is highly dependent on the spatial distribution of the nodes. Therefore, UAN MAC protocols may choose different time models, as shown in Fig. 1. The first model is synchronous time model where all packet transmissions are assumed to start at the beginning of each time slot and finish within this time slot. The length of one time slot has to cover the maximum propagation delay and the packet length. That is

$$t_{slot} \geq D_{max}/v + T_{packet} \quad (1)$$

where $v = 1500$ m/s is the nominal sound speed in water, and T_{packet} represents one packet duration. This model requires time synchronization among all nodes, and the time slot has to be long enough to accommodate the node at D_{max} .

Figure 1: Time models in time-shared UWA MAC

The two asynchronous time models share the transmission time with more flexibility. The asynchronous time model-I allows each source node a random start time, denoted by $t_s[n]$ ($n \in N$), from the beginning of the time slots for the n -th node, but each node keeps the same start time in all time slots. The synchronous time model and the asynchronous model-I are similar to the slotted Floor Acquisition Multiple Access (FAMA)[10]. The asynchronous time model-II is truly asynchronous as each source node may transmit at different random start time, represented by $t_s[n, t]$, for the n -th node in any time slots. In both asynchronous models, the length of

the time slots may be shorter than the maximum propagation delay. Therefore, if designed properly, these time models can dramatically improve the network time efficiency. In theory, the starting time $t_s[n]$ or $t_s[n, t]$ may be uniformly distributed in $[0, t_{slot}]$, but in our simulation, a time resolution Δt is selected for reduced complexity.

3 MAC PROTOCOL BASED ON DRL

In our reinforcement learning scheme, the agent interacts with the environment in discrete time steps. let t denote the time step, and let a_t and s_t be the action and the state of the environment, respectively. let r_t denote the reward when the agent in state s_t takes an action a_t . Consequently, the environment moves to the new state s_{t+1} at time $t+1$. The agent then obtains one experience sample (s_t, a_t, r_t, s_{t+1}) . With sufficient experience samples, the agent may learn an optimal decision policy to maximize the long-term accumulated reward. Our goal is to maximize the total throughput in the UAN, which is defined as the average reward of the agent. We apply the agent-environment model proposed in [7] to the UAN to capture the free time slots of the network through DRL. Since we only consider one DRL node as the source node, the actions of the node include $a_t = \text{Transmit}$ or $a_t = \text{Wait}$, but no Receive . According to the actions, the sink will observe $c_t = \text{Transmit Success}$, $\text{Transmit Collision}$ and Idle . Combining the observations with actions, the state of the network at time t is defined as the action-observation pair $\bar{s}_t = \{a_t, c_t\}$. There are a total of 5 states, as shown in Table 1. To effectively adapt to the changing environment, the agent keeps the state history for T_N time instants. Therefore, the states over the time window T_N is defined as

$$s_t = \{\bar{s}_{t-T_N+1}, \bar{s}_{t-T_N+2}, \dots, \bar{s}_t\}$$

The reward is defined as $r_t = 1$ if transmission is a success, and $r_t = 0$ if collision occurs or nodes are idle. The definitions are summarized in Table 1.

Table 1: Agent-environment definitions

Action (a_t)	<i>Transmit(T), Wait(W)</i>
Observation (c_t)	<i>Success(S), Collision(C), Idle(I)</i>
State ($\bar{s}_t = \{a_t, c_t\}$)	<i>(T, S) (T, C) (W, S) (W, C) (W, I)</i>
States over a window (s_t)	$\{\bar{s}_{t-T_N+1}, \bar{s}_{t-T_N+2}, \dots, \bar{s}_t\}$
Reward (r_t)	$r_t = 1$ if $c_t = S$; $r_t = 0$ if $c_t = C$ or I

The DRL node will utilize the Q -learning — a model-free reinforcement learning algorithm. When action a_t is taken at the state s_t under the policy π , the Q -value is represented by $Q^\pi(s_t, a_t)$. The core of the algorithm is a value iteration update by weighted average of the past Q -value and the future information [11], which is

$$Q_t(s_t, a_t) \leftarrow (1 - \alpha)Q_t(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q_t(s_{t+1}, a') \right] \quad (2)$$

where α is the learning rate and γ is the discount factor to determine the importance of the future value. The last term in (2) is the maximum Q -value chosen from all actions a' of the next state s_{t+1} .

Generally, we use Q -table, represented by an matrix, to store and update the Q -value in each step. However, it is quite difficult to deal

with the Q -table if the number of states is huge. Therefore, it is a good way to approximate the Q -value by use of an function, which is the main idea of DRL. Unlike the update equation in (2), the DRL uses deep neural network (DNN) to approximate the Q -value by $Q(s_t, a_t; \theta) \approx Q^*(s_t, a_t)$, where θ is the weights of the DNN and $Q^*(s_t, a_t)$ is the Q -value obtained by optimal policy. To train the DNN and obtain θ , the loss function is defined as

$$L(\theta) = [y(r_t, s_{t+1}) - Q(s_t, a_t; \theta)]^2 \quad (3)$$

where $y(r_t, s_{t+1})$ is the target output of the neural network. For time t , the target value comes from the Q -learning. That is

$$y(r_t, s_{t+1}) = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta) \quad (4)$$

where r_t is the current reward feedback from the environment, and $\max_{a'} Q(s_{t+1}, a'; \theta)$ is the maximum Q -value output by the DNN network with weights θ . As a result, the target value and loss function are obtained from (3) and (4), and the network weights θ is updated by back propagation. The activation function is ReLU, and the Stochastic Gradient Descent (SGD) is used to obtain DNN weight parameters.

This kind of DRL is unstable if a nonlinear function approximation is used. The instability is caused by the correlations of the observation sequences, and small updates to Q may significantly change the policy and the data distribution. As a result, two key ideas are developed in [12] to address the instability problem. First, the “experience replay” is used to store the agent’s experiences, $e_t = (s_t, a_t, r_t, s_{t+1})$, into a replay memory $U = \{e_1, \dots, e_t\}$. The experience samples are selected randomly from U to update the parameters. The advantage of experience replay is to randomize over the data, thus remove the correlations of the continuous data. Second, the agent updates the target network parameters periodically, thus reduce the correlation between Q -value and the DNN target output in (4). Then the loss function is revised as

$$L(\theta) = \sum_{(s, a, r, s') \in U} (y(r, s') - Q(s, a; \theta))^2 \quad (5)$$

$$y(r, s') = r + \gamma \max_{a'} Q(s', a'; \hat{\theta}) \quad (6)$$

where $\hat{\theta}$ is the weights used to compute the target network parameter which only update θ in every C steps. In other words, the parameter $\hat{\theta}$ of the target DNN remains unchanged for C consecutive time steps, thus reduce the instability. We summarize the DRL algorithm applied to UAN in Algorithm 1. Please note that the time instant index t means one time slot as shown in Fig. 1, and each time slot is divided into smaller steps Δt in the computer simulation.

To account for the long propagation delay in UAN, we design a collision detection function to model the collision in underwater environment. The sink declares a collision when at least two packets are received with power greater than zero and with packet overlay in time. When the waveform of any node arrives at the sink, the sink begins to perform collision detection procedure which is detailed in Algorithm 2. In the algorithm, we add a condition $\delta > T_{\text{packet}} + t_c$ to ensure that the collision is counted only once within one packet duration. After one packet passes the sink, the sink is ready to detect the next collision.

Algorithm 1 MAC protocol for UAN based on DRL

```

Initialize  $\alpha, \gamma, \epsilon, T_N, C$ , replay memory  $U$ 
Initialize  $Q$  and  $\hat{Q}$  with random weights  $\theta, \hat{\theta}$ 
Initialize state randomly
for each time slot ( $t$ ) do
    Input  $s_t$  into DNN to output  $Q$ -value;
    Select an action  $a_t$  with  $\epsilon$ -greedy strategy;
    Input action  $a_t$  into CollisionDetection() and compute rewards
    by  $r_t := 1$  if  $c_t == < S >$ ,  $r_t := 0$  if  $c_t == < C >$  or  $c_t == < I >$ ;
    Compute the next state  $s_{t+1}$ ;
    Store experience  $e_t = (s_t, a_t, r_t, s_{t+1})$  into replay memory  $U$ ;
    Select random sample minibatch of  $e_j = (s_j, a_j, r_j, s_{j+1})$  from  $U$ ;
    if  $s_{j+1}$  is the terminal state then
         $y(r_j, s_{j+1}) := r_j$ 
    else
         $y(r_j, s_{j+1}) := r_j + \gamma \cdot \max_{a'} Q(s_{j+1}, a'; \hat{\theta})$  computed by (6);
    end if
    Compute loss function:  $L(\theta) = (y(r_j, s_{j+1}) - Q(s_j, a_j; \theta))^2$  by (5);
    Perform Gradient Descent on  $L(\theta)$  to update  $\theta$  in DNN;
    if  $(t/C) == 0$  then
        Copy current network to target network:  $\hat{\theta} := \theta$ ;
    end if
     $s_t := s_{t+1}$ .
end for

```

To apply the DRL algorithm to the asynchronous time models, we need to revise the simulated system time to implement the random start transmission time of two models. We describe the simulation implementation of asynchronous models briefly in Section 4.

4 SIMULATION RESULT

The simulation assumes that the network has one agent node using DRL method, and the other source nodes use either TDMA or Aloha protocol. All source nodes use 1 kbps data rate and 144 bits for one packet length. Considering sound speed and maximum distance of the nodes, the time slot is determined by (1), and we use $t_{\text{slot}} = 0.4$ s in the simulation. The hyper parameters of DNN are defined as $\alpha = 0.1$, $\gamma = 0.9$, $\epsilon = 0.1$, $U = 500$, $C = 200$ and $T_N = 10$.

To apply the DRL based MAC protocol in asynchronous time model-I, we generate a random start time t_s for each node, and the n th node takes an action only when the system time δ matches $t_s[n]$ in each time slot t . The sink performs the collision detection in Algorithm 2 after transmission. Meanwhile, the agent performs the learning process in Algorithm 1 and the time vector is updated for each node. The overall simulation loop is based on the time resolution Δt , and the time slot t is an integer multiple of Δt . We choose $\Delta t = 0.01$ s and $t_{\text{slot}}/\Delta t = 40$ in this paper. When a transmission starts in time slot t at a start time $t_s[n]$, it is possible that the collision or success occurs in time slot $t + i$ with $i = 0, 1, \dots, \lceil \frac{\tau_{\text{max}}}{t_{\text{slot}}} \rceil$, where $\tau_{\text{max}} = \max(\tau[n]) + t_s[n]$. The reward mechanism is also different from the RF networks because the transmission in slot t is only successful or collided in $t + i$, therefore the reward is available

Algorithm 2 Collision due to transmissions in time slot t

Function CollisionDetection ()

Constants: locations of source nodes and sink node, number of source nodes N , length of a packet T_{packet} and time resolution Δt .

Inputs: action of the agent a_t , actions of other source nodes $\tilde{a}_t$.

Initialize time storage: $t_c := 0$;

Compute propagation delay from each source node n to sink: $\tau[n] := D_n/v$, $n \in N$;

Set maximum delay $\tau_{max} = \max(\tau[n]) + t_s[n]$, ($n = 1 : N$);

for $\delta = 0 : \Delta t : \tau_{max}$ **do**

 Compute the wave front and wave rear at time instant δ : $w_f[n, \delta], w_r[n, \delta]$, $n \in N$;

 Compute wave radius: $r_d[n, \delta] := (w_f[n, \delta] + w_r[n, \delta])/2$;

 Compute wave power: $p[n, \delta] := 10^{-r_d[n, \delta]/1000}$;

if $a_t == < T >$ or $\tilde{a}_t == < T >$ **then**

if $\tau[n] < \delta < \tau[n] + T_{packet}$ **then**

 Collision test:

if $p[n, \delta] > 0$ & $p[n, \delta] - p[n', \delta] > 0$, for any $(n, n') \in N$ & $\delta > T_{packet} + t_c$ **then**

 Collision: $c_t := < C >$;

$t_c := \delta$;

else

 Success: $c_t := < S >$;

end if

end if

else if $a_t == < W >$ & $\tilde{a}_t == < W >$ **then**

 Idle: $c_t := < I >$.

end if

end for

Return: c_t

in slot $t + i$. We assume that the agent node knows the reward at $t + i$ which ignores the feedback time delay from sink to source node. For asynchronous time model-II implementation, the random start time $t_s[n, t]$ changes for each time slot t of n -th node, and the rest of the simulation setup is the same as that of asynchronous time model-I.

Here the TDMA node means that the node transmits packet at the fixed time slots (denoted by K) in the whole time slots (denoted by M), and the left time slots ($K - M$) are free. All of its transmission start strictly at the beginning of each time slot. The Aloha node is denoted by q -Aloha in this paper. For the synchronous and asynchronous time model-I, q -Aloha node means the node occupies the time slots with q -probability to transmit packets, but the start time is still at the beginning of each time slot just like the TDMA node. We denote it as q -Aloha-I. For the asynchronous time model-II, q -Aloha node remains to occupy the time slots with q -probability, but starts transmission at random time within each time slot. We take it as q -Aloha-II. The q -Aloha-I is actually a special case of q -Aloha-II.

We first evaluate the throughput performance of synchronous time model. We use the average rewards of each node to evaluate the throughput. Fig. 2a shows the throughput of a network consisting of a TDMA node and a DRL node. We assume that the TDMA node occupies K time slots out of a total $M = 10$ time slots, and

the simulation of M slots is repeated 10^3 times. When the distance of agent-sink ($D1$) and TDMA-sink ($D2$) is 150 m and 424 m, respectively, it shows that even though the throughput of the TDMA node increases with the increase of K , the throughput of the agent node doesn't reduce. It remains in high throughput around 0.995. The reason is that the distance difference $\Delta D = D2 - D1$ is large enough to allow the agent node to avoid collision with the TDMA node in each time slot. Thereby the agent node learns to occupy all of the time slots when the TDMA node does not occupy. In contrast, when we set the distances of $D1$ and $D2$ equal to 200 m, or $\Delta D = 0$, then the collision will occur if TDMA node takes transmit action. Therefore, only free time slots not used by TDMA node will be captured by agent node, resulting in the lower throughput and no distance gain being exploited. We derive the total optimal throughput of network as

$$R_{max} = \begin{cases} 1 + R_{td}, & \text{if } \Delta D/v > T_{packet} \\ 1, & \text{if } \Delta D/v \leq T_{packet} \end{cases} \quad (7)$$

where $R_{td} = K/M$ with $K = 1, 2, \dots, 9$ is the optimal throughput of one TDMA node. The simulation results in Fig. 2a confirms that the UAN approaches R_{max} .

Figure 2: Throughput in synchronous time model

Fig. 2b shows the results of a network that assumes one agent node and one q -Aloha-I node. The total number of time slots simulated is 10^4 . Similar to the Fig. 2a, when the distances of agent-sink ($D1$) and (q -Aloha-I)-sink ($D3$) are 150 m and 424 m, respectively, the agent node can capture all of time slots regardless of the time slots being used by q -Aloha-I or not. When $D1 = D3 = 200$ m, the agent node chooses to occupy all time slots when $q < 0.5$ because the q -Aloha-I node transmits packets in relatively low probability. In contrast, if the q -Aloha-I node transmits with a relatively large probability ($q \geq 0.5$), then the agent will give up transmission and leave most time slots to the q -Aloha-I node. The total optimal throughput in this case is derived as

$$R_{max} = \begin{cases} 1 + q & \text{if } \Delta D/v > T_{packet} \\ q & \text{if } \Delta D/v \leq T_{packet}, q \geq 0.5 \\ 1 - q & \text{if } \Delta D/v \leq T_{packet}, q < 0.5 \end{cases} \quad (8)$$

The simulation results in Fig. 2b still approach the optimal throughput.

The throughput in asynchronous time model-I with TDMA and q -Aloha-I node are shown in Fig. 3a and Fig. 3b, respectively. It is assumed that the agent begins to transmit at 0 s in both cases, and the start transmission time $t_s[n]$ of the other nodes is randomly

Figure 3: Throughput of an UAN consisting of one DRL node + one traditional node in asynchronous time models.

selected from $[0, 2]$ s. The simulation results demonstrate that the throughput of each case is either 1 or $1 + R_{td}$ in (7). The asynchronous start transmission time and different locations of two nodes imply space-time uncertainty that leads to different propagation delays. The relationship between propagation delay and T_{packet} determines the throughput in (7). Therefore, the throughput may be changed. Similar results are demonstrated in the case of coexisting with q -Aloha-I node. The throughput almost achieves optimal value depended on the asynchronous transmission start time.

For the asynchronous time model-II, we simulate the throughput performance, and the results are shown in Fig. 3c. The distances of agent-sink $D1=200$ m and (q -Aloha-II)-sink $D4=424$ m. The start transmission time of all nodes $t_s[n, t]$ are randomly chosen within one time slot. The throughput results demonstrate that the agent cannot achieve optimal throughput. It cannot capture the most free time of the network because of the random asynchronous start transmission time. Therefore, the DRL with the defined agent-environment in this paper is ineffective for this case. In order to use time more flexibly, a new agent-environment model shall be developed to let the agent learn the most suitable start transmission time, which will avoid collision with the q -Aloha-II node. This will be the future work.

5 CONCLUSION

This paper has applied DRL to MAC protocol design for underwater acoustic wireless networks where a new agent-environment model is developed to consider the underwater acoustic transmission property. Transmission success or collision is then based on the node distances to the sink, propagation speed, and the synchronous or asynchronous transmission models in the UAN. Computer simulations implement the DRL based MAC protocol in the synchronous time model and the asynchronous time model-I, but has trouble to allow the DRL agent fully capture the free time slots in asynchronous time model-II, because the agent lacks the ability to learn the starting transmission time. This work can be improved in the future by redefining the agent-environment model that incorporates the start time as part of the action.

ACKNOWLEDGMENTS

The work of X. Geng is funded by National Natural Science Foundation of China under U1701265 and 61401270. The work of Y.R. Zheng is supported in part by the National Science Foundation of USA under CPS-1853257. The authors would like to thank Dr. Mohammendhossein Behgam for sharing the matlab simulation program that accounts for UAN power attenuation and propagation delay.

REFERENCES

- [1] Joon Ahn, Affan Syed, Bhaskar Krishnamachari, and John Heidemann. Design and analysis of a propagation delay tolerant aloha protocol for underwater networks. *Ad Hoc Networks*, 9(5):752–766, 2011.
- [2] Yin-Jun Chen and Hao-Li Wang. Ordered CSMA: a collision-free MAC protocol for underwater acoustic networks. In *OCEANS 2007*, pages 1–6. IEEE, 2007.
- [3] Phil Karn. MACA-a new channel access method for packet radio. In *ARRL/CRRL Amateur radio 9th computer networking conference*, volume 140, pages 134–140. London, Canada, 1990.
- [4] Chane L Fullmer and JJ Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *ACM SIGCOMM computer communication review*, volume 25, pages 262–273. ACM, 1995.
- [5] Zhenzhen Liu and Itamar Elhanany. RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks. *International Journal of Sensor Networks*, 1(3-4):117–124, 2006.
- [6] Yi Chu, Paul D Mitchell, and David Grace. ALOHA and Q-learning based medium access control for wireless sensor networks. In *2012 International Symposium on Wireless Communication Systems (ISWCS)*, pages 511–515. IEEE, 2012.
- [7] Yiding Yu, Taotao Wang, and Soung Chang Liew. Deep-reinforcement learning multiple access for heterogeneous wireless networks. *IEEE Journal on Selected Areas in Communications*, 37(6):1277–1290, 2019.
- [8] Lu Jin and Defeng David Huang. A slotted CSMA based reinforcement learning approach for extending the lifetime of underwater acoustic wireless sensor networks. *Computer Communications*, 36(9):1094–1099, 2013.
- [9] Sung Hyun Park, Paul Daniel Mitchell, and David Grace. Performance of the ALOHA-Q MAC protocol for underwater acoustic networks. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 189–194. IEEE, 2018.
- [10] Marcal Molins and Milica Stojanovic. Slotted FAMA: a MAC protocol for underwater acoustic networks. In *OCEANS 2006-Asia Pacific*, pages 1–7. IEEE, 2007.
- [11] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.