Journal of
Marine Science
and Engineering

MDPI

Article

Deep Reinforcement Learning Based Time-Domain Interference Alignment Scheduling for Underwater Acoustic Networks

Nan Zhao [1], Nianmin Yao [1,2,*], Zhenguo Gao [3,4,*] and Zhimao Lu [1]

[1] Department of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China; nanzhao@mail.dlut.edu.cn (N.Z.); zhimaolu@163.com (Z.L.)
[2] Ningbo Institute of Dalian University of Technology, No. 26 Yucai Road, Jiangbei District, Ningbo 315016, China
[3] Department of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
[4] Key Laboratory of Computer Vision and Machine Learning (Huaqiao University), Fujian Province University, Xiamen 361021, China
* Correspondence: lucos@dlut.edu.cn (N.Y.); gzg@hqu.edu.cn (Z.G.)

Abstract: Message conflicts caused by large propagation delays severely affect the performance of Underwater Acoustic Networks (UWANs). It is necessary to design an efficient transmission scheduling algorithm to improve the network performance. Therefore, we propose a Deep Reinforcement Learning (DRL) based Time-Domain Interference Alignment (TDIA) scheduling algorithm (called DRLSA-IA). The main objective of DRLSA-IA is to increase network throughput and reduce collisions. In DRLSA-IA, underwater nodes are regarded as agents of DRL. Nodes intelligently learn the scheduling by continuously interacting with the environment. Therefore, DRLSA-IA is suitable for the highly dynamic underwater environment. Moreover, we design a TDIA-based reward mechanism to improve the network throughput. With the TDIA-based reward mechanism, DRLSA-IA can achieve parallel transmissions and effectively reduce conflicts. Unlike other TDIA-based algorithms that require enumeration of the state space, nodes merely feed the current state to obtain the transmission decision. DRLSA-IA solves the problem of computational expense. Simulation results show that DRLSA-IA can greatly improve the network performance, especially in terms of throughput, packet delivery ratio and fairness under different network settings. Overall, DRLSA-IA can effectively improve network performance and is suitable for ever-changing underwater environments.

Keywords: underwater acoustic networks; medium access control protocol; deep reinforcement learning; time-domain interference alignment

check for updates

Citation: Zhao, N.; Yao, N.; Gao, Z.; Lu, Z. Deep Reinforcement Learning Based Time-Domain Interference Alignment Scheduling for Underwater Acoustic Networks. *J. Mar. Sci. Eng.* **2022**, *10*, 903. https://doi.org/10.3390/jmse10070903

Academic Editor: Myung-Il Roh

Received: 16 May 2022
Accepted: 28 June 2022
Published: 30 June 2022

1. Introduction

The vast ocean is rich in resources, such as metals, energy, and biological assets. Underwater Acoustic Networks (UWANs) provide technical support for the development and utilization of marine resources, which has attracted widespread attention [1,2]. UWANs consist of multiple underwater nodes. The underwater nodes cooperate to complete important underwater tasks, such as pollution monitoring, geographic exploration, and emergency rescue [3,4].

Different from terrestrial wireless networks that use radio waves to communicate, UWANs rely on acoustic communication [5,6]. However, the propagation speed of acoustic signals is approximately 1500 m/s, five orders of magnitude slower than that of radio waves. The slow propagation speed of acoustic signals leads to large propagation delays of UWANs. To this end, the terrestrial Medium Access Control (MAC) protocols that ignore the effects of propagation delays have poor performance in UWANs. It is important to design MAC protocols that are suitable for UWANs [7,8].

Several studies have focused on the design of underwater MAC protocols [9–11]. A limitation of these works is the assumption that the underwater environment is static and propagation delays between nodes do not change. However, the location of underwater nodes is constantly changing due to internal waves and other factors. The propagation delays between nodes vary with the location of the nodes. Consequently, the proposed underwater MAC protocol needs to fit well in the highly dynamic underwater environment.

Time-domain Interference Alignment (TDIA)-based underwater MAC protocols have attracted wide attention due to their excellent performance [12,13]. These protocols regard large propagation delays of UWANs as an advantage and achieve parallel transmissions using large propagation delays. The protocols usually adopt enumeration or greedy methods to obtain optimal scheduling [14]. However, these protocols are computationally expensive in general due to the massive state space. For example, in an underwater node collaboration network with $N-$nodes, the complexity of the enumerated decision space is N^3, and the state space is even larger. The protocols are not feasible in terms of run-time and computation when the number of nodes slightly increases. Therefore, we need to design a scheduling algorithm that does not need to enumerate the decision space and state space.

To address the above issues, we design a Deep Reinforcement Learning (DRL)-based TDIA scheduling algorithm for UWANs (named DRLSA-IA). We regard the underwater nodes as the DRL agents. In particular, nodes learn transmission scheduling as they continuously interact with the environment. Therefore, DRLSA-IA adapts to the highly dynamic underwater environment. Furthermore, instead of enumerating the state space, nodes merely feed the observed state to the trained algorithm to obtain transmission decisions. Hence, DRLSA-IA does not face computational restriction. To improve the network throughput, we design a novel TDIA-based reward mechanism. With the reward mechanism, the DRLSA-IA can align interference in time slots that have collided or interfered. In this way, the number of conflicting and interfered time slots can be effectively reduced. Additionally, nodes can use the free time slots to transmit messages and receive expected messages. Parallel transmission of messages also can be implemented. Therefore, DRLSA-IA can effectively improve the network throughput and reduce collisions.

The main contributions of this paper are as follows:

- We propose a deep reinforcement learning based TDIA scheduling algorithm (called DRLSA-IA). Nodes learn efficient transmission scheduling while continuously interacting with the environment without enumerating the state space. Therefore, DRLSA-IA is suitable for the highly dynamic underwater environment and has no computational restriction.
- We put forth a TDIA-based reward mechanism to increase the network throughput. With the reward mechanism, nodes can take full advantage of large propagation delays to achieve parallel transmissions.
- Considering large propagation delays of UWANs, we use an Long Short-Term Memory (LSTM) layer in DRLSA-IA. LSTM maintains an internal state and aggregates observations over time. Thus, DRLSA-IA can estimate the true state using history information.
- We conduct extensive simulation experiments to demonstrate the performance of the proposed algorithm. DRLSA-IA outperforms in terms of throughput, packet delivery ratio, and fairness compared to the other MAC protocols.

The remaining sections of the paper are structured as follows. We review the underwater MAC protocols in Section 2. The system model and problem statement are described in Section 3. Section 4 presents the proposed DRLSA-IA in detail. Section 5 shows the simulation results. Finally, we conclude the paper in Section 6.

2. Related Works

Recently, there has been a lot of research work on underwater MAC protocols [15–17]. The MAC protocols of UWANs fall into two categories: schedule-based MAC protocols and contention-based MAC protocols.

The schedule-based MAC protocols mean that the channel resources are allocated to each node according to a predetermined mechanism. For example, Time Division Multiple Access (TDMA)-based protocols divide time into several time slots. Nodes use pre-allocated time slots to transmit messages. Frequency Division Multiple Access (FDMA)-based protocols divide the available frequency band of acoustic channels, and each user owns their unique frequency channel. Code Division Multiple Access (CDMA)-based protocols allow multiple users to transmit data simultaneously, and the receiving node uses pseudo-random code to distinguish different sending nodes.

Contention-based protocols are mainly divided into random access protocols and handshake-based protocols. Among them, the ALOHA protocol [18] is a classic random access protocol. As long as there is a message, the node will send it immediately. Thus, collisions occur frequently. Slotted-Aloha [19] is an improved version of pure ALOHA with the time slot. In Slotted-Aloha, the node will start to transmit a message in the next time slot. This algorithm prevents nodes from sending messages randomly and reduces message conflicts to a certain extent. Random access protocols lead to frequent conflicts in networks with high traffic, but it performs better in low-traffic networks due to the simple access mechanism.

To avoid collisions, the handshake-based protocols send control packets to compete for the channel before sending data. However, additional control packets increase network load and network delay. To reduce the number of control packets, Lee et al. [20] proposed a Hybrid Sender and Receiver-initiated (HSR) protocol. In HSR, multiple nodes share control packages by inviting neighbor nodes to join the current handshake.

The time slot-based scheduling algorithms convert channel allocation into time slot scheduling. Nodes use the allocated time slots to transmit or receive messages. Sivakumar et al. [21] proposed an evolutionary genetic node scheduling algorithm called GA. However, they assumed that the network was static. In [14], the authors proposed a packet-level slot scheduling algorithm. The slot scheduling problem was formulated into a combinatorial optimization problem. Similarly, the algorithm assumed that the network was static. In [22], the authors considered the message scheduling problem in a linear topology network and proposed a transmission scheduling algorithm. The limitation of these algorithms is that they are not flexible enough to change with the underwater environment.

In recent years, the studies on resource allocation in wireless sensor networks using reinforcement learning algorithms received considerable interest [23,24]. In UWANs, most of the reinforcement learning-based works focus on routing protocols [25,26]. The studies on the underwater MAC protocol based on reinforcement learning are in their infancy. In [27], the goal was to redesign a protocol (UW-ALOHA-Q) based on the established RL-based protocol (ALOHA-Q). UW-ALOHA-Q was more suitable for UWANs. UW-ALOHA-Q algorithm used the Q-learning algorithm to learn optimal action. However, Q-learning used the Q-table to store the Q-values of action-state pairs. Like [13], UW-ALOHA-Q also faced the computation expense problem.

Different from the above algorithms, we propose a deep reinforcement learning-based scheduling algorithm (called DRLSA-IA). DRLSA-IA learns transmission scheduling as nodes constantly interact with the environment. Therefore, it can adapt to the highly dynamic underwater environment. DRLSA-IA does not need to enumerate the state space. Thus, it is not subject to computationally expensive constraints. We compare DRLSA-IA and classical related algorithms in Table 1.

Table 1. Comparison of typical relevant works.

Research Work	TDMA	TDIA	Adapt to the Dynamic Environment	Solve the Computation Expensive Problems
Slotted-Aloha	✓			
UW-ALOHA-Q			✓	
GA	✓			
DRLSA	✓	✓	✓	✓

✓ indicates that the work has the corresponding ability.

3. System Model and Problem Statement

3.1. System Model

In this paper, we consider an underwater node collaboration network comprising N sensor nodes. As shown in Figure 1, nodes are divided into several clusters depending on the physical proximity of the nodes. The nodes within a cluster communicate with each other to accomplish cooperative tasks. Each node has a unique identity. As shown in Figure 2, the network topology is mesh-type. The transmission range is the entire cluster. Any two nodes can communicate with each other via a shared acoustic channel. We assume the message is backlogged, i.e., a source node always has messages to send to other nodes in the cluster. Each node is equipped with only one transceiver. Thus, only one message can be successfully received by a node at a time. If more than one message simultaneously reaches a node, no messages can be received successfully. This is because these messages are conflict. Except for the source node and the destination node, the message is regarded as interference. In addition, we assume that the node is half-duplex, so the node is incapable of simultaneously sending and receiving messages. The network carries only unicast messages, i.e., each message has a single destination node.

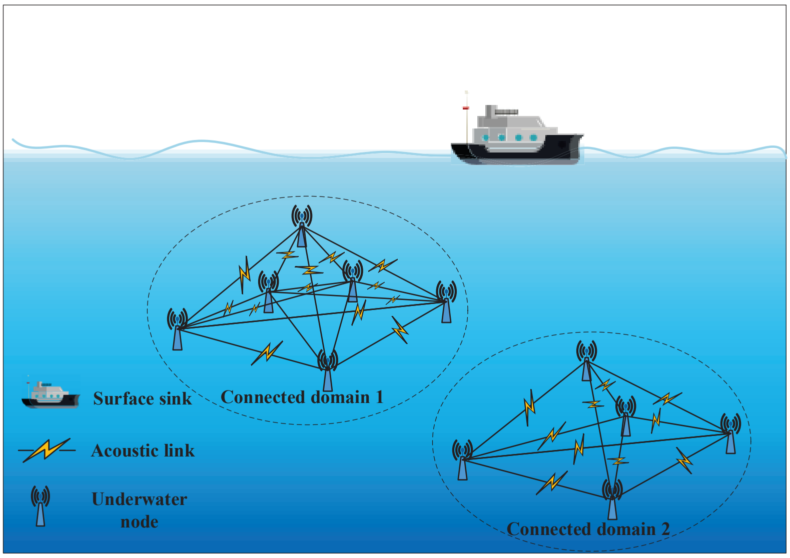

Figure 1. The underwater node collaboration networks.

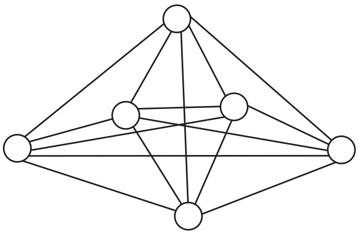

Figure 2. Mesh-type network topology.

We use the TDMA technique. In the time domain, the network channel is divided into times slots of the same length. Nodes independently use single time slots to transmit or receive messages. After the message reaches the destination node, the destination node returns an Acknowledge Character (ACK) packet containing the message reception. All messages are of equal length and all ACK packets are of equal length. Moreover, each time slot corresponds to the time duration for a message transmission plus the time duration for an ACK packet transmission.

Like most scheduling-based MAC protocols, we use the time slot as the unit of propagation delay between nodes, as shown in Equation (1),

$$d_{ij} = \frac{|v_i - v_j|}{v \times \eta}, \tag{1}$$

where i and j denote the nodes in the network. v_i and v_j are the position vectors of node i and node j, respectively. v is the propagation speed of the sound in the water. η denotes the length of a time slot. The propagation delay between node i and node j is d_{ij}. It takes d_{ij} time slots to transmit messages between node i and node j.

$$G = \max_{i,j} d_{ij}, \tag{2}$$

where G is the maximum propagation delay and indicates the network range. The unit of d_{ij} and G is time slot.

We assume that the propagation delay from node i to node j is equal to the propagation delay from node j to node i, i.e., $d_{ij} = d_{ji}$. Due to the non-negligible propagation delay, node i sends a message in slot 1, but it does not reach node j until time slot $t + d_{ij}$. When the message arrives at node j, node j returns an ACK packet with the message reception status. As the ACK packet is very short, we assume that ACK packets do not conflict.

3.2. Problem Statement

In this paper, we aim to design a transmission scheduling algorithm that can improve the network performance. Message conflicts not only severely reduce network throughput, but also cause energy waste. DRLSA-IA is based on the TDMA, so the main task is to reasonably schedule the transmission time slots to eliminate message collisions and increase network throughput.

Due to large propagation delays of UWANs, simultaneous transmissions may be collision-free, while nonsimultaneous transmissions may still collide at the receivers. For example, as shown in Figure 3, both node A and node B transmit messages in time slot 2, but the two messages reach node C in different time slots. The reason is that the propagation delays are different, i.e., $d_{AC} \neq d_{BC}$. The two messages can both be received successfully. However, node A and node D transmit messages in different time slots, and the two messages conflict in time slot 4. Therefore, large propagation delays need to be considered when designing the transmission scheduling algorithm.

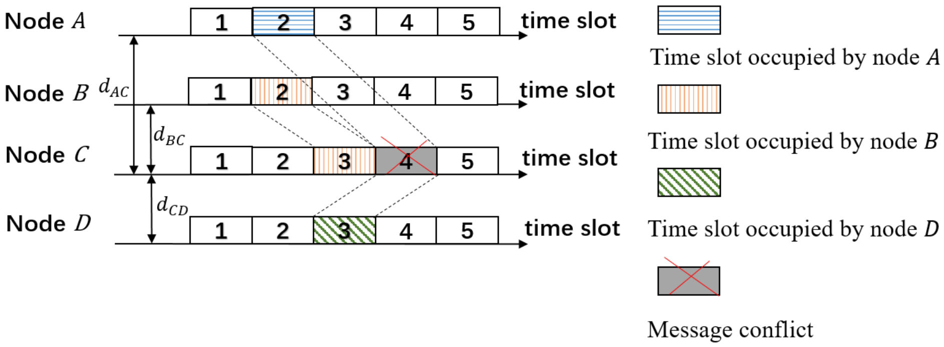
Node A
Node B
Node C
Node D
d_{AC}
d_{BC}
d_{CD}
1
2
3
4
5
time slot
Time slot occupied by node A
Time slot occupied by node B
Time slot occupied by node D
Message conflict

Figure 3. Complex collision situations caused by large propagation delay.

In addition, it is worth noting that the underwater environment is ever-changing and the propagation delays between nodes may be different at each time. For example, as shown in Figure 4, the propagation delays of node A, node B, node D, and node C at time $t1$ are $d_{AC} = 2, d_{BC} = 1, d_{CD} = 1$, respectively. However, due to environmental factors, nodes move, and the propagation delays of node A, node B, node D, and node C at time $t2$ are $d_{AC} = 2, d_{BC} = 3, d_{CD} = 2$ respectively, as shown in Figure 5.

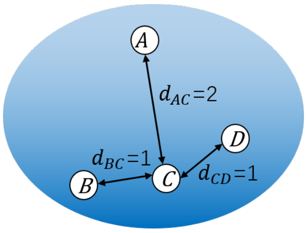

Figure 4. The propagation delays between nodes at time $t1$.

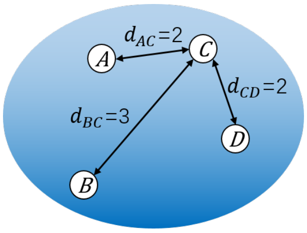

Figure 5. The propagation delays between nodes at time $t2$.

In summary, we need to take into account the above characteristics of UWANs when designing a transmission scheduling algorithm. Additionally, the algorithm produces a schedule that maximizes the network throughput. A schedule **S** determines when each node transmits and receives messages, where S_{it} is an element of **S**. If $S_{it} = j > 0$, then node i transmits a message to node j in time slot t. If $S_{it} = 0$, then node i is idle in time slot t. If $S_{it} = -j < 0$, then node i receives a message from node j in time slot t. The average throughput thr (unit: packets/slots) with period T can be computed from the number of successful transmissions in schedule **S**, as shown in Equation (3),

$$thr = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}(S_{it} > 0), \tag{3}$$

where $\mathbb{I}(X)$ is the indicator function. The function value is 1 if the event occurred, 0 otherwise.

Therefore, we formulate the throughput maximization problem as follows:

$$\max_{\mathbf{S}} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}(S_{it} > 0)\right]. \tag{4}$$

4. DRL Based TDIA Transmission Scheduling Algorithm (DRLSA-IA)

In this paper, we proposed a deep reinforcement learning-based time-domain interference alignment scheduling algorithm (named DRLSA-IA). In DRLSA-IA, nodes learn efficient transmission scheduling in continuous interaction with the environment. Therefore, DRLSA-IA can effectively adapt to the highly dynamic underwater environment. Unlike other algorithms that require enumeration of the state space, DRLSA-IA can obtain the current transmission decision when the observed state is fed to the trained algorithm, so there is no computational restriction.

Specifically, we first describe the transmission scheduling problem in UWANs as a DRL problem. Furthermore, we present our time-domain interference alignment-based reward mechanism to improve the network throughput. Considering large propagation delays and the partially observed network state of UWANs, we proposed a novel DRL algorithm in DRLSA-IA. Finally, we describe DRLSA-IA in detail.

4.1. Formulated as a DRL Problem

The transmission scheduling problem is formulated as a DRL problem. We give the following definitions: agent, action space, state space and objection function.

(a) Agent: We regard the node that uses DRLSA-IA as a DRL agent. There are N nodes in the networks.

(b) Action space: In an underwater node collaboration network, nodes communicate with each other. Each node selects an expected node to send a message or remain idle in each time slot t. Therefore, the action space A of nodes is defined as,

$$\mathbf{A} = \{0, 1, 2, \ldots, N\}. \tag{5}$$

Each node i selects an action $a_i(t)$ from the action space $\mathbf{A}$ in every time slot t, i.e., $a_i(t) \in \mathbf{A}$. If $a_i(t) = 0$, node i will be idle in time slot t. If $a_i(t) = k \neq 0$, node i will transmit a message to node k in time slot t.

(c) State space: Each node i can only observe part of the network state. The network state $S_i(t)$ observed by node i in time slot t consists of all ACK packets received and the actions performed by node i before time slot t, as shown in Equation (6).

$$S_i(t) = \left\{ a_i(t), \left\{ o_j^i(t) \right\}_{j=1}^{N} \right\}_{t=1}^{t}, \tag{6}$$

where $o_j^i(t)$ denotes the ACK packet returned by node j, $o \in \{-1, 0, 1\}$. $o = 1$ indicates that the message is successfully received. $o = 0$ indicates that the message conflicts with an unintended message. $o = -1$ indicates that the message conflicts with an intended message. $\left\{ o_j^i(t) \right\}_{j=1}^{N}$ denotes all ACK packets received by node i in time slot t.

(d) Reward function: To improve the network throughput, we design a TDIA-based reward function. We explain the reward function in detail in Section 4.2.

(e) Objective function: We formulate the transmission scheduling problem as a DRL problem, so the throughput maximization problem can be formulated as follows:

$$\max_{\mathbf{S}} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R(t) \right], \tag{7}$$

where $\gamma \in (0, 1)$ is a discount factor and $R(t)$ is the reward in time slot t. $\mathbf{S}$ is the transmission schedule. The objective function is the expected value of the discounted sum of reward over a time horizon of length T.

A Q-function is used to evaluate the accumulative reward of the system, as shown in Equation (8),

$$Q(S, a) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R(t) \mid S_i(t) = S, a_i(t) = a \right]. \tag{8}$$

The node follows a policy $\mathbf{ß}$ to interact with the environment. The $\mathbf{ß}$ maps states to actions. We aim to find the optimal policy $\mathbf{ß}^*$ that maximizes the Q-function as $\mathbf{ß}^*(S) = \arg\max_{a}\{Q(S, a)\}$. Equation (9) is the optimal Q-function,

$$Q^*(S, a) = \mathbb{E}\left[R(t) + \gamma \max_{a'} Q^*(S', a') | S_i(t) = S, a_i(t) = a \right], \tag{9}$$

where S' and a' are the next state and action after the node takes action $a_i(t)$ when in state $S_i(t)$, respectively.

4.2. Reward Mechanism Based on Time-Domain Interference Alignment

Different from other TDIA-based algorithms that obtain optimal scheduling using the enumeration method, we utilize deep reinforcement learning to achieve the TDIA-based scheduling. In reinforcement learning, the agent judges whether his behavior is good or bad by accepting rewards from the environment, to ensure that the agents can move toward the goal state by choosing behaviors with higher returns. In our algorithm, the goal is to achieve the TDIA-based scheduling. Therefore, we design the TDIA-based reward mechanism.

Time-domain interference alignment means that the interference is aligned in a time slot that has already collided or interfered. In this way, each node can reserve more free time slots to receive and transmit messages. We give an example to intuitively explain time-domain interference alignment, as shown in Figure 6. At time $t3$, the propagation delays between node A, node B, node C, and node D are $d_{AD} = 1, d_{BD} = 2$, and $d_{CD} = 1$, respectively. Node A is expected to send the message m_A to node D. The expected node of the messages m_B and m_C sent by node B and node C is not node D. Thus the two messages m_B and m_C are the interference for node D. The message m_B is sent by node B in time slot 1. The message reaches node D in time slot 3 due to the propagation delay. Thus, time slot 3 is an interfered time slot for node D. To reduce the number of interfered time slots, message m_C preferably also reaches node D in time slot 3. Therefore, message m_C must be transmitted by node C in time slot 1. In order to successfully receive message m_A, message m_A cannot reach node D in time slot 3. Message m_A must be sent by node A in a time slot outside of time slot 2.

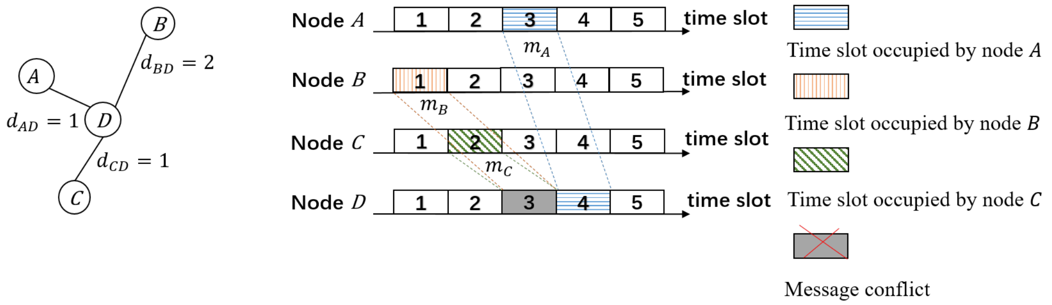

Figure 6. Time-domain interference alignment diagram.

As shown in Equation (10), if the current transmission action $a_i(t)$ follows the TDIA idea, it will be rewarded. In particular, a transmission action $a_i(t)$ will not allow self-transmission, i.e., the source node i and the destination node j cannot be the same. The destination node j successfully receives the message in the receiver time slot. This transmission does not interfere with the ongoing message reception. If the above conditions are satisfied, this transmission satisfies the time-domain interference alignment idea.

$$R(t) = \begin{cases} 0, & \text{if } a_i(t) = j \,\&\, i = j \\ 0, & \text{if } a_i(t) = j \,\&\, o_j^i(t + 2d_{ij}) \neq 1 \\ 0, & \text{if } a_i(t) = j \,\&\, \exists\, k \ s.t.\ o_k^i(t + 2d_{ik}) \neq 0. \\ 1, & otherwise \end{cases} \tag{10}$$

The reward of time slot t is dependent on the observations of time slot $t + 2G$ due to large propagation delays. Considering this issue, we make corresponding improvements in DRLSA-IA and describe DRLSA-IA in detail in Section 4.3.

4.3. The Describe of DRLSA-IA

In this paper, we propose a deep reinforcement learning-based scheduling algorithm, called DRLSA-IA. An illustration of the DRLSA-IA is shown in Figure 7.

Instead of obtaining transmission decisions by enumerating the decision space and state space, we use value function approximation to express the Q-value of each state-action pair, i.e., $Q(S,a) \approx f(S,a,\omega)$. A vector containing the Q-values of all state-action pairs can be obtained when the state is input. We use the deep neural network to fit the value function. The Q-value of each action in each state can be calculated by the deep neural network.

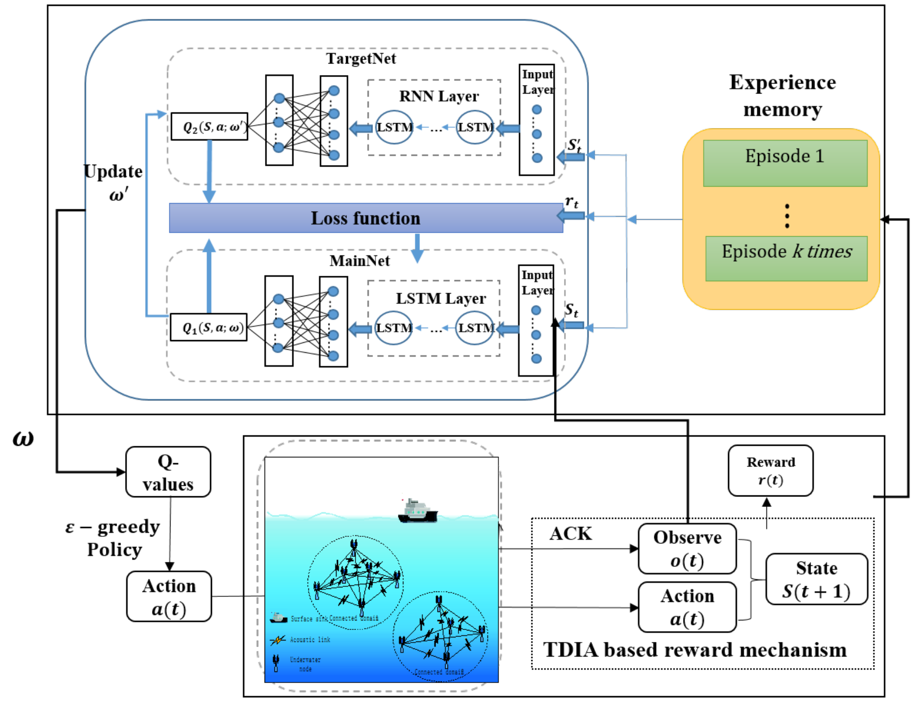

Figure 7. Illustration of the DRLSA-IA algorithm.

To avoid selecting overestimated Q-values that degrade performance, we use two deep neural networks to decouple the selection of the action from the calculation of target Q-values. The two deep neural networks are named as Q_1 and Q_2. Specifically, we use Q_1 to find the action with the largest Q-value and use Q_2 to estimate the target Q-value. Q_1 and Q_2 have the same network structure.

For each node in the network, it is difficult to obtain the global network state. Thus, the class DQN does not perform well in this setting. In DRLSA-IA, we use an Long Short-Term Memory (LSTM) layer to replace the first fully connected layer of the deep neural network. Message transmission scheduling is a sequential decision problem. When making transmission decisions, it is not only based on the currently obtained information, but also needs to refer to previously stored prior information. LSTM maintains an internal state and aggregates observations over time. This gives the network the ability to estimate the true state using the history of the process.

Considering the delayed reward caused by large propagation delays, we collect the *ktimes* episodes at each iteration. All experiences in an episode are used to update. As shown in Equation (11), we create target values for all the episodes. By the LSTM layer, DRLSA-IA can aggregate observations from time slot t to time slot $t+2G$ over time.

$$\begin{aligned} Q_{target}\left(a^j(t)\right) &= R(t)+ \\ &\gamma Q_2\left(S^j(t+1), \arg\max_{a'} Q_1\left(S^j(t+1), a', \omega\right), \omega'\right), \end{aligned} \tag{11}$$

where j is the j-sample experience.

Each node in the network executes DRLSA-IA. The training process of DRLSA-IA is shown in Algorithm 1. Specifically, there are the following several steps: (1) The first step is when the node inputs the current observed state to the eval-net Q_1 and obtains the estimation Q-values of all of the actions. (2) The second step occurs when the node determines to transmit a message to a destination node or wait according to the Q-values of all the actions. The node uses the ϵ-greedy method to select the action. (3) The third step is when the node takes the action and obtains the reward. Then, the network transforms to the next state. (4) The next step is when the node feeds the new network state to the eval-net Q_1 and the target-net Q_2, as well as obtains the Q-values of all the actions. (5) The target Q-value is computed according to Equation (11). After each iteration, the eval-net Q_1 is trained with all experiences of *ktimes* episode, as in Equation (12). After every c iteration, the parameters of the target-net Q_2 are updated.

We use the Mean Squared Error (MSE) loss function to update all parameters ω of the eval-net Q_1 through gradient backpropagation of the neural network, as shown in Equation (12). The node obtains all experiences of *ktimes* episodes and uses these experiences to update the parameters of the model. The node inputs the current state $S^j(t)$ of an experience to the eval-net Q_1 and obtains the estimation Q-values of all of the actions. Then, target Q-values are calculated, as shown in Equation (11). We calculate the distance between the current Q-value and the target Q-value through the MSE loss function. Finally, all the parameters (ω) of the eval-net Q_1 are updated through gradient backpropagation of the neural network.

$$Loss = \frac{1}{T\ ktimes} \sum_{j=1}^{T\ ktimes} \left(Q_{target} - Q_1\left(S^j(t), a^j(t), \omega \right) \right)^2. \tag{12}$$

Algorithm 1 Training Process of DRLSA-IA

Input: Iteration times *Itimes*, episode size *ktimes*, the number of nodes N, the number of time slots T.

Input: discount factor γ, learning rate α, exploration probability ϵ, the updated frequency of target-net Q_2 parameters c.

Output: the parameters of the eval-net Q_1.

```
Initialize eval-net $Q_1$ parameters $\theta$ with random weights $\omega \leftarrow \omega_0$;
Set target-net $Q_2$ parameters $\omega' \leftarrow \omega'_0$;
for Iteration $m = 1, \cdots, Itimes$ do
  for episode $k = 1, \cdots, ktimes$ do
    for timeslot $t = 1, \cdots, T$ do
      for node $i = 1, \cdots, N$ do
        Feed network state $S_i(t)$ into the $Q_1$;
        $Q_1$ generates an estimation of Q-values for all actions $A = \{0, 1, 2, \cdots, N\}$;
        Use $\epsilon$-greedy method to select action $a_i(t)$;
        Take action $a_i(t)$, then update network state $S_i(t+1)$ and obtain reward $R(t)$;
        Feed state $S_i(t+1)$ into the $Q_1$, $Q_2$;
        $Q_1$ and $Q_2$ generate estimations of Q-value, for all available actions;
        Form a target Q-value of $a_i(t)$ by (11);
      end for
    end for
  end for
  Train $Q_1$ with the all experiences in ktimes episode;
  Every $c$ iterations update target Q-Network $Q_2$ parameter $\omega' = \omega$;
end for
```

Each node uses the trained DRLSA-IA to distributely decide on transmission in real-time. In each time slot t, node i obtains observation $o_i(t)$, and then feeds observed state $S_i(t)$ into the trained Q_1. Q-values are generated by Q_1 for all available actions $a_i(t) \in \{0, 1, 2, \ldots, N\}$. Then, node i selects an action according to the ϵ-greedy policy. The network state is updated.

5. Performance Evaluation

To demonstrate the effectiveness of the proposed DRLSA-IA, we carry out numerous simulation experiments. We use the Python programming language to implement and simulate DRLSA-IA based on the TensorFlow framework. We assume that nodes are randomly placed. The details of the simulation parameter settings are summarized in Table 2.

Table 2. Simulation Parameter Settings

Parameter Settings	**Value**
Reward discount factor γ	0.95
Learning rate α	Decay from 0.01 to 0
Exploration probability ϵ	Decay from 0.1 to 0.001
The physical size of the network G	100 time slots
The range of node communication	Entire network
The range of interference communication	Entire network
The number of nodes	10∼100
The network traffic	10∼200 packets/slots
Number of simulation time slots T	100

In addition, the network traffic is defined as the number of messages created every time slot. We compare DRLSA-IA with Slotted-Aloha [19] and GA [21] under the same conditions. Among them, Slotted-Aloha is a classic time-slotted MAC protocol. GA is the latest genetic algorithm-based scheduling algorithm.

5.1. Performance Metrics

We compare DRLSA-IA with the other two algorithms in the same simulation environment. Following the past works, the throughput, packet delivery ratio, and fairness index are used to evaluate the performance of algorithms.

- We use the number of successfully received messages over the period T to calculate the average throughput.
- As shown in Equation (13), the Packet Delivery Ratio (PDR) is calculated by dividing the number of messages created by all nodes (M_c) by the number of messages that were successfully sent (M_s).
$$PDR = \frac{M_s}{M_c}. \tag{13}$$
- The Fairness Index (FI) is used to evaluate the fairness of the node access channel. It is an important metric to measure the performance of the underwater MAC protocol. As shown in Equation (14), we use the Jain's Fairness Index.
$$FI = \frac{(\sum_{i=1}^{N} x_i)^2}{(N \times \sum_{i=1}^{N} x_i^2)}, \tag{14}$$
where N denotes the total number of nodes. The throughput of node i is represented by x_i. The fairness index has a value between 0 and 1. Network fairness rises when the fairness index draws closer to 1. Network fairness decreases as the fairness index gets closer to 0. When the fairness index is equal to 1, accessing the channel is entirely fair.

5.2. Simulation Results

Figure 8 shows the average throughput as a function of the iteration times for the three different episode times. As expected, the average throughput first increases as the number of iterations increases, and then it stabilizes. The environment is constantly changing as the nodes perform transmission actions. Thus, the average throughput fluctuates within a certain range. The greater the number of episodes, the more sample experiences DRLSA-IA collects, and thus the better the learning results and the higher the efficiency.

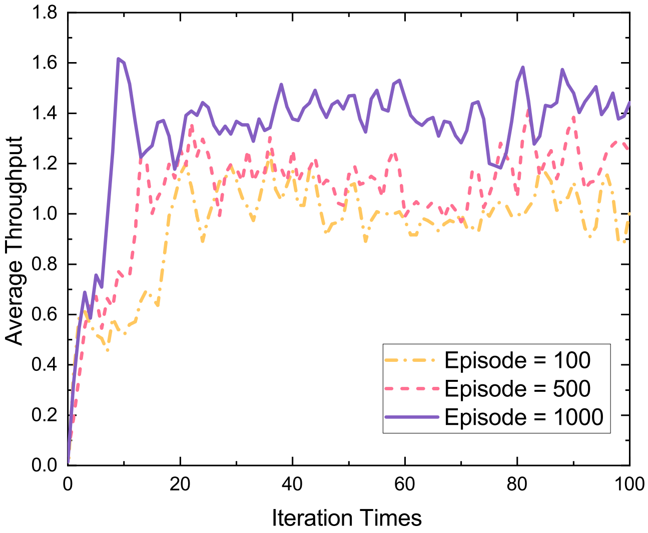

Figure 8. Correlation between the average throughput and iteration times for different episode times.

Figure 9 shows the average throughput as a function of the iteration times with different discount factors. We chose a configuration with 10 nodes, 10 time slots, and 100 episodes. As expected, the larger the discount factor, the higher the average throughput, but the slower the convergence rate. The discount factor γ is used to adjust the impact of short-term and long-term rewards. The value range of the discount factor is $(0, 1]$. The closer the discount factor is to 0, the more the agent cares about short-term returns and the faster the algorithm converges. Otherwise, the agent considers long-term rewards, and the the algorithm converges more slowly. If the agent only focuses on short-term rewards, the algorithm may fall into a local optimum. If the discount factor is set to 1, the algorithm may be stuck in an infinite loop of states and is impossible to converge. In DRLSA-IA, the transmission of a node may not only interfere with ongoing transmissions but also have an impact on future transmissions. Therefore, we need to consider long-term rewards and set the discount factor to 0.95.

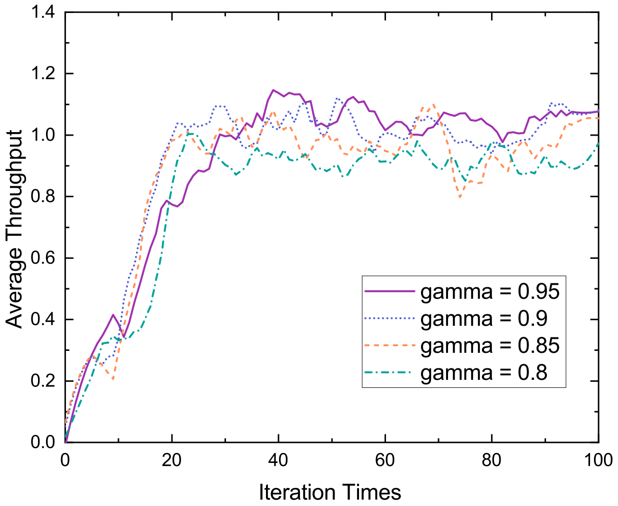

Figure 9. Correlation between the average throughput and iteration times for different discount factors.

We compare the throughput of DRLSA-IA with the other two algorithms with various network traffic. The number of nodes is 10. As shown in Figure 10, the throughput of DRLSA-IA outperforms that of the other two algorithms. Compared with Slotted-ALOHA and GA algorithm, the throughput of DRLSA-IA increase by 55.23% and 19.85%, respectively. Slotted-ALOHA sends the message as soon as there is a message. As a result, the likelihood of conflicts increases as network traffic increases. Slotted-ALOHA cannot reasonably schedule message transmissions, and thus the throughput decreases. GA can reduce conflicts to some extent, but cannot achieve parallel transmission. However, due to TDIA, more than one node can concurrently transmit messages in DRLSA-IA, effectively improving the network throughput. Furthermore, interference is aligned in the interfered time slot, reducing conflicts.

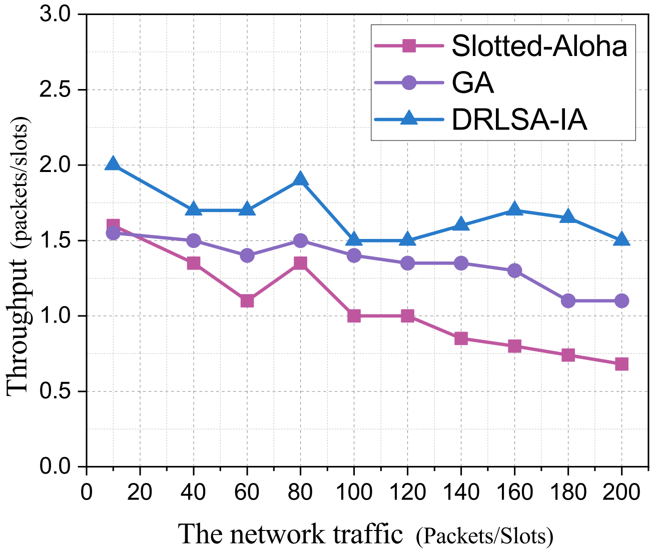

Figure 10. Throughput of protocols with different network traffic.

We next compare the PDR of DRLSA-IA with the other two algorithms with different network traffic. The number of nodes equals 10. As shown in Figure 11, the packet delivery ratio of DRLSA-IA outperforms that of the other two algorithms. Compared with Slotted-Aloha and GA, the PDR of DRLSA-IA is increased by 38.9% and 34.37%, respectively. With the increase in network traffic, the PDR of the three algorithms drops sharply. High traffic means there are more messages to be generated. That is, the possibility of message conflicts increases. Thus, the PDR of algorithms decreases. However, DRLSA-IA allows multiple nodes to transmit simultaneously without collisions, which greatly improves the transmission efficiency. The other two algorithms only allow one message transmission at most. Therefore, the PDR of DRLSA-IA is higher than the other two algorithms.

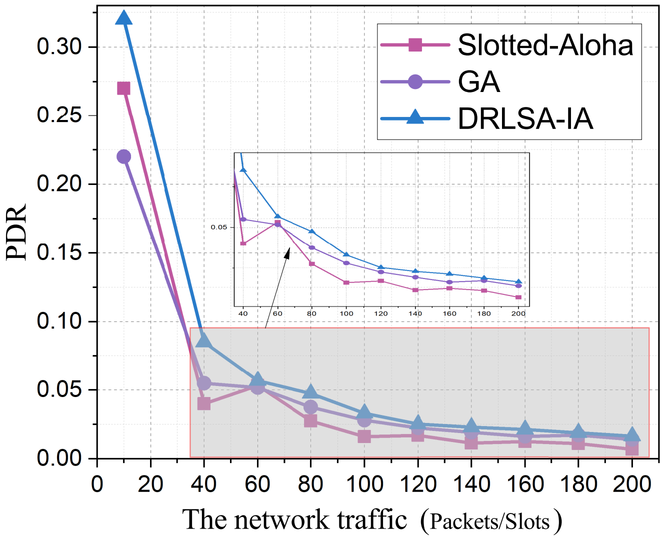

Figure 11. Packet delivery ratio of algorithms with different network traffic.

We also compare the fairness index of the three algorithms with different network traffic. The number of nodes equals 10. As shown in Figure 12, the *FI* of DRLSA-IA is marginally less than the *FI* of Slotted-Aloha, remaining approximately 0.9. The reason is that Slotted-Aloha starts transmission immediately when there is a message. However, the Slotted-Aloha throughput is considerably low. Compared with Slotted-ALOHA, the throughput of DRLSA-IA increases by 55.23%. Moreover, the *FI* of DRLSA-IA is greater than the *FI* of GA.

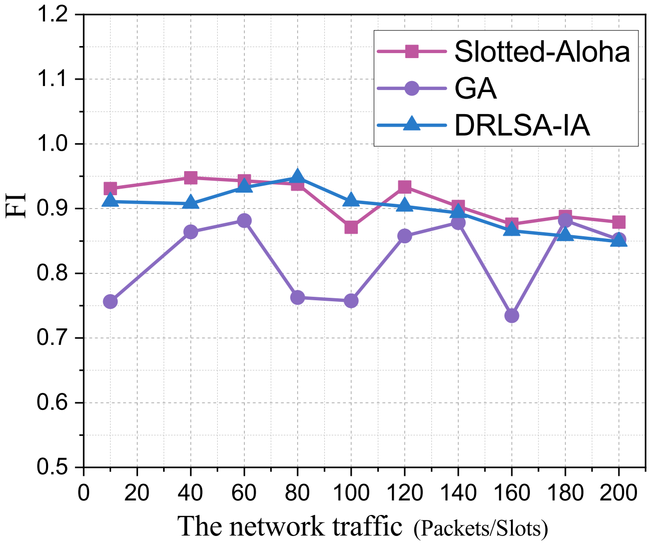

Figure 12. Fairness index of algorithms with different network traffic.

We then study the network throughput of algorithms with a different number of nodes; the offered network traffic is 80 packets/slots. As shown in Figure 13, the throughput of DRLSA-IA outperforms the other two algorithms. Compared with GA and Slotted-Aloha, the throughput of DRLSA-IA increases by 22.77% and 3.36 times, respectively. The main reason is that DRLSA-IA uses deep reinforcement learning algorithms to intelligently learn the idea of time-domain interference alignment. Moreover, as there is no need to enumerate the state space, DRLSA-IA can be applied to a network with a large number of nodes.

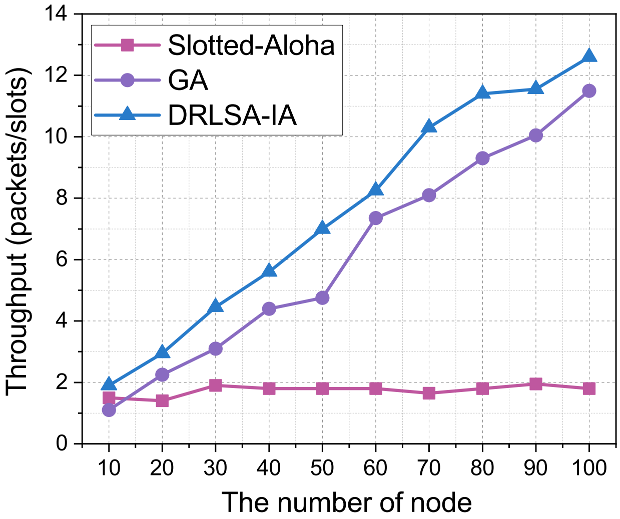

Figure 13. Throughput of algorithms with different number of nodes.

We also study the packet delivery ratio of algorithms with a different number of nodes; the offered network traffic is 80 packets/slots. As shown in Figure 14, the *PDR* of DRLSA-IA outperforms that of the other two algorithms. Compared with GA and Slotted-Aloha algorithm, the PDR of DRLSA-IA increases by 20.46% and 3.38 times, respectively. The start transmission time and propagation delay between sender and receiver determine whether there will be a collision at the destination node. Thus, reception time can be shown as a duality between the transmission time and the location of the nodes. In DRLSA-IA, we use reinforcement learning to learn the relationship between them. In addition, we design the TDIA-based reward mechanism to improve the PDR and reduce message collisions. DRLSA-IA can better adapt to different network scales and maintain a highly PDR in large-scale networks.

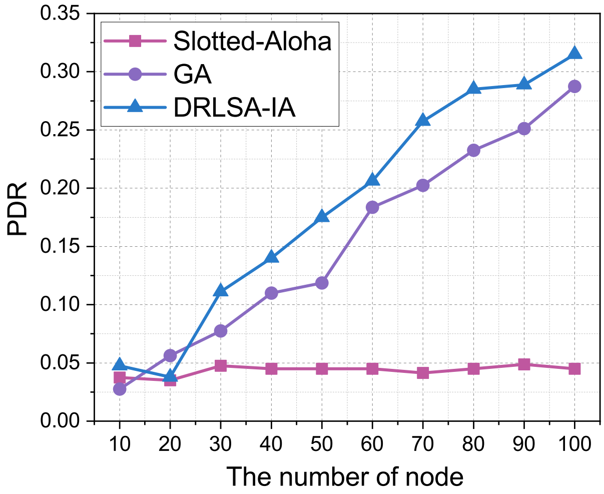

Figure 14. Packet delivery ratio of algorithms with different number of nodes.

We evaluate the fairness index of algorithms with a different number of nodes. The network traffic is 80 packets/slots. As shown in Figure 15, the fairness index of DRLSA-IA outperforms that of the other two algorithms. Compared with Slotted-Aloha and GA algorithm. the fairness index of DRLSA-IA increases by 14.23% and 75.75%, respectively. As the number of nodes increases, the fairness index of DRLSA-IA and GA algorithms decreases, while the Slotted-Aloha remains basically unchanged. The main reason is that as long as there is a message, Slotted-Aloha will immediately send it in the next slot. Changes in the number of nodes do not affect the fairness of Slotted-Aloha. The fairness of DRLSA-IA decreases slightly as the number of nodes increases, but that of DRLSA-IA is still fairer than the other two algorithms.

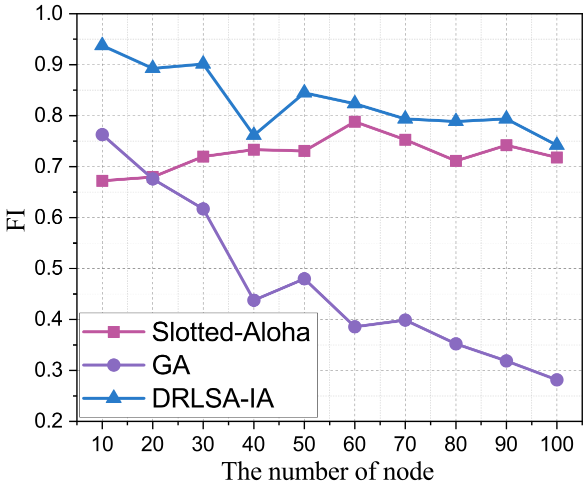

Figure 15. Fairness index of protocols with different number of nodes.

6. Conclusions

In this paper, we proposed a deep reinforcement learning-based time-domain interference alignment scheduling algorithm (called DRLSA-IA). In DRLSA-IA, nodes learned efficient transmission scheduling in continuous interaction with the environment. With the TDIA-based reward mechanism, nodes achieved parallel transmission of messages, and collisions effectively reduced. We also considered large propagation delays of UWANs and proposed a novel DRL algorithm in DRLSA-IA. Simulation results have shown that DRLSA-IA performs well in terms of throughput, packet delivery ratio and fairness index with the different network traffic and the different number of nodes. We believe that DRLSA-IA is a promising scheduling algorithm as it achieved better performance than other algorithms. In future work, we plan to consider the energy consumption of the algorithm based on deep reinforcement learning.

Author Contributions: Conceptualization, N.Z., N.Y. and Z.G.; methodology, N.Z. and N.Y.; validation, N.Z. and Z.G.; formal analysis, N.Z., N.Y. and Z.G.; writing—original draft preparation, N.Z.; writing—review and editing, N.Z., N.Y., Z.G. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly supported by China National Natural Science Foundation grant number No. 61972166, No. 61671169 and Industry University Cooperation Project of Fujian Province under Grant 2021H6030. Thanks to Research fund of Talents of QuanZhou City, No. 2018C109R.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saeed, N.; Celik, A.; Al-Naffouri, T.Y.; Alouini, M.S. Underwater optical wireless communications, networking, and localization: A survey. *Ad Hoc Netw.* **2019**, *94*, 101935. [CrossRef]
2. Songzuo, L.; Iqbal, B.; Khan, I.U.; Ahmed, N.; Qiao, G.; Zhou, F. Full Duplex Physical and MAC Layer-Based Underwater Wireless Communication Systems and Protocols: Opportunities, Challenges, and Future Directions. *J. Mar. Sci. Eng.* **2021**, *9*, 468. [CrossRef]
3. Jiang, S. On reliable data transfer in underwater acoustic networks: A survey from networking perspective. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1036–1055. [CrossRef]
4. Song, Y. Underwater acoustic sensor networks with cost efficiency for internet of underwater things. *IEEE Trans. Ind. Electron.* **2020**, *68*, 1707–1716. [CrossRef]
5. Jiang, S. State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 96–131. [CrossRef]
6. Alfouzan, F.A. Energy-efficient collision avoidance MAC protocols for underwater sensor networks: Survey and challenges. *J. Mar. Sci. Eng.* **2021**, *9*, 741. [CrossRef]
7. Bouabdallah, F.; Boutaba, R.; Mehaoua, A. Collision avoidance energy efficient multi-channel MAC protocol for underwater acoustic sensor networks. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2298–2314. [CrossRef]
8. Khater, E.; El-Fishawy, N.; Tolba, M.; Ibrahim, D.M.; Badawy, M. Buffering Slotted ALOHA protocol for underwater acoustic sensor networks based on the slot status. *Wirel. Netw.* **2021**, *27*, 3127–3145. [CrossRef]
9. Zhao, R.; Long, H.; Dobre, O.A.; Shen, X.; Ngatched, T.M.; Mei, H. Time reversal based MAC for multi-hop underwater acoustic networks. *IEEE Syst. J.* **2019**, *13*, 2531–2542. [CrossRef]
10. Sun, N.; Wang, X.; Han, G.; Peng, Y.; Jiang, J. Collision-free and low delay MAC protocol based on multi-level quorum system in underwater wireless sensor networks. *Comput. Commun.* **2021**, *173*, 56–69. [CrossRef]
11. Alablani, I.A.; Arafah, M.A. EE-UWSNs: A Joint Energy-Efficient MAC and Routing Protocol for Underwater Sensor Networks. *J. Mar. Sci. Eng.* **2022**, *10*, 488. [CrossRef]
12. Zhao, N.; Yao, N.; Gao, Z. A message transmission scheduling algorithm based on time-domain interference alignment in UWANs. *Peer-Netw. Appl.* **2021**, *14*, 1058–1070. [CrossRef]
13. Chitre, M.; Motani, M.; Shahabudeen, S. Throughput of networks with large propagation delays. *IEEE J. Ocean. Eng.* **2012**, *37*, 645–658. [CrossRef]
14. Liu, M.; Zhuo, X.; Wei, Y.; Wu, Y.; Qu, F. Packet-level slot scheduling MAC protocol in underwater acoustic sensor networks. *IEEE Internet Things J.* **2021**, *8*, 8990–9004. [CrossRef]

15. Su, Y.; Liu, X.; Han, G.; Fu, X. A Traffic Load-Aware OFDMA-Based MAC Protocol for Distributed Underwater Acoustic Sensor Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10501–10513. [CrossRef]
16. Dong, W.; Yang, Q.; Chen, Y.; Sun, S.; Huang, X. RHNE-MAC: Random Handshake MAC Protocol Based on Nash Equilibrium for Underwater Wireless Sensor Networks. *IEEE Sens. J.* **2021**, *21*, 21090–21098.
17. Chen, W.; Guan, Q.; Yu, H.; Ji, F.; Chen, F. Medium Access Control Under Space-Time Coupling in Underwater Acoustic Networks. *IEEE Internet Things J.* **2021**, *8*, 12398–12409. [CrossRef]
18. Geethu, K.S.; Babu, A.V. Energy optimal channel attempt rate and packet size for ALOHA based underwater acoustic sensor networks. *Telecommun. Syst.* **2017**, *65*, 429–442. [CrossRef]
19. Chirdchoo, N.; Soh, W.S.; Chua, K.C. Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks. In Proceedings of the IEEE INFOCOM 2007—26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6–12 May 2007; pp. 2271–2275.
20. Lee, J.-W.; Cho, H.-S. A Hybrid Sender- and Receiver-Initiated Protocol Scheme in Underwater Acoustic Sensor Networks. *Sensors* **2015**, *15*, 28052–28069. [CrossRef]
21. Sivakumar, V.; Rekha, D. Node scheduling problem in underwater acoustic sensor network using genetic algorithm. *Pers. Ubiquitous. Comput.* **2018**, *22*, 951–959. [CrossRef]
22. Lmai, S.; Chitre, M.; Laot, C.; Houcke, S. Throughput-Efficient Super-TDMA MAC Transmission Schedules in Ad Hoc Linear Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* **2017**, *42*, 156–174. [CrossRef]
23. Kaur, A.; Kumar, K. Energy-efficient resource allocation in cognitive radio networks under cooperative multi-agent model-free reinforcement learning schemes. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1337–1348. [CrossRef]
24. Naparstek, O.; Cohen, K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. *IEEE Trans. Wirel. Commun.* **2018**, *18*, 310–323. [CrossRef]
25. Jin, Z.; Zhao, Q.; Su, Y. RCAR: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks. *IEEE Sens. J.* **2019**, *19*, 10881–10891. [CrossRef]
26. Di Valerio, V.; Presti, F.L.; Petrioli, C.; Picari, L.; Spaccini, D.; Basagni, S. CARMA: Channel-aware reinforcement learning-based multi-path adaptive routing for underwater wireless sensor networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2634–2647. [CrossRef]
27. Park, S.H.; Mitchell, P.D.; Grace, D. Reinforcement learning based MAC protocol (UW-ALOHA-Q) for underwater acoustic sensor networks. *IEEE Access* **2019**, *7*, 165531–165542. [CrossRef]