electronics

MDPI

Article

An Improved CSMA/CA Protocol Anti-Jamming Method Based on Reinforcement Learning

Zidong Ming [1], Xin Liu [1,2,*], Xiaofei Yang [1,2] and Mei Wang [1,2]

[1] College of Information Science and Engineering, Guilin University of Technology, Guilin 541006, China; zdming@glut.edu.cn (Z.M.); 2006016@glut.edu.cn (X.Y.); mwang@glut.edu.cn (M.W.)
[2] Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin 541006, China
* Correspondence: liuxin2017125@glut.edu.cn

Abstract: The CSMA/CA algorithm uses the binary backoff mechanism to solve the multi-user channel access problem, but this mechanism is vulnerable to jamming attacks. Existing research uses channel-hopping to avoid jamming, but this method fails when the channel is limited or hard to hop. To address this problem, we first propose a Markov decision process (MDP) model with contention window (CW) as the state, throughput as the reward value, and backoff action as the control variable. Based on this, we design an intelligent CSMA/CA protocol based on distributed reinforcement learning. Specifically, each node adopts distributed learning decision-making, which needs to query and update information from a central status collection equipment (SCE). It improves its anti-jamming ability by learning from different environments and adapting to them. Simulation results show that the proposed algorithm is significantly better than CSMA/CA and SETL algorithms in both jamming and non-jamming environments. And it has little performance difference with the increase in the number of nodes, effectively improving the anti-jamming performance. When the communication node is 10, the normalized throughputs of the proposed algorithm in non-jamming, intermittent jamming, and random jamming are increased by 28.45%, 21.20%, and 17.07%, respectively, and the collision rates are decreased by 83.93%, 95.71%, and 81.58% respectively.

Keywords: CSMA/CA; anti-jamming; SETL; reinforcement learning

check for updates

Citation: Ming, Z.; Liu, X.; Yang, X.; Wang, M. An Improved CSMA/CA Protocol Anti-Jamming Method Based on Reinforcement Learning. *Electronics* **2023**, *12*, 3547. https://doi.org/10.3390/electronics12173547

Academic Editor: Juan-Carlos Cano

Received: 5 July 2023
Revised: 10 August 2023
Accepted: 19 August 2023
Published: 22 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Wireless LAN is a technology that uses wireless communication to connect data, which has brought effective impacts on the development of smart terminal, smart home, smart manufacturing, and smart automobile applications. The next generation of multi-user wireless channel access devices requires high-bandwidth and low-latency services [1]. Multi-user wireless channel access devices generally adopt the CSMA/CA protocol. The CSMA/CA protocol, as a common wireless channel access mechanism algorithm of 802.11, uses binary backoff mechanism to solve the channel access problem of multi-users. When the nodes compete with each other, the competition window is gradually increased in a binary way, thus reducing collisions between the nodes. When the load is heavy, several trials need to be conducted each time, which leads to large waiting delay and low channel utilization. Due to the openness of wireless channels, wireless communication is vulnerable to path weakness, environmental noise, and malicious jamming [2–5]. The binary backoff mechanism is easy to exploit by malicious jammers, and the smart jammer significantly degrades the network performance by imposing targeted jamming on the backoff mechanism [6,7]. Figures 1 and 2 represent the collision process caused by nodes competing for channels and the collision process caused by jamming, respectively. In Figure 1, we can see that the collision of sending nodes is mainly caused by channel competition between nodes. In Figure 2, due to the influence of jamming, the data cannot be correctly parsed by the receiving node. The sending node cannot receive the ACK data frame. This leads

to abnormal backoff, which results in a significant decline in performance. As a common anti-jamming technology in wireless communication, frequency-hopping technology [8] achieves anti-jamming by hopping frequently in the frequency dimension during the communication process. This method can effectively counter the influence of various jamming sources such as signal interference, noise interference, and intentional interference on wireless communication. In order to improve the anti-jamming performance of the CSMA/CA protocol, some scholars proposed an anti-jamming scheme based on jamming detection, and then via channel-hopping to achieve the purpose of anti-jamming [9–11]. Although these methods can counter jamming, they will fail when the channel is limited or hard to jump. At this time, anti-jamming performance can be improved from the CSMA/CA protocol, which is essentially time-hopping. The CSMA/CA protocol can achieve the effect of avoiding jamming through the backoff mechanism. Reinforcement learning [12] learns an optimal policy by constantly interacting with the environment. Reinforcement learning can learn and make decisions in complex, unknown, and dynamic environments. At present, reinforcement learning has been applied in wireless communication [13–15]. Reference [13] proposed a packet flow-based reinforcement-learning MAC protocol, which effectively improves the channel utilization. Reference [14] proposed a MAC protocol based on reinforcement learning, which reduces the probability of channel conflict through Q-learning. Based on reinforcement learning and deep reinforcement learning, reference [15] designed a multi-user and multi-task mobile edge computing algorithm, which reduces resource constraints and communication delays. Hence, we designed an intelligent CSMA/CA protocol backoff action based on reinforcement learning from the network protocol, and verified its performance in wireless networks through simulation. Our contributions are summarized below:

- We analyze the impact of malicious jamming on the CSMA/CA algorithm through simulations. In the jamming environment, the CSMA/CA protocol is difficult to distinguish between link collision and malicious jamming, which leads to a significant decrease in the performance of the CSMA/CA algorithm;
- We propose a Markov decision process (MDP) model for CSMA/CA. We consider the contention window (CW) as an environmental state, throughput as the reward value, and backoff action as the control variable;
- We propose an improved CSMA/CA protocol anti-jamming method based on distributed reinforcement learning. Each node adopts distributed learning decision-making, which needs to query and update information from a central SCE. By learning different environments, the optimal backoff action is learned, and adaptive changes are made for different environments to improve its anti-jamming performance;
- The proposed algorithm is compared with CSMA/CA and smart exponential threshold linear algorithm (SETL) in a non-jamming environment, intermittent jamming environment, and random jamming environment. The simulation results show that our proposed algorithm is significantly better than CSMA/CA and SETL algorithms in different environments; it significantly reduces the collision rate and effectively improves the network throughput performance.

The remaining parts of this paper are presented as follows. Section 2 introduces the work related to CSMA/CA optimization. Section 3 introduces the system model. Section 4 presents the algorithm principle. Section 5 gives the simulation results and analysis. Section 6 gives the conclusion.

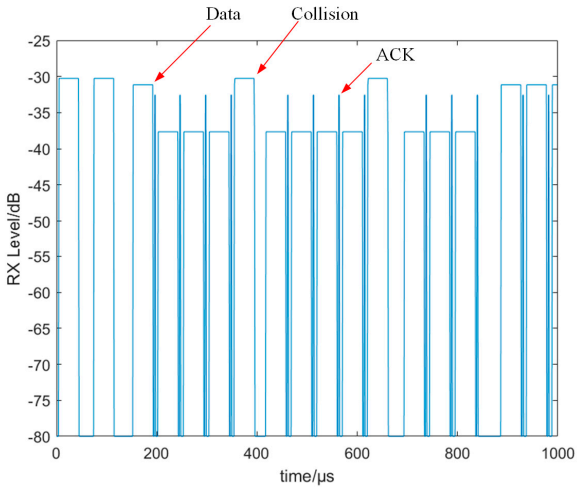

Figure 1. Node channel competition leads to the collision process.

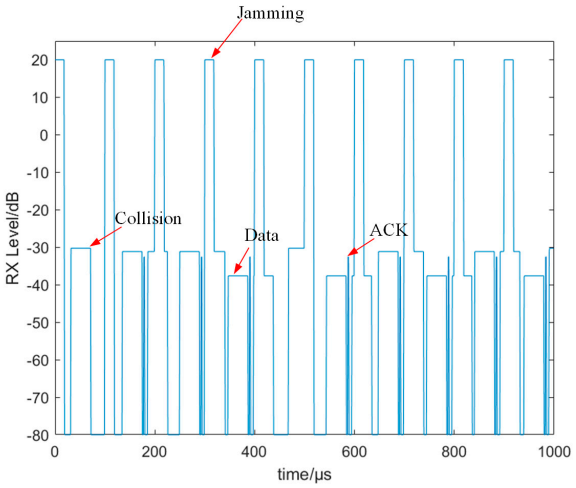

Figure 2. The jamming leads to the collision process.

2. Related Work

As a common channel access protocol, the performance of CSMA/CA protocol directly affects the channel utilization and reliability of wireless channels. Nowadays, a large number of CSMA/CA protocol optimization schemes have been proposed. In [16], a model based on Discrete-Time Markov Chain (DTMC) was proposed to analyze the network performance of IEEE 802.11 Distributed Coordination Function (DCF) and In-Band Full-Duplex (IBFD). This model had a small analysis error. Reference [17] proposed an adaptive CSMA/CA method. Firstly, the priority of each packet is set, and then the size of the contention window is dynamically adjusted according to the priority. However, this method needs to obtain network-related information first, which is not feasible in the actual operation process. In [18], a backoff algorithm based on collision rate for home WLAN (BA-HAN) was proposed. This method achieves high channel utilization while reducing the collision rate. Reference [19] proposed an improved non-timeslot CSMA/CA model, which minimized delay and took energy consumption and reliability as constraints to improve its algorithm performance. Reference [20] proposed an improved CSMA/CA backoff algorithm, which modifies the transmission rate of each node according to each node and network communication. The simulation results show that this method improves the channel utilization and ensures the fairness of each node. In [21], a new media access control method based on inductive reasoning (MACIR) was designed by adopting the minority game (MG) method. This method selects the optimal grouping strategy by learning the historical grouping situation, which does not need information interaction with the node.

Simulation results show that the simulation can improve network performance effectively. Reference [22] proposed a reservation-based MAC scheme. This method can ensure conflict-free data transmission and control delay minimization. Reference [23] proposed a modified binary exponential backoff (M-BEB) considering the fair access mechanism of the channel. According to different thresholds, the backoff rule is changed. Reference [24] combined the advantages of the linear increase linear decrease algorithm (LILD) [25] and the exponential increase exponential decrease algorithm (EIED) [26] to propose the smart exponential threshold linear algorithm (SETL), which considers the network load situation to select linear decision or binary decision. However, most of the research is only for ideal channels, without considering malicious jamming scenarios. In [27], a theoretical model of CSMA/CA under a jamming environment was established based on discrete Markov model, and the network performance under malicious jamming environment with different jamming rates and different data streams was analyzed by simulation experiments; the simulation results were verified by a semi-physical simulation platform. In [28], a jamming model based on random geometry was proposed. The influence of this model on the performance of CSMA/CA protocol was analyzed by simulation experiments. Research shows that malicious jamming leads to a serious decline in wireless communication performance, making wireless communication service quality worse. In [29], a hardware-in-the-loop simulation platform for a jamming environment was designed. The network performance of the CSMA/CA protocol was evaluated by data transmission rate (PDR), received signal strength (RSS), and energy consumption. The influence of the jamming environment on network performance was verified by simulation experiments. In [30], an improved two-dimensional Markov model was proposed. The coexistence of TDMA and CSMA/CA was analyzed—TDMA is regarded as a periodic jamming. The effectiveness of the model was verified by simulation, but the scene of malicious jamming was not considered. The CSMA/CA for jamming environment performance research shows that malicious jamming leads to a significant reduction in the performance of CSMA/CA. However, due to the openness of wireless links, malicious jamming is an important problem that cannot be ignored. In order to improve its anti-jamming performance, some scholars have proposed jamming detection algorithms for CSMA/CA. Reference [9], based on cognitive radio (CR) through perception, realized the dynamic change in channel access, so as to achieve the purpose of anti-jamming. Reference [10] proposed a jamming early stop detection method based on packet transmission time, which compares packet transmission time with the threshold value to determine whether to continue retransmission. Simulation results show that this method can effectively improve its anti-jamming performance. Reference [11] proposed a jamming detection algorithm based on channel utilization. When the channel utilization exceeds the defined threshold, the channel is switched to achieve the purpose of anti-jamming. Although these methods can improve the anti-jamming performance of CSMA/CA, most of these methods do not solve the problem of weak anti-jamming performance of CSMA/CA due to the network protocol. Therefore, this article uses the backoff action in combination with intensive study, based on the optimal resignation action under different environments of interest to study, so as to improve anti-jamming ability. We consider the anti-jamming problem of CSMA/CA with regard to the network protocol, which is a novel solution.

3. System Model

The node topology model is shown in Figure 3. The node decision and information interaction time series model is shown in Figure 4. In a single channel, there are multiple communication nodes communicating with each other, and there is a jammer as well as status collection equipment (SCE). Each node in the network has a receiver and a transmitter, which adopts the half-duplex communication process and uses the CSMA/CA protocol to avoid conflict. In order to facilitate the simulation, we have discretized the simulation time. Each node adopts distributed learning decision-making, which needs to query and update information from a central SCE.

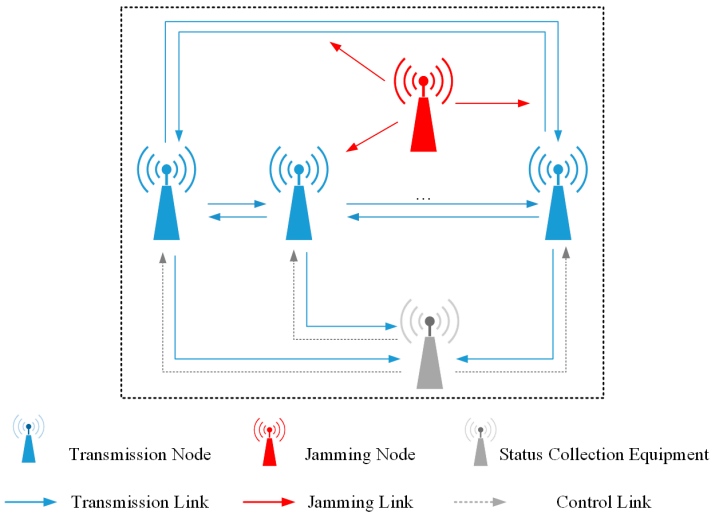

Figure 3. Node topology model.

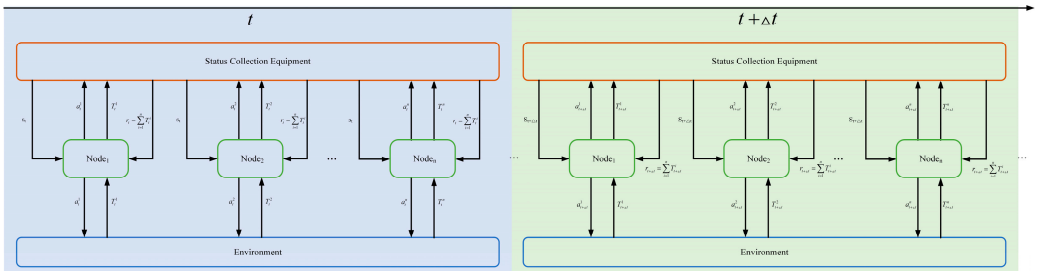

Figure 4. Node decision and information interaction time series model.

In Figure 4, each node obtains all CW from SCE as its own state s_t before decision-making. The decision a_t^i is made according to the current state s_t to obtain its own throughput T_t^i, which is then passed to SCE. SCE calculates the global throughput performance, and then sends the global throughput as a reward r_t to each node. Each node updates the Q value according to the decision and state changes. T_t^i represents the throughput performance of node i at time t. The decision time interval Δt is 200 slots.

Assuming that the transmission power of the communication node is P^i, its path gain is g^i and the environmental noise is $n(t)$. The transmission power of the interference node is P^J, and its path gain is g^J. Then, the instantaneous signal to interference plus noise ratio (SINR) at the moment is

$$SINR = 10 \cdot log \frac{\sum_{i=1}^{n} P_t^i \cdot g_t^i}{n(t) + P_t^J \cdot g_t^J} \tag{1}$$

$TranFlag$ indicates whether the data frame is successfully transmitted. When $SINR$ exceeds the threshold η, $TranFlag = 1$ indicates that the data frame is successfully transmitted. Otherwise, $TranFlag = 0$, indicating that the data frame transmission failed. Then $TranFlag_t$ at time t is

$$TranFlag_t = \begin{cases} 1, SINR_t \geq \eta \\ 0, SINR_t < \eta \end{cases} \tag{2}$$

The data frame is shown in Figure 5. The data frames, from front to back, are destination address frame (DST), source address frame (SRC), Type frame, DATA frame, and frame check sequence (FCS).



Figure 5. The data frame.

Jammers affect the communication process of legitimate nodes by sending jamming signals. We consider two different jammers: intermittent jamming and random jamming. It is assumed that the packet transmission is successful as long as 70% of the data frames are correctly parsed during the packet transmission process.

The SCE can interact with each node, which is mainly used to store the contention window information and throughput information of each node.

4. Principle of Algorithm

4.1. CSMA/CA Protocol Algorithm

As a common channel access mechanism in WLAN, CSMA/CA mainly solves the problem of channel competition between nodes through a binary backoff mechanism, so as to improve its channel utilization and reduce collision rate. When a communication node needs to send data, it first listens to the channel. If the channel is idle, the data packet is transmitted after the DIFS time slot. Otherwise, the backoff mechanism is carried out. After receiving the packet and waiting for the SIFS time slot, the destination node sends an ACK frame to the sender node. The specific process is shown in Figure 6. The backoff mechanism of CSMA/CA is that the node randomly generates the $BackOffTime$ when it senses that the channel is busy ($BackOffTime \in [0, CW_t - 1]$). On the first fallback attempt, the contention window is set to $CW_t = CW_{min}$. The contention window is doubled on each failed attempt, that is, $CW_t = 2 * CW_{t-1}$, until the maximum value is reached. When the contention window reaches the maximum number of backoffs, the packet is dropped. When a backoff attempt is successful, it is set to $CW_t = CW_{min}$. The CSMA/CA algorithm reduces the collision probability by increasing the waiting time of the sending node during the collision. Although it can solve the node collision problem, the large waiting time leads to low channel utilization and poor throughput performance. Furthermore, this mechanism is easy to be exploited by jamming, which leads to significant degradation of network performance.

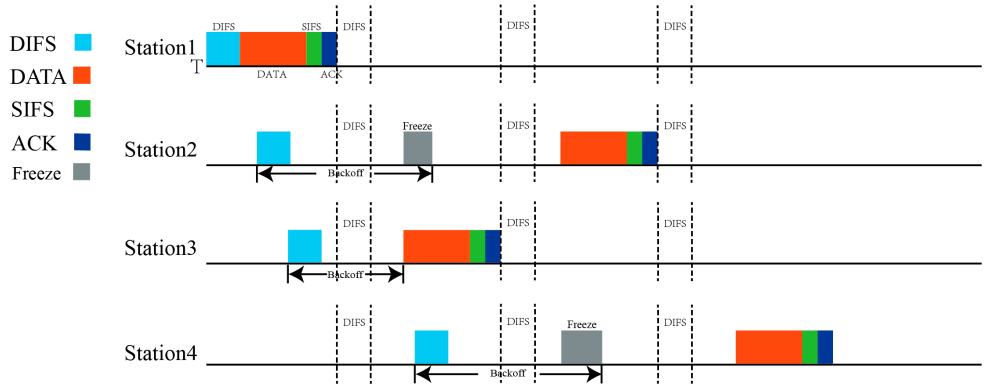

Figure 6. The backoff mechanism of CSMA/CA.

4.2. SETL Algorithm

The SETL [13] combines the advantages of the LILD and EIED algorithms, considers the network load, and sets the threshold $CW_{Threshold}$. When $CW \geq CW_{Threshold}$, it is considered that the current node is heavily loaded, and it is difficult for the communication node to compete for the communication channel. Otherwise, it is considered that the current network load is light and the communication node can easily obtain the communication channel. According to the network load, the SETL algorithm decides to choose a linear decision or a binary decision. This method can make the communication node choose a more suitable backoff window value, so that it can effectively improve its throughput and effectively reduce the collision rate. The backoff idea of the SETL algorithm is as follows: when a collision occurs, if $CW_t \geq CW_{Threshold}$ at time t, this indicates that the network load is serious; then, the CW is increased in a linear way, that is, $CW_{t+1} = \min[CW_t + CW_{min}, CW_{max}]$, so that it will not lead to a large waiting de-

lay due to a large CW by slowly increasing. Otherwise, this indicates that the network load is light, so the binary mode is adopted to increase the competition window, that is, $CW_{t+1} = 2 * CW_t$, so that a smaller CW can avoid a larger collision probability. When the communication nodes communicate successfully, if $CW_t \geq CW_{Threshold}$ at time t, this indicates that the network load is heavy at the moment; then, the CW is reduced in a linear way, that is, $CW_{t+1} = CW_t - CW_{min}$, so that the collision probability will not be larger by ensuring that it is slowly decreasing. Otherwise, this indicates that the network load is light, so the binary mode is adopted to reduce the competition window, that is, $CW_{t+1} = \max[CW_t/2, CW_{min}]$, so as to avoid the large waiting delay caused by a large competition window. Because of this advantage of SETL, we chose it as the comparison algorithm for the later simulations.

4.3. Reinforcement Learning

Reinforcement learning is a method for learning optimal policies through continuous interaction with the environment to maximize the cumulative reward value. Specifically, at time t, when the agent makes a decision a_t in the state s_t, the environment gives a reward value r_t. Then, it updates the state s_t to s_{t+1}, performs the new action a_{t+1}, and repeats the process. Figure 7 shows the reinforcement learning framework. The decision process of reinforcement learning can be generally modeled as a Markov decision process (MDP). The Markov Decision Process is represented by the tuples (S, A, P, R), where S represents environment state quantity set, A represents decision set, P represents transition state probability of decision, and R represents the reward value. In reinforcement learning, agents change their state by choosing actions to influence the environment. The environment updates the state of the agent according to the state transition probability and the action chosen by the agent, and then gives the agent the corresponding reward. The goal of the agent is to maximize the long-term cumulative reward. Each node adopts distributed learning decision-making, which needs to query and update information from a central SCE. We model the process of each node as a Markov decision process. The definitions are as follows:

(1) Define the state set S. $s_i \in S : S = \{s_1, s_2, \ldots, s_n\}$, $s_i = CW$ denotes the user state at the moment i in time. In our algorithm, we consider the CW as the environment state, and learn it through reinforcement learning to improve its network performance;
(2) Define the action set A. $a_i \in A : A = \{a_1, a_2, \ldots, a_n\}$, where a_i represents the adjustment action of CW at moment t, and the node backoff decision is optimized through different actions;
(3) Define the action transfer probability P. $P = p\{s_{i+1}|s_i, a_i\}$, which denotes the probability of picking an action a_i to transfer to s_{i+1} in state s_i;
(4) Define the reward value R. $R(s_i, a_i)$ denotes the reward value obtained by picking action a_i in state s_i. In the process of communication, the throughput performance of an environment can be tested by SCE, so it will be normalized, with throughput as a reward value.

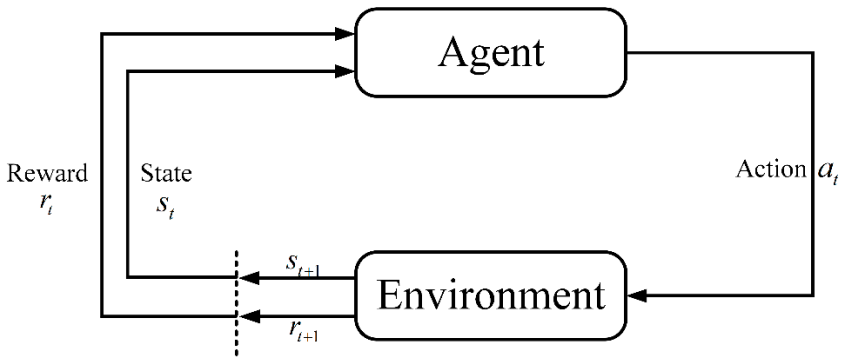

Figure 7. The reinforcement learning framework.4.4. Proposed Algorithm.

The CSMA/CA algorithm mainly deals with the collision problem between nodes through the binary backoff mechanism. However, this mechanism is easily exploited by jamming. Malicious jamming affects the normal retreat process by sending jamming signals. The CSMA/CA finds it difficult to distinguish between link collision and malicious jamming. Hence, we propose an improved CSMA/CA backoff decision based on distributed reinforcement learning. Specifically, each node adopts distributed learning decision-making, which needs to query and update information from a central SCE. The training process and learning process are summarized as follows:

(1) Before making a decision, each node obtains the CW value of all nodes from the SCE device as the status information. Then, query the Q value table to obtain the optimal decision based on ε-greedy. This value is fed back to the SCE;
(2) Wait for a fixed period to calculate the throughput of the time period itself. Its own throughput value is sent to SCE, and the instantaneous throughput and CW value of other nodes are queried from SCE;
(3) The throughput of the whole network is calculated as the reward value. Then, the Q value table is updated based on the decision and the state changes.

Repeat the above process until the algorithm converges. Through continuous iteration and optimization, the proposed algorithm can gradually learn an optimal strategy. As a result, better decision making and performance can be achieved in the environment.

In the optimization of the backoff decision, our optimization objective is to find the optimal backoff decision to maximize the throughput and, thus, improve the network performance. Therefore, the agent's optimization objective is defined as maximizing the cumulative reward, which is defined as follows:

$$\Theta = \max \mathrm{E}_{\pi}\left[\sum_{i=0}^{n} R_i\right], \pi \in \Omega \tag{3}$$

In this paper, Q-learning is used to approximate the model by iterating over the Q values so as to find the optimal decision π^*. The Q value updating formula is as follows:

$$Q(s_i, a_i) = (1-\theta) \cdot Q(s_i, a_i) + \theta \cdot \left(r_i + \gamma \cdot \max_{a_{i+1}} Q(s_{i+1}, a_{i+1})\right) \tag{4}$$

θ denotes the learning rate, which adjusts for the effect of state and reward values on the Q value, and γ denotes the discount factor, which is the effect of long-term payoff on decision selection. r_i denotes the payoff value obtained by selecting action a_i in state s_i. s_{i+1} denotes the transfer of the intelligent perceptron to the next state after action a_i is selected in state s_i. $\max_{a_{i+1}} Q(s_{i+1}, a_{i+1})$ denotes the maximum Q value selected in state s_{i+1}.

The CW of each node reflects its judgment on the channel congestion state. A larger CW means that the channel is considered congested. Therefore, the CW can reflect the channel state most of the time. However, in the case of some periodic jamming (jamming does not occupy all the communication time; the user may adjust the transmission strategy to obtain communication opportunities—otherwise, any decision is meaningless), the situation becomes slightly more complicated. In addition to the increase in CW caused by multi-user node collision, jamming also increases the contention window value. Thus, users become very conservative, and even communication is completely suppressed. However, reinforcement learning can make up for this shortcoming because nodes can constantly try different actions. Even when all other user contention windows have larger values, it is possible to select the corresponding action when the user contention window is relatively idle. If the reward value increases, it is inferred that the current state is not channel congestion but possible jamming. If the conclusion is opposite, this shows that it may be true congestion at this time. In this state, the tendency of the decision is reduced, avoiding the selection of a smaller contention window value. Therefore, to a certain extent, the proposed algorithm can indirectly distinguish the two states of jamming and collision, so as to achieve the effect of intelligent anti-jamming.

The details of the proposed algorithm are given in Algorithm 1.

Algorithm 1: An improved CSMA/CA protocol anti-jamming method based on reinforcement learning

Initialize each node Q table
Initialize discount factor γ
Initialize learning factor θ
Initialize epsilon greedy ε
Initialize epsilon greedy decrement
Initialize normalized throughput T
Initialize starte state s_1
For each, do:
The each node gets the state s_i from SCE, select action a_i based on ε-greedy, and a_i is fed back to SCE;
Execute action a_i, update each node $CW_{node} = CW$, get the next state s_{i+1} from SCE;
Wait for a fixed period, observe its normalized throughput T^i, and T^i is fed back to SCE;
The SCE calculates the current total normalized throughput T;
$r_i = T$;
Update each node Q table: $Q(s_i, a_i) = (1-\theta) \cdot Q(s_i, a_i) + \theta \cdot \left(r_i + \gamma \cdot \max_{a_{i+1}} Q(s_{i+1}, a_{i+1}) \right)$;
Update s_i to s_{i+1};
Decrease ε;
End

5. Simulation Results and Analysis

In order to verify the effectiveness of our proposed algorithm, we built a wireless communication simulation platform with a jamming environment based on a Python environment. Specifically, the basic environment of CMSA/CA protocol communication simulation was built. Then, we developed intermittent jamming and random jamming algorithms; next, the reinforcement learning algorithm was developed, based on which the improved CSMA/CA protocol anti-jamming simulation based on reinforcement learning was developed. Using the simulation platform developed by us, we compared the proposed algorithm with CSMA/CA and SETL algorithms in a non-jamming environment, intermittent jamming, and random jamming, and analyzed the normalized throughput, collision rate, and average reward value. The experimental parameters of wireless communication were set as follows: the transmission rate of wireless communication channel was 1 Mbits/s, the length of each time slot was 20 μs, the length of data packet was 400 μs, the length of DIFS was 50 μs, the length of SIFS was 10 μs, the length of ACK was 20 μs, the minimum contention window value was 8, and the maximum contention window value was 256. The detailed network simulation parameters are shown in Table 1. The parameters of the jammer experiment were set as follows: the intermittent jammer took 400 time slots as a cycle, jamming 20 time slots at the beginning of each cycle, while no jamming signal was launched at other times. The random jammer took 200 time slots as the period and used the transfer probability 0.1 to carry out random jamming. Agent reward discount factor $\gamma = 0.9$, learning rate $\theta = 0.5$, greed factor $\varepsilon = 1.0$, and attenuation index was 0.0001; detailed agent simulation parameters are shown in Table 2. Hardware used in the simulation experiment was CPU: Intel(R) Core(TM) i5-8600K, memory: 16 G, GPU: NVIDIA Ge Force GTX 2080Ti.

Table 1. Network simulation parameters.

Parameters	Value
Channel bit rate	1 Mbits/s
Slots	20 μs
SIFS	10 μs
DIFS	50 μs
Data	400 μs
ACK	20 μs
Decision time interval Δt	200 slots
CW_{min}	8
CW_{max}	256

Table 2. Agent simulation parameters.

Parameters	Value
Discount factor γ	0.9
Learning factor θ	0.5
Actions list A	(8, 16, 32, 48, 64, 96, 128, 192, 256)
Epsilon greedy ε	1.0
Epsilon greedy decrement	0.0001

5.1. Analysis of the Average Reward

The reward value can judge the pros and cons of an action, in which it can adjust the action. Reinforcement learning maximizes the cumulative reward value by interacting with the environment. In the proposed algorithm, the optimal action is selected by learning which action will bring higher reward value by trying different actions. Figures 8–10, respectively, show the variation curves of the average reward value of different communication nodes in a non-jamming, intermittent jamming, and random jamming environment as a function of the number of iterations. From the simulation curve, we can find that, as the number of iterations increases, the average reward value gradually increases and eventually converges. This shows that our proposed algorithm gradually improves its throughput performance and improves the average reward value by learning the optimal decisions in different environments. Whether it is a non-jamming environment or a jamming environment, the average reward value of each node decreases with the increase in nodes, and the difference is not large, which is basically consistent with the trend in our throughput simulation performance. This is because we define normalized throughput as reward value. In the intermittent jamming and random jamming scenarios, the average reward value of node 2 is lower than that of other nodes in the previous iteration process. This is because, in the light load network, our proposed algorithm performs random exploration in this process, which leads to poor throughput performance due to large backoff decisions. However, with the increase in the number of iterations, the algorithm gradually learns the optimal decision, so that the performance of the normalized throughput is gradually improved. In the intermittent jamming scenario, when the number of iterations is about 4000, the average reward value of node 2 is better than that of other nodes. In the random jamming scenario, when the number of iterations is about 7900, the average reward value of node 2 is better than that of other nodes. From the simulation curve, it can be seen that our proposed algorithm is effective in both the non-jamming environment and jamming environment, demonstrating its effectively improved communication performance.

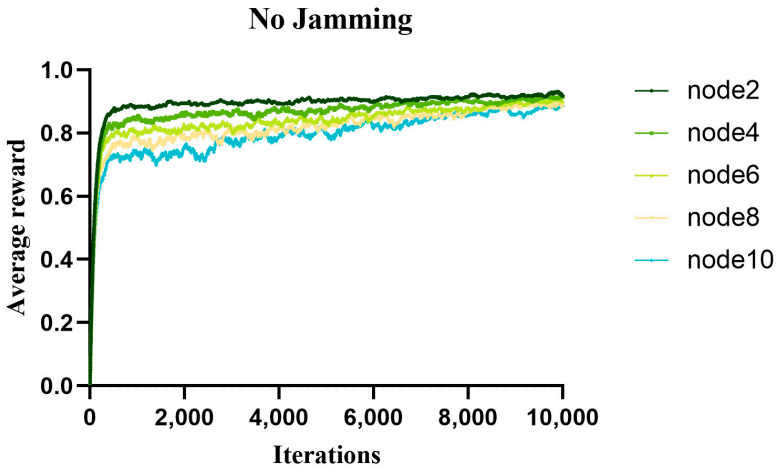

Figure 8. The average reward value of different nodes with the number of iterations in a non-jamming environment.

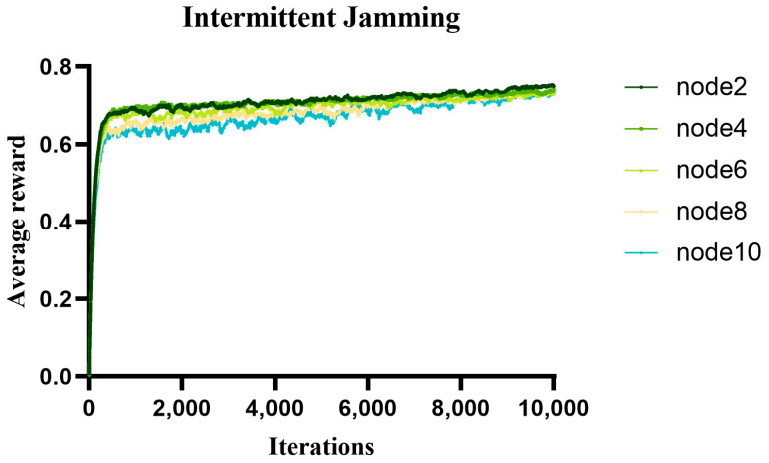

Figure 9. The average reward value of different nodes with the number of iterations in an intermittent jamming environment.

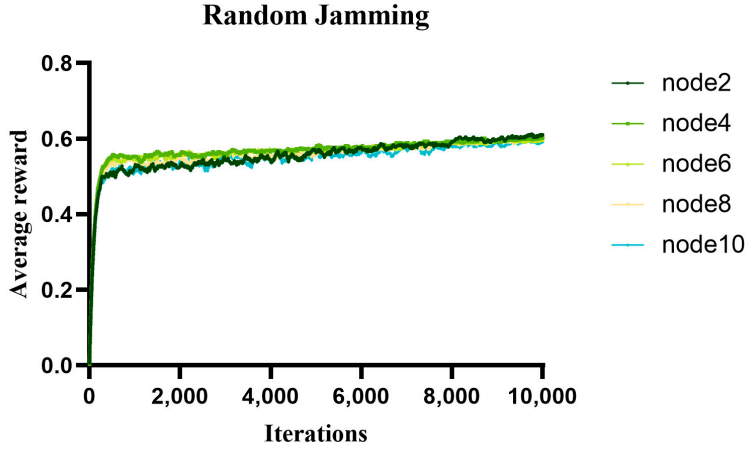

Figure 10. The average reward value of different nodes with the number of iterations in a random jamming environment.

5.2. Analysis of Throughput

The throughput plays an important role in network performance analysis. The throughput directly affects the efficiency of data transmission and the response speed of network services, which can help evaluate the network performance. Figures 11–13

represent the normalized throughput curves in non-jamming, intermittent jamming, and random jamming environments, respectively, as a function of the number of communication nodes. In the simulation curve, we can find that the normalized throughput performance of the network decreases with the increase in nodes. This is because the increase in the number of nodes under the same conditions increases the collision rate of nodes in the channel competition process. In order to reduce the collision of nodes, a fallback window is generated and the waiting time of nodes becomes longer, which leads to the decline of normalized throughput. In terms of different environments, the normalized throughput performance, from high to low, is the proposed method, the SETL, and then the CSMA algorithm. This is because SETL dynamically adjusts its contention window value according to the network load, thus improving its normalized throughput. However, our proposed algorithm aims to learn the optimal competitive window decision that can obtain the maximum normalized throughput for different environments by reinforcement learning for different environments. In terms of non-jamming, intermittent jamming, and random jamming environments, the normalized throughput, from high to low, is non-jamming environment, then intermittent jamming environment, and, lastly, random jamming environment. This is because, in a non-jamming environment, only the collision of nodes affects the channel competition of nodes. The influence of malicious jamming in the jamming environment makes it difficult for nodes to distinguish malicious jamming or node collision, so that the normalized throughput performance is significantly reduced in the jamming environment. The intermittent jammer has a cycle of 400 time slots, and it disturbs 20 time slots at the beginning of each cycle, while it does not initiate jamming signals at other times. The random jammer has a period of 200 time slots and uses a transition probability of 0.1 for random jamming. Therefore, random jamming affects the normalized performance more than intermittent jamming. It is not difficult to find from the simulation curve that our proposed algorithm shows little difference in normalized throughput performance as the number of nodes increases. Moreover, our proposed algorithm outperforms CSMA/CA and SETL in terms of normalized throughput performance. When the number of communication nodes is 10, the normalized throughput performance in different environments is improved by 28.45% and 16.67%, respectively, compared with the CSMA/CA algorithm and the SETL algorithm in the non-jamming environment. In the intermittent jamming environment, it increased by 21.20% and 4.81%, respectively; in the random jamming scenario, it increased by 17.07% and 3.12%, respectively.

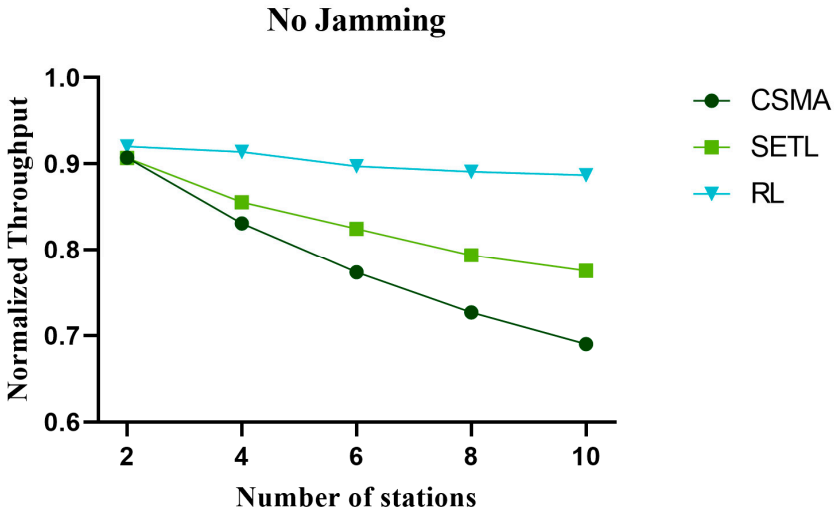

Figure 11. The change in normalized throughput with the number of nodes in the non-jamming environment.

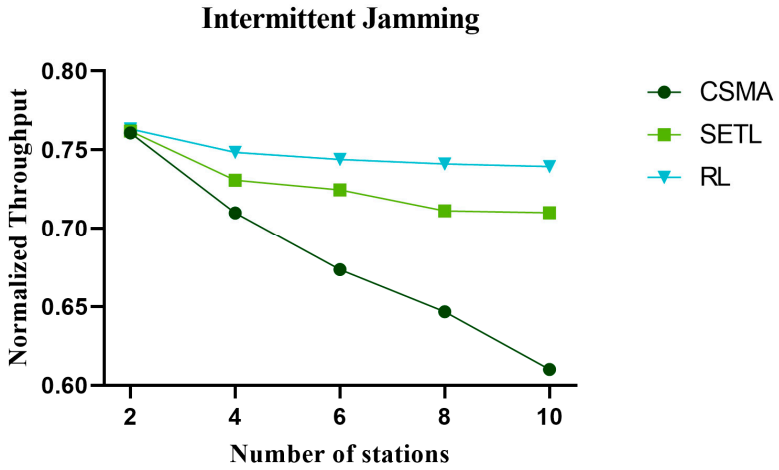

Figure 12. The change in normalized throughput with the number of nodes in the intermittent jamming environment.

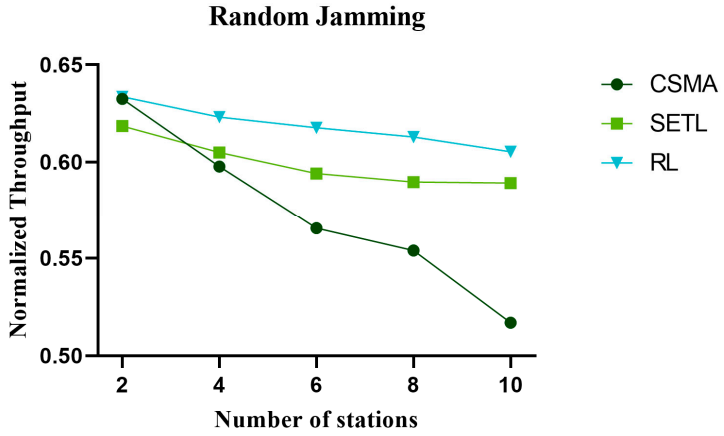

Figure 13. The change in normalized throughput with the number of nodes in the random jamming environment.

5.3. Analysis of the Collision Probability

The collision rate directly affects the network throughput performance. The higher collision rate leads to packet loss or retransmission, reducing the channel transmission rate, thus affecting the network throughput performance. Figures 14–16 represent the collision rate curves in non-jamming, intermittent jamming, and random jamming environments, respectively, as a function of the number of communication nodes. In the simulation curve, we can find that the collision probability of the network increases with the increase in nodes. This is due to the increase in the number of nodes, which increases the probability of mutual competition between nodes and, thus, increases the collision rate. From Figures 14–16, it can be found that the collision rate performance of our proposed algorithm is increasingly better than that of the CSMA/CA algorithm and SETL algorithm as the number of nodes increases. This is because our proposed algorithm reduces the collision rate by learning optimal decisions for different environmental states. Although the SETL algorithm can, according to the situation of the network, carry on the back window of the dynamic transformation, competition in the process of this window is not necessarily optimal. When the number of communication nodes is 10, the collision probability decreases by 83.93% and 31.24% compared with the CSMA/CA algorithm and SETL algorithm in the non-jamming environment, respectively. In the intermittent jamming environment, it is reduced by 95.71% and 27.42%, respectively; in the random jamming scenario, they were reduced by 81.58% and 15.09%, respectively. And with the increase in the number of nodes, the collision rate of our proposed algorithm has little change in its trend.

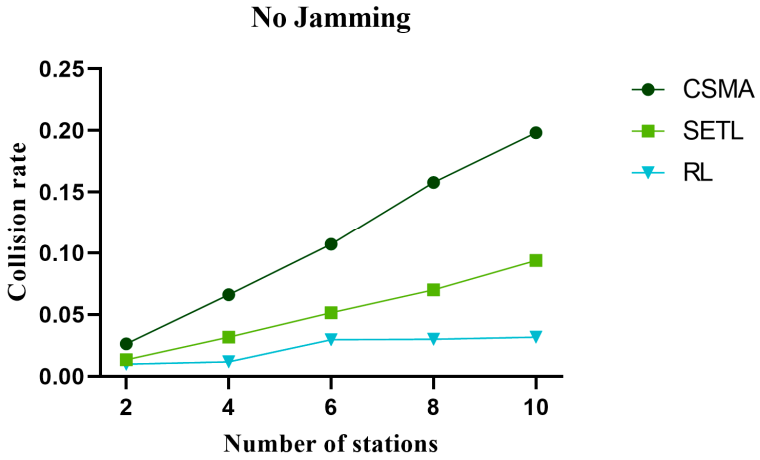

Figure 14. The change in collision rate with the number of nodes in the non-jamming environment.

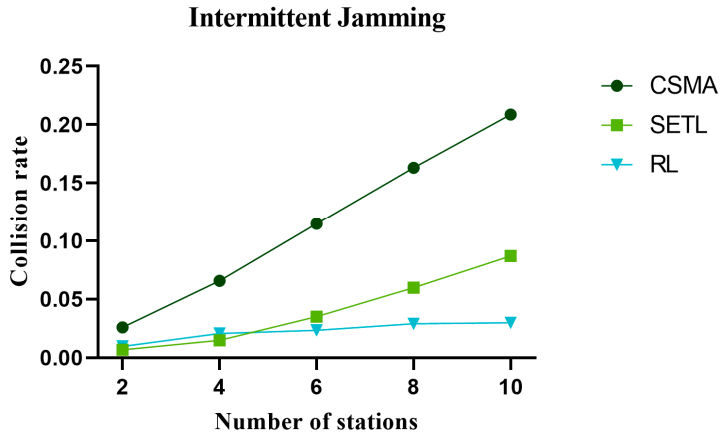

Figure 15. The change in collision rate with the number of nodes in the intermittent jamming environment.

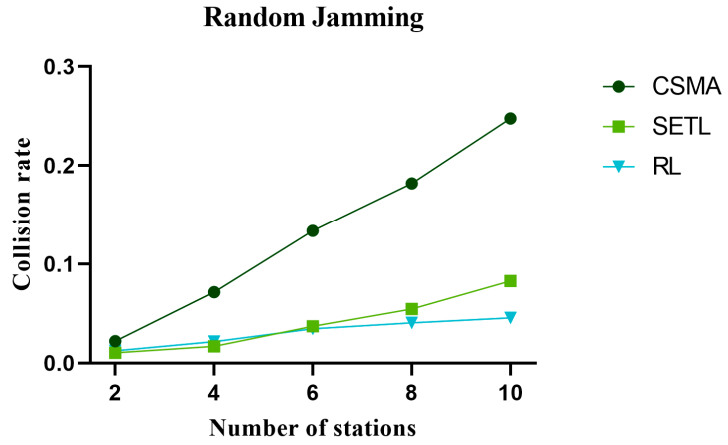

Figure 16. The change in collision rate with the number of nodes in the random jamming environment.

5.4. The Influence of Different Data Lengths on the Performance of the Algorithm

Figures 17–19 show the variation in normalized throughput of different algorithms with data length in non-jamming, intermittent jamming, and random jamming scenarios when the number of nodes is 10. Figures 20–22 show the changes in collision rate of different algorithms with data length in non-jamming, intermittent jamming, and random jamming scenarios when the number of nodes is 10. As shown in Figures 17–22, the proposed algorithm can achieve better throughput performance and lower collision rate

than the CSMA/CA and STEL algorithms. In Figure 17, in the non-jamming scenario, the normalized throughput of the SETL algorithm decreases with the increase in data length, while the normalized throughput of CSMA/CA and our proposed algorithm increases with the increase in data length. In Figure 20, the trend of collision rate changes is just the opposite, which is due to the fact that higher collision rate leads to poorer throughput performance. The SETL algorithm considers the network load, so it is more suitable for short data frames. Although the normalized throughput of our algorithm decreases with the decrease in data length, it is significantly better than the CSMA/CA and SETL algorithms in terms of network performance. In Figure 18, the normalized throughput of the CSMA/CA and SETL algorithms show a decreasing trend with the increase in data length in the intermittent jamming scenario, while the proposed algorithm shows an increasing trend. This is because the intermittent jamming adopts a periodic strategy, and, for CSMA/CA and SETL, short data frames lead to a larger collision rate, so that a larger backoff window is obtained, which improves the anti-jamming performance to a certain extent. The proposed algorithm aims to maximize throughput performance, so the performance for short frames decreases slightly. In Figures 19 and 22, CSMA/CA, SETL, and our proposed algorithm have the same change trend in the random jamming environment. In the random jamming scenario, these algorithms are more suitable for short data frames. The proposed algorithm selects the more suitable competition window after learning the environment, so as to achieve better communication performance. Figures 23–25 represent the normalized throughput curves at different data lengths in the proposed algorithm with the number of communication nodes in non-jamming, intermittent jamming, and random jamming environments. In Figures 23 and 24, the normalized throughput performance of our proposed algorithm is Data length = 400, Data length = 300, Data length = 200 (from high to low). In Figure 25, the trend is reversed. Figures 26–28 represent the collision rate curves of different data lengths of the proposed algorithm with the number of communication nodes in the environments of non-jamming, intermittent jamming, and random jamming. Figures 26–28 and Figures 23–25 have the opposite trend because higher throughput leads to smaller collision rate performance. In conclusion, the CSMA/CA algorithm is suitable for long data frames without jamming, while the jamming scenario is more suitable for short data frames. The SETL algorithm is more suitable for short data frames in both non-jamming and jamming scenarios. The proposed algorithm is suitable for long data frames in non-jamming and intermittent jamming scenarios, while it is suitable for short data frames in random jamming scenarios. Moreover, the proposed algorithm has better communication performance than the CSMA/CA and SETL algorithms in different environments and different data frame lengths.

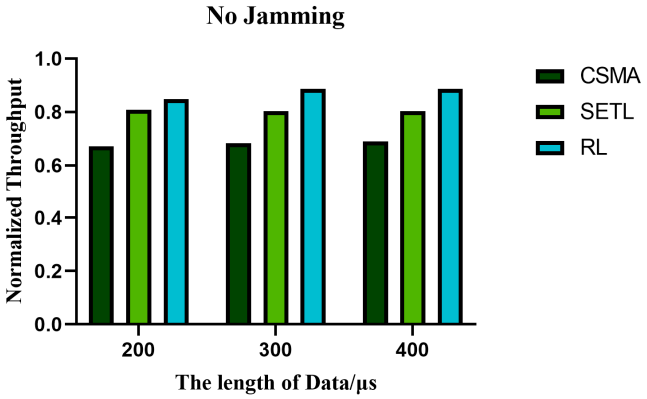

Figure 17. When the number of nodes is 10 in the non-jamming environment, the normalized throughput of different algorithms varies with the data length.

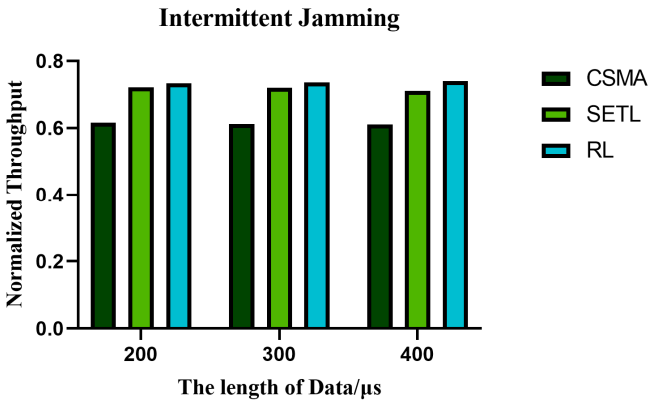

Figure 18. When the number of nodes is 10 in the intermittent jamming environment, the normalized throughput of different algorithms varies with the data length.

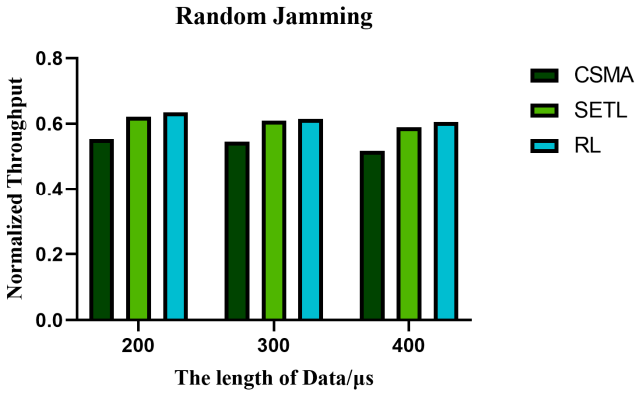

Figure 19. When the number of nodes is 10 in the random jamming environment, the normalized throughput of different algorithms varies with the data length.

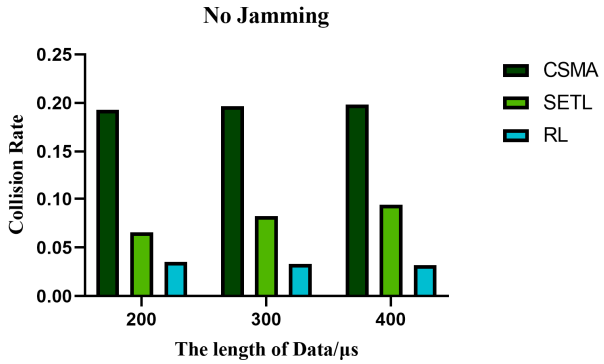

Figure 20. When the number of nodes is 10 in the non-jamming environment, the collision rate of different algorithms varies with the data length.

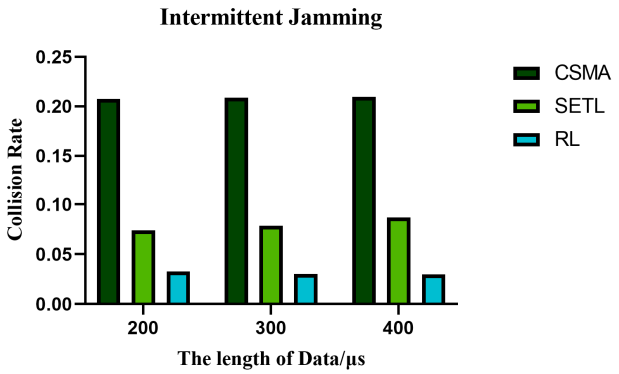

Figure 21. When the number of nodes is 10 in the intermittent jamming environment, the collision rate of different algorithms varies with the data length.

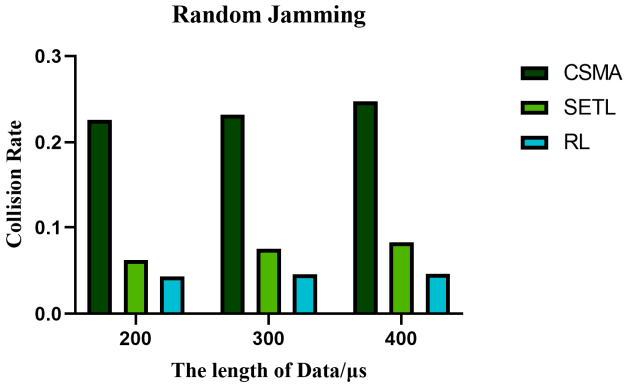

Figure 22. When the number of nodes is 10 in the random jamming environment, the collision rate of different algorithms varies with the data length.

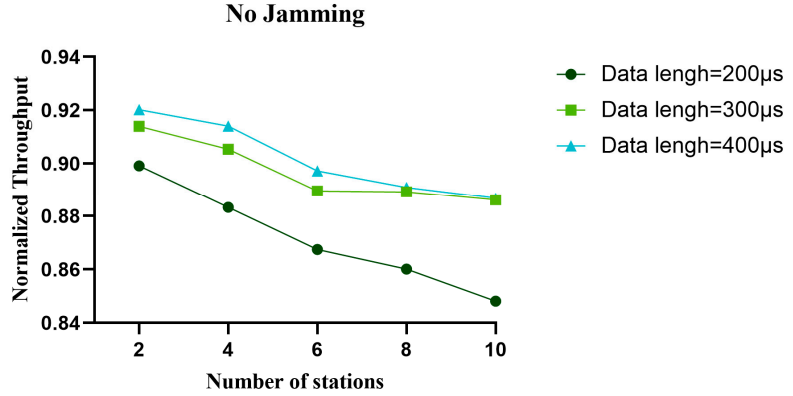

Figure 23. In the non-jamming environment, the normalized throughput of the proposed algorithm with different data lengths varies with the number of nodes.

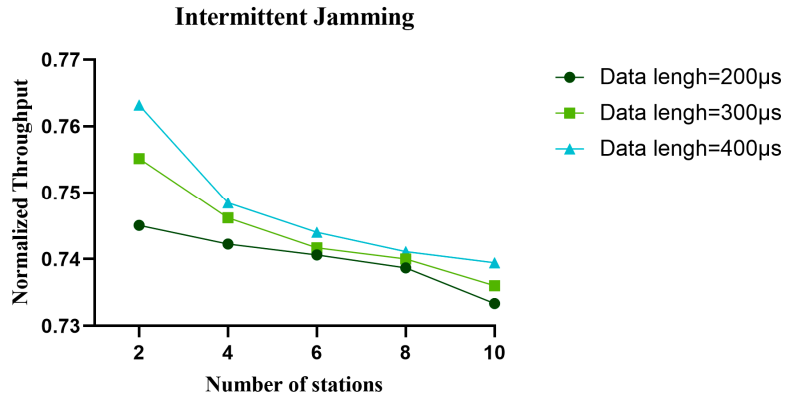

Figure 24. In the intermittent jamming environment, the normalized throughput of the proposed algorithm with different data lengths varies with the number of nodes.

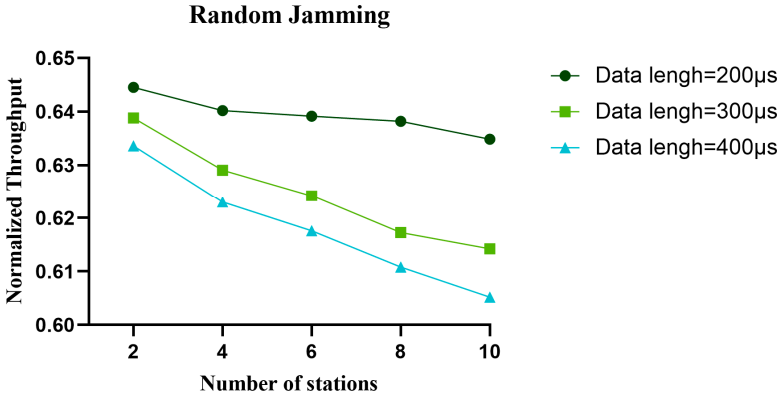

Figure 25. In the random jamming environment, the normalized throughput of the proposed algorithm with different data lengths varies with the number of nodes.

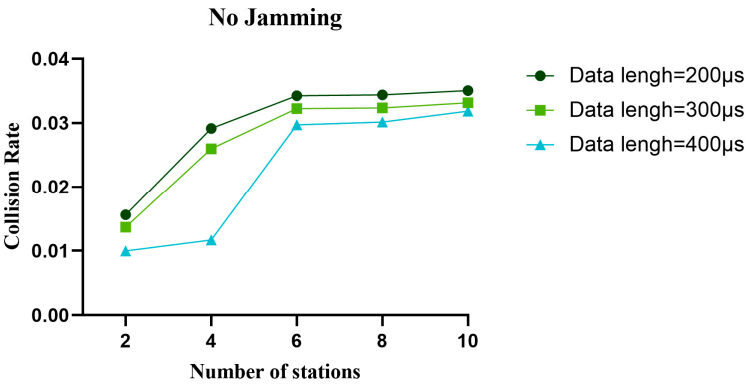

Figure 26. In the non-jamming environment, the collision rate of the proposed algorithm with different data lengths varies with the number of nodes.

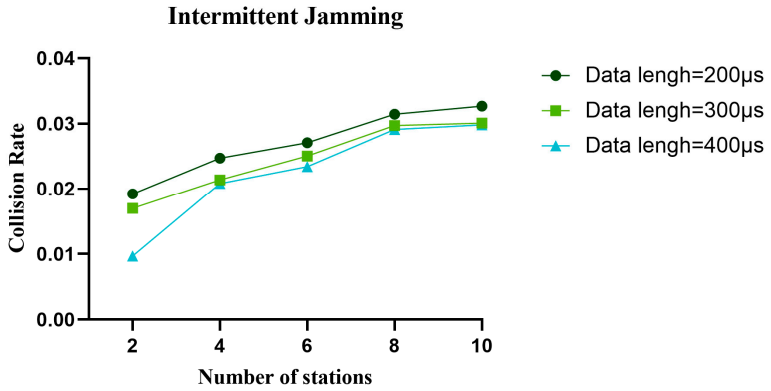

Figure 27. In the intermittent jamming environment, the collision rate of the proposed algorithm with different data lengths varies with the number of nodes.

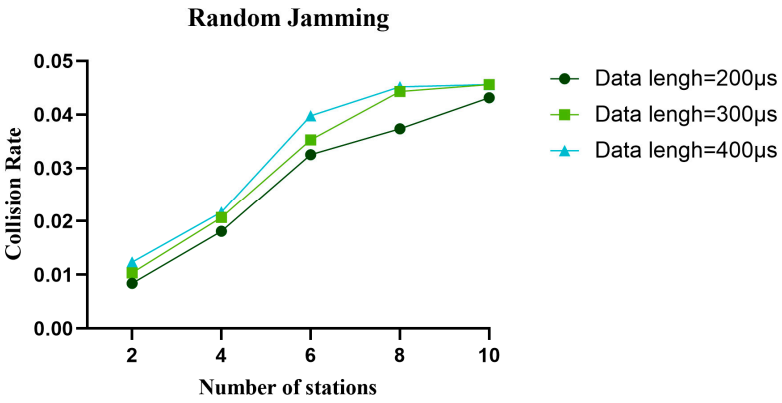

Figure 28. In the random jamming environment, the collision rate of the proposed algorithm with different data lengths varies with the number of nodes.

6. Conclusions

Through the simulation of a CSMA/CA jamming environment, we conclude that the CSMA/CA protocol has difficulty in distinguishing between node collision and malicious jamming, resulting in a significant decline in anti-jamming performance. In order to solve this problem, we proposed an improved CSMA/CA anti-jamming method based on distributed reinforcement learning, which can distinguish between jamming and collision. Specifically, by learning from different environments, the optimal backoff decision for different environments can be learned, so that it can be adaptively changed according to the environment, thereby improving its anti-jamming performance. Hence, we propose an MDP model with CW as the state, throughput as the reward value, and backoff action as the control variable. The simulation results show that the proposed algorithm is significantly better than CSMA/CA and SETL algorithms. And it has little performance difference with the increase in the number of nodes, effectively improving the anti-jamming performance. When the communication node is 10, the normalized throughput of the proposed algorithm in non-jamming, intermittent jamming, and random jamming environments is increased by 28.45%, 21.20%, and 17.07%, respectively, and the collision rate is decreased by 83.93%, 95.71%, and 81.58%, respectively. This method effectively improves the anti-jamming performance. Simulation experiments with different data lengths show that the proposed algorithm is suitable for long data frames in non-jamming and intermittent jamming scenarios, while it is suitable for short data frames in random jamming scenarios. However, the proposed algorithm also has some disadvantages. The algorithm interacts with nodes in the learning process, which brings great complexity in practical application scenarios. Moreover, the algorithm can only be effective for experienced or similar jamming. When there is a new form of jamming, the proposed algorithm needs a long time to converge. In

the next step, we will focus on how to detangle from the control channel, how to distinguish jamming collisions from the state, and how to quickly adapt unknown jamming.

Author Contributions: Conceptualization, Z.M. and X.L.; software, Z.M. and X.L.; experiment analysis, Z.M. and X.Y.; writing—review and editing, Z.M., X.L. and M.W.; funding acquisition, X.L. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported by the National Natural Science Foundation of China under Grant 61961010, 62071135, the Key Laboratory Found of Cognitive Radio and Information Processing, Ministry of Education (Guilin University of Electronic Technology) under Grant No. CRKL200204, No. CRKL220204, RZ18103102, and the 'Ba Gui Scholars' program of the provincial government of Guangxi.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the grants for this manuscript are still in the research phase, and some research data or key codes are currently limited to disclosure within the project team. However, the datasets used and/or analyzed during the current study are available via email (zdming@glut.edu.cn) on reasonable request.

Acknowledgments: We are very grateful to volunteers from GLUT for their assistance in the experimental part of this manuscript.

Conflicts of Interest: The authors declare that there are no conflict of interest due to personal relationships.

References

1. Rong, B. *6G: The Next Horizon: From Connected People and Things to Connected Intelligence*; Cambridge Univ. Press: Cambridge, UK, 2021.
2. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
3. Mpitziopoulos, A.; Gavalas, D.; Konstantopoulos, C.; Pantziou, G. A survey on jamming attacks and countermeasures in WSNs. *IEEE Commun. Surv. Tutorials* **2009**, *11*, 42–56. [CrossRef]
4. Zou, Y.; Zhu, J.; Wang, X.; Hanzo, L. A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. *Proc. IEEE* **2016**, *104*, 1727–1765. [CrossRef]
5. Tianyang, P.; Yonggui, L.; Yingtao, N.; Chen, H.; Zhi, X. Classification and Development of Communication Electronic Interference. *Commun. Technol.* **2018**, *51*, 2271–2278.
6. Parras, J.; Zazo, S. Wireless Networks under a Backoff Attack: A Game Theoretical Perspective. *Sensors* **2018**, *18*, 404. [CrossRef] [PubMed]
7. Medal, H.R. The wireless network jamming problem subject to protocol interference. *Networks* **2016**, *67*, 111–125. [CrossRef]
8. Liu, X.; Xu, Y.; Jia, L.; Wu, Q.; Anpalagan, A. Anti-Jamming Communications Using Spectrum Waterfall: A Deep Reinforcement Learning Approac. *IEEE Commun. Lett.* **2018**, *22*, 998–1001. [CrossRef]
9. Salameh, H.B.; Al-Quraan, M. Securing Delay-Sensitive CR-IoT Networking under Jamming Attacks: Parallel Transmission and Batching Perspective. *IEEE Internet Things J.* **2020**, *7*, 7529–7538. [CrossRef]
10. Halloush, R.D. Transmission Early-Stopping Scheme for Anti-Jamming over Delay-Sensitive IoT Applications. *IEEE Internet Things J.* **2019**, *6*, 7891–7906. [CrossRef]
11. Ng, J.; Cai, Z.; Yu, M. A New Model-Based Method to Detect Radio Jamming Attack to Wireless Networks. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 1–6. [CrossRef]
12. Martín-Guerrero, J.D.; Lamata, L. Reinforcement Learning and Physics. *Appl. Sci.* **2021**, *11*, 8589. [CrossRef]
13. Alhassan, I.B.; Mitchell, P.D. Packet Flow Based Reinforcement Learning MAC Protocol for Underwater Acoustic Sensor Networks. *Sensors* **2021**, *21*, 2284. [CrossRef]
14. Lee, T.; Jo, O.; Shin, K. CoRL: Collaborative Reinforcement Learning-Based MAC Protocol for IoT Networks. *Electronics* **2020**, *9*, 143. [CrossRef]
15. Elgendy, I.A.; Zhang, W.-Z.; He, H.; Gupta, B.B.; El-Latif, A.A.A. Joint computation offloading and task caching for multi-user and multi-task MEC systems: Reinforcement learning-based algorithms. *Wirel. Netw.* **2021**, *27*, 2023–2038. [CrossRef]
16. Murad, M.; Eltawil, A.M. Performance Analysis and Enhancements for In-Band Full-Duplex Wireless Local Area Networks. *IEEE Access* **2020**, *8*, 111636–111652. [CrossRef]
17. Lin, C.-L.; Chang, W.-T.; Lu, M.-H. MAC Throughput Improvement Using Adaptive Contention Window. *J. Comput. Chem.* **2015**, *3*, 1–14. [CrossRef]

18. Lin, C.-H.; Tsai, M.-F.; Hwang, W.-S.; Cheng, M.-H. A Collision Rate-Based Backoff Algorithm for Wireless Home Area Networks. In Proceedings of the 2020 2nd International Conference on Computer Communication and the Internet (ICCCI), Nagoya, Japan, 26–29 June 2020; pp. 51–56. [CrossRef]
19. Gamal, M.; Sadek, N.; Rizk, M.R.; Ahmed, M.A.E. Optimization and modeling of modified unslotted CSMA/CA for wireless sensor networks. *Alex. Eng. J.* **2020**, *59*, 681–691. [CrossRef]
20. Qiao, J.; Shen, X.S.; Mark, J.W.; Cao, B.; Shi, Z.; Zhang, K. CSMA/CA-based medium access control for indoor millimeter wave networks. *Wirel. Commun. Mob. Comput.* **2014**, *16*, 3–17. [CrossRef]
21. Park, J.; Yoon, C. Distributed Medium Access Control Method through Inductive Reasoning. *Int. J. Fuzzy Log. Intell. Syst.* **2021**, *21*, 145–151. [CrossRef]
22. Chao, I.-F.; Lai, C.-W.; Chung, Y.-H. A reservation-based distributed MAC scheme for infrastructure wireless networks. In Proceedings of the 2018 3rd International Conference on Intelligent Green Building and Smart Grid (IGBSG), Yilan, Taiwan, 22–25 April 2018; pp. 1–4. [CrossRef]
23. Zerguine, N.; Aliouat, Z.; Mostefai, M.; Harous, S. M-BEB: Enhanced and Fair Binary Exponential Backoff. In Proceedings of the 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 17–18 November 2020; pp. 142–147. [CrossRef]
24. Ke, C.-H.; Wei, C.-C.; Lin, K.W.; Ding, J.-W. A smart exponential threshold-linear backoff mechanism for IEEE 802. *11 WLANs. Int. J. Commun. Syst.* **2011**, *24*, 1033–1048. [CrossRef]
25. Bharghavan, V.; Demers, A.; Shenker, S.; Zhang, L. MACAW: A Media Access Protocol for Wireless LAN's. In Proceedings of the ACK SIGCOMM'94, London, UK, 31 August–2 September 1994.
26. Song, N.-O.; Kwak, B.-J.; Song, J.; Miller, L. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. In Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, Jeju, Republic of Korea, 22–25 April 2003; IEEE: Manhattan, NY, USA, 2003; Volume 4, pp. 2775–2778. [CrossRef]
27. Bayraktaroglu, E.; King, C.; Liu, X.; Noubir, G.; Rajaraman, R.; Thapa, B. Performance of IEEE 802.11 under Jamming. *Mob. Netw. Appl.* **2013**, *18*, 678–696. [CrossRef]
28. Wei, X.; Wang, T.; Tang, C. Throughput Analysis of Smart Buildings-oriented Wireless Networks under Jamming Attacks. *Mob. Netw. Appl.* **2021**, *26*, 1440–1448. [CrossRef]
29. López-Vilos, N.; Valencia-Cordero, C.; Azurdia-Meza, C.; Montejo-Sánchez, S.; Mafra, S.B. Performance Analysis of the IEEE 802.15.4 Protocol for Smart Environments under Jamming Attacks. *Sensors* **2021**, *21*, 4079. [CrossRef] [PubMed]
30. Jiaqi, C.; Jianfeng, Y.; Jun, X.; Chengcheng, G. Performance analysis of CSMA/CA in CSMA/TDMA mixed network. *Comput. Eng. Appl.* **2017**, *53*, 83–88+127.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.