sensors

MDPI

Article

Deep Reinforcement Learning for Optimizing Restricted Access Window in IEEE 802.11ah MAC Layer

Xiaojun Jiang [1,2], Shimin Gong [1,2], Chengyi Deng [1], Lanhua Li [1,*] and Bo Gu [1]

[1] School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-Sen University, Shenzhen 518107, China; jiangxj5@mail2.sysu.edu.cn (X.J.); gongshm5@mail.sysu.edu.cn (S.G.); dengchy9@mail2.sysu.edu.cn (C.D.); gubo@mail.sysu.edu.cn (B.G.)
[2] Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology, Guangzhou 510006, China
* Correspondence: lilh65@mail.sysu.edu.cn

Abstract: The IEEE 802.11ah standard is introduced to address the growing scale of internet of things (IoT) applications. To reduce contention and enhance energy efficiency in the system, the restricted access window (RAW) mechanism is introduced in the medium access control (MAC) layer to manage the significant number of stations accessing the network. However, to achieve optimized network performance, it is necessary to appropriately determine the RAW parameters, including the number of RAW groups, the number of slots in each RAW, and the duration of each slot. In this paper, we optimize the configuration of RAW parameters in the uplink IEEE 802.11ah-based IoT network. To improve network throughput, we analyze and establish a RAW parameters optimization problem. To effectively cope with the complex and dynamic network conditions, we propose a deep reinforcement learning (DRL) approach to determine the preferable RAW parameters to optimize network throughput. To enhance learning efficiency and stability, we employ the proximal policy optimization (PPO) algorithm. We construct network environments with periodic and random traffic in an NS-3 simulator to validate the performance of the proposed PPO-based RAW parameters optimization algorithm. The simulation results reveal that using the PPO-based DRL algorithm, optimized RAW parameters can be obtained under different network conditions, and network throughput can be improved significantly.

Keywords: IEEE 802.11ah; restricted access window (RAW); deep reinforcement learning (DRL)

Citation: Jiang, X.; Gong, S.; Deng, C.; Li, L.; Gu, B. Deep Reinforcement Learning for Optimizing Restricted Access Window in IEEE 802.11ah MAC Layer. *Sensors* **2024**, *24*, 3031. https://doi.org/10.3390/s24103031

Academic Editor: Francesco Mercaldo

Received: 28 March 2024
Revised: 7 May 2024
Accepted: 7 May 2024
Published: 10 May 2024

1. Introduction

With the rapid development of internet of things (IoT) applications and technologies, IoT has emerged as a pivotal enabler bridging the physical and digital realms. IoT has been widely used in industry, agriculture, healthcare, and other fields. Statistics show that IoT connected devices are expected to exceed 30 billion units by 2025, more than doubling from 13.8 billion in 2021 [1]. With the expanding scope of applications, IoT has its own set of requirements: very low power, longer-range connections, and support for a greater number of client devices per access point (AP) [2]. The fulfillment of these requirements relies on the selection of wireless communication technologies.

To meet the key requirements of IoT applications, the Wi-Fi Alliance has introduced Wi-Fi HaLow technology [3], which is based on the IEEE 802.11ah standard [4], operating in the unlicensed sub-1 GHz radio frequency spectrum band and utilizing narrower channels. IEEE 802.11ah is built upon the IEEE 802.11 standards with modifications for IoT applications. The physical (PHY) layer of IEEE 802.11ah is designed for long-range communication. At the medium access control (MAC) layer, novel channel access control mechanisms are introduced to facilitate access for a large number (up to 8191) of stations (STAs) and to support low power consumption. Leveraging the novel features at the PHY and MAC layers, IEEE 802.11ah offers up to 100 times longer range compared to other

IoT technologies, with a data rate ranging from approximately 150 kbps to a maximum of around 86.7 Mbps [4]. As shown in Figure 1, IEEE 802.11ah-based Wi-Fi HaLow technology provides a well-balanced combination of data rate, coverage range, and energy efficiency, outperforming low-power IoT technologies such as LoRa, NB-IoT, and Zigbee [3]. Wi-Fi HaLow also features easier deployment and integration into IP networks compared to other technologies, with scalability similar to LoRa. Therefore, Wi-Fi HaLow is well-suited to meet the key requirements of IoT applications.

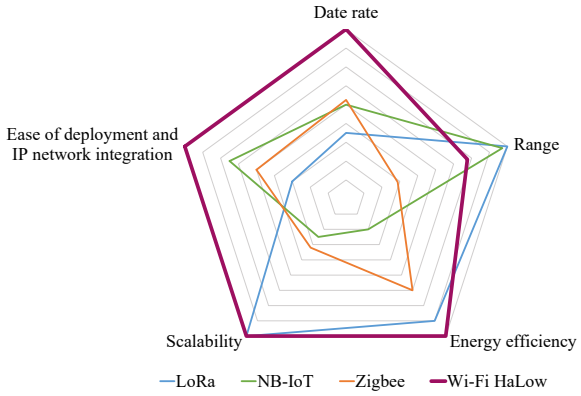

Figure 1. Comparison of the IEEE 802.11ah-based Wi-Fi HaLow technology with other low-power IoT technologies in terms of key aspects.

In the MAC layer, the restricted access window (RAW) mechanism is introduced to manage the significant number of STAs accessing the network [4]. The idea of RAW is to divide the channel time into one or more access windows, where only some of the STAs can access the channel in the designated access windows, while the others are restricted from random access. As shown in Figure 2, for STAs with certain traffic patterns, the AP divides them into one or more RAW groups during a traffic indication map (TIM) beacon interval. On the arrival of each RAW, the STAs assigned to the current RAW have the right to access the channel for data transmission, while the other STAs remain dormant and cache non-urgent data until the arrival of their corresponding RAW. To further alleviate contention, each RAW is subdivided into multiple time slots with equal duration. The STAs are uniformly distributed among these slots by default. During each slot, only the STAs assigned to the current slot are permitted to contend for data transmission, ensuring that STAs restricted in different slots do not conflict with each other.

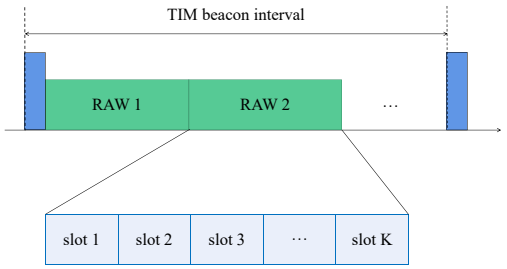

Figure 2. A simple demonstration of RAW.

The operation of RAW mainly consists of two parts. One is the division of STAs into different RAW groups. The other is the configuration of RAW-related parameters, including the number of RAW groups, the number of slots in each RAW, the duration of each RAW, and the number of STAs in each RAW group. Different RAW parameters can change the users' transmission strategies and thus influence the network performance, such as network throughput, latency, and energy efficiency [5,6]. However, details about RAW parameters setting and RAW grouping are not specified in the IEEE 802.11ah standard. This allows researchers the flexibility to customize the RAW configuration to meet the specific

requirements of different application scenarios. Moreover, the performance of RAW can be validated in an NS-3 simulator. In [5], the authors constructed simulation environments for IEEE 802.11ah sensor networks in an NS-3 simulator that closely resembled real-world network conditions. Through simulations, detailed analyses of the impacts of RAW parameters (i.e., number of RAW groups, RAW group duration, and station division) on network throughput, transmission delay, and energy consumption have been conducted in the literature. The experiments in [6] also revealed that network performance largely depends on these RAW parameters settings.

Based on this observation, some studies have focused on finding the optimal RAW parameters to improve network performance. Researchers have conducted complicated mathematical models and have proposed heuristic methods to determine the optimal RAW parameters or grouping scheme [7–9]. However, most of the analytical models fail to consider the complexities and dynamic changes of network conditions, leading to discrepancies between the results derived by analytical models and those obtained from an NS-3 simulator. Moreover, heuristic methods for optimizing RAW parameters are often constrained by specific assumptions, such as fixed network topologies and known traffic patterns. The applicability of these methods in various scenarios requires further validation. Therefore, this paper aims to propose a flexible model-free learning method for finding the optimal RAW parameters, which is scalable, robust, lightweight, and capable of generalizing across different scenarios.

Due to high efficiency and strong generalization capabilities, artificial intelligence (AI) methods have found broad applications in wireless networks in recent years. There is a growing number of studies employing AI methods to solve RAW parameters optimization and grouping problems. Researchers in [10] used neural networks to decide the optimal number of RAW groups and the number of slots in each RAW for given network conditions. Moreover, machine learning (ML) methods such as K-means have been used to solve grouping problems [11]. It is noteworthy that deep reinforcement learning (DRL) integrates deep learning (DL) and reinforcement learning (RL) by using deep neural networks (DNNs) to approximate value functions or optimal policies, thereby enabling the handling of high-dimensional and complex state and action spaces. DRL's strong performance in dealing with complex and dynamic environments endows it with powerful generalization capability, making it widely applied in wireless networks for solving parameterized optimization problems such as resource allocation and scheduling [12,13]. Therefore, it is feasible to employ the DRL approach to solve RAW parameters and network performance optimization problems.

In this paper, we propose a DRL method referring to the proximal policy optimization (PPO) algorithm to optimize the configuration of RAW parameters including the number of RAW groups, the number of slots in each RAW, and the duration of each slot, in the uplink IEEE 802.11ah-based IoT network. To improve the AP's data collection, we aim to enhance the throughput for the overall network. Note that the proposed model-free PPO-based DRL algorithm is flexible and capable of generalizing across different scenarios. It can be easily extended to other RAW parameter optimization problems that aim to enhance other performance metrics, such as latency and energy efficiency. Specifically, we propose an efficient DRL algorithm to optimize RAW parameters to enhance network throughput. We construct different network environments using an NS-3 simulator and evaluate the learning performance of the proposed PPO-based algorithm in different scenarios. To the best of our knowledge, there are limited prior studies on RAW mechanism optimization using DRL-based approaches. Our study can serve as a reference for applying DRL to the RAW mechanism and further extensions for optimizing other mechanisms of IEEE 802.11ah. We summarize our contributions as follows:

- Performance modeling for RAW parameters optimization: The impact of RAW parameters on network throughput in the uplink IEEE 802.11ah-based IoT network is studied. To optimize network throughput, a performance analytical model is established, and a RAW parameter optimization problem is formulated.

- Guiding PPO-based DRL with NS-3 simulated network environments: An efficient learning framework is proposed to interact with different network environments constructed in an NS-3 simulator. The PPO-based DRL algorithm is designed to find the preferable RAW parameters to improve network throughput. The NS-3 simulator adeptly replicates real-world network scenarios, facilitating the training of the DRL agent. Simulation results demonstrate the effectiveness of the PPO-based DRL algorithm, with significant improvements in network throughput of 80% compared to that of the default settings schemes.

The remainder of this paper is organized as follows. Prior studies on RAW-based network performance optimization and related works using AI-based methods for the RAW mechanism are presented in Section 2. In Section 3, the network model considered in this paper and the operation of RAW are elaborated, throughput modeling with respect to RAW parameters is presented, and the RAW parameters optimization problem is established. The problem is reformulated as a Markov decision process (MDP), and a PPO-based DRL algorithm for RAW parameters optimization is proposed in Section 4. In Section 5, the performance of the proposed DRL algorithm is evaluated in simulation environments built in an NS-3 simulator. Conclusions are drawn and future studies are discussed in Section 6.

2. Related Work

2.1. Analytical Modeling for RAW Mechanism

To investigate the impact of the RAW mechanism on network performance, researchers have developed several evaluation models for the RAW-based channel access process. Typically, researchers introduce characteristics of RAW into the analytical model of the distributed coordination function (DCF) of IEEE 802.11 standards [14]. Given the known number of STAs in the network, researchers analyze the transmission and collision probabilities in a single RAW slot. The analysis is then extended to one or multiple RAWs to derive formulas for calculating network performance metrics such as throughput, delay, and energy consumption [7,15,16]. These analytical models are validated by comparing the results with those obtained from an NS-3 simulator. However, they require a series of assumptions, including saturated network traffic, ideal channel conditions, and packet loss solely caused by collisions. To obtain more accurate models, researchers have further taken unsaturated traffic, heterogeneous networks, signal capture, and other network conditions into account [6,17,18].

Moreover, researchers have investigated the impact of different RAW parameters on network performance based on analytical models or simulation results. The authors in [19] pointed out that dividing more RAWs in a beacon interval period can reduce collision probability as the total number of competing STAs in each RAW group decreases. However, this also leads to increased delay, as larger RAW segmentations increase the probability of packet buffering. Similarly, the authors in [18] stated that the more slots divided in each RAW, the fewer number of STAs competing for channel access in a single slot, thereby reducing the probability of collisions. However, the time overhead increases due to the non-cross-slot-boundary setting. The authors in [5] emphasized that a longer RAW duration generally results in better throughput. However, excessively long RAW durations perform worse in terms of latency. Moreover, the duration of a RAW should be determined based on the traffic load in each RAW group. The critical impact of RAW duration on network performance was further discussed in [6].

2.2. Optimization in RAW Mechanism

Given the critical impact of RAW parameters on network performance, an important issue in optimizing the RAW mechanism is the optimization of RAW parameters. RAW parameters include the number of RAW groups, the number of slots in each RAW, the duration of each RAW (which can be calculated given the slot count and slot duration), and the number of STAs in each RAW group (which can be calculated given the number of RAW groups and STAs in the network). It has been validated that the optimization of

RAW parameters depends on various network variables, such as number of STAs, traffic load, and traffic patterns [5]. Most of the studies are network performance optimization-oriented, in which the authors formulate RAW parameters optimization problems and obtain one or more optimal RAW parameters using various optimization methods. To jointly maximize uplink energy efficiency and delay, the authors in [7] proposed an energy-delay-aware-window control algorithm based on the gradient descent method, enabling adaptive adjustment of slot count and slot duration according to the number of STAs in each RAW group. Similarly, the authors in [20] proposed a group-size-adaptive algorithm to determine the duration of each RAW. To cope with dynamic changes in the network size and heterogeneous traffic conditions in sensor networks with uplink traffic, the authors in [21] proposed TAROA, which can adaptively adjust RAW parameters according to the current (or estimated) traffic conditions and assign STAs to different RAW groups based on the estimated transmission frequency. TAROA has been further refined in [22]. Oriented towards delay-sensitive emergency alarm sensor networks and closed-loop communication scenarios, the authors in [23] proposed a RAW parameters selecting algorithm to minimize channel time-sharing consumption. Additionally, in [8], the authors formulated the optimal RAW scheduling problem as an integer nonlinear programming problem with the objective of minimizing channel time at key STAs and designed a heuristic algorithm to find the optimal RAW configurations.

Moreover, some studies have focused on RAW grouping, which allocates STAs to different RAW groups based on the various characteristics of the STAs. According to the priority level of the STAs, the authors in [24] proposed a QoS-aware priority grouping and scheduling algorithm. Considering the traffic characteristics (e.g., traffic demand, multi-rate) of STAs in heterogeneous networks, the authors in [25] proposed MoROA, which employs mathematical methods to solve the grouping problem and to determine the optimal RAW configurations. To achieve fairness in inter-group throughput and channel utilization the authors in [9,26] proposed heuristic grouping algorithms. Furthermore, in [27,28], the authors introduced grouping strategies based on greedy algorithms and on genetic algorithms, respectively.

2.3. AI-based Methods for RAW Mechanism

It is noteworthy that in recent years there has been a growing number of studies employing AI methods to solve RAW parameter optimization and grouping problems. The authors in [29] proposed a surrogate model for RAW performance in realistic IoT scenarios by integrating ML methods such as support vector machine and artificial neural networks (ANNs). This model accurately predicts network performance for given RAW configurations in heterogeneous networks. The predicted values can serve as inputs for real-time RAW parameters optimization algorithms, thereby enhancing algorithm accuracy. In [10], the authors used ANNs to find the optimal number of RAW groups given the network size, data rate, and RAW duration. Using ML methods such as K-means, the authors implemented traffic classification and grouping schemes that can dynamically adapt to various network conditions (e.g., received signal strength, multiple rates, traffic load, and traffic arrival interval) [11,30–32]. In a recent study [33], the authors employed a recurrent neural network based on gated recurrent units to estimate the optimal number of RAW slots, enhancing the performance in dense IEEE 802.11ah IoT network. To the best of our knowledge, there are limited prior studies using DRL methods for RAW mechanism optimization.

3. RAW Mechanism in Wireless IoT Networks

In this paper, we consider uplink data transmissions in a wireless IoT network employing the RAW mechanism. As shown in Figure 3, the network consists of one center-located AP and N randomly distributed STAs within a coverage range of several hundred meters. The STAs transmit sensory data to the AP using a specific channel access control protocol. The network traffic includes periodically generated data, as well as randomly generated

data following a certain probability distribution. Since the IEEE 802.11ah standard is an ideal choice for low-power IoT networks, we employ the IEEE 802.11ah-based RAW mechanism for multiple STAs access. We further describe the RAW mechanism operating in the IoT network.

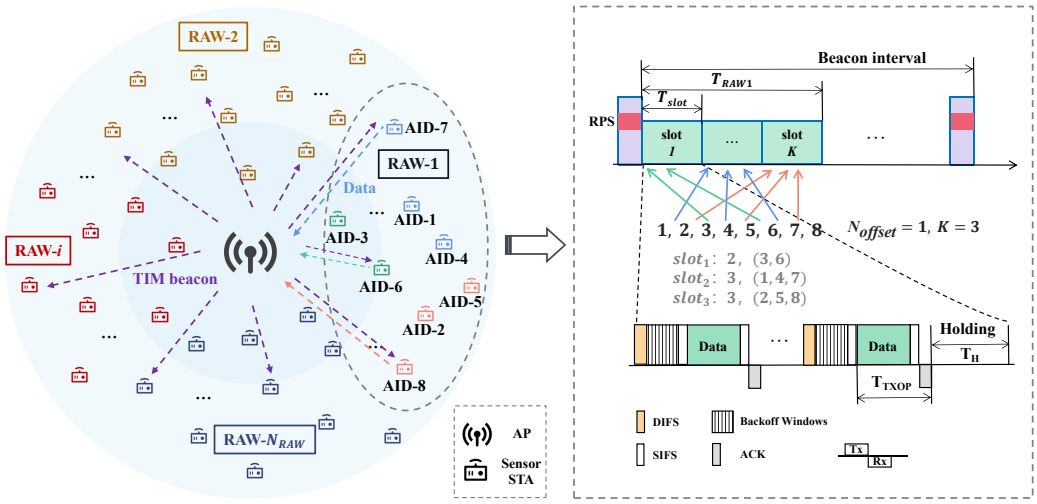

Figure 3. The IEEE 802.11ah-based IoT network model with RAW operations.

3.1. Operation of the RAW Mechanism

In Section 1, we briefly introduced the idea of RAW. In this section, we elaborate on the RAW parameter set involved in RAW configuration and the channel access process based on RAW in a beacon interval. We aim to explain how key RAW parameters influence network performance at the mechanism principle level.

3.1.1. Structure of the RAW Parameter Set

The IEEE 802.11ah standard defines an information element field in the beacon frame for group-based restricted channel access, known as the RAW parameter set (RPS) [4]. In general, the operation of RAW is mainly implemented through the definition of the RPS in a TIM beacon, the slot allocation scheme, cross slot restrictions, and other necessary mechanisms. In IEEE 802.11ah networks, once the STAs join the network and are assigned their association identifier (AID) they listen for TIM beacon frames that carry the RPS elements, which are periodically broadcast by the AP. Consequently, the STAs in the network can know exactly the status of RAW and and their membership in a RAW group, enabling them to perform channel access and data transmission accordingly.

Specifically, RPS primarily consists of one or more RAW assignment subfields. Each RAW assignment subfield contains necessary RAW control subfields, RAW slot definition subfields, and RAW grouping subfields, for performing restricted channel access to one or multiple STAs in a RAW. According to specific requirements, elements such as RAW start time, channel indication, and periodic operation parameters subfields are conditionally present. The RAW slot definition subfield further defines the slot duration, slot count, and access restrictions between slots. As beacon frames are broadcast by the AP, STAs can learn from the related subfields of the RPS element which RAW group they belong to, as well as the number of RAW groups in a beacon interval, the number of slots in each RAW, and the duration of a single slot in each RAW. The specific rules for calculation are described as follows.

(1) Slot duration and slot count: The formula for calculating the duration of a single slot in a RAW is as follows [4]:

$$T_{slot} = 500us + C \times 120us. \quad (1)$$

Let the length of the slot duration count field be y. According to the IEEE 802.11ah standard, when $y = 11$ bits, $C = 2^{11} - 1 = 2047$, the maximum duration of a slot is $T_{slot}^{max} = 246.14$ ms

and the maximum number of slots in a RAW is $K_{max} = 2^{14-y} = 7$. When $y = 8$ bits, $C = 2^8 - 1 = 255$, $T_{slot}^{max} = 31.1$ ms, $K_{max} = 2^{14-y} = 63$. The selection of y depends on the number of STAs in each RAW. Apparently, the duration of a RAW can be calculated as $T_{RAW} = K \cdot T_{slot}$.

(2) Slot assignment: A mapping method for allocating STAs into the corresponding slots in a RAW is defined in the IEEE 802.11ah standard [4]. It is implemented by defining a mapping function,

$$i = f(x) = (x + N_{offset}) \mod K, \tag{2}$$

where x is the AID of the STA in a RAW group, N_{offset} is the allocation offset, which means that the first STA in the group will be allocated to the $N_{offset} - th$ time slot, and K is the number of slots in a RAW.

We provide an illustration of slot allocation in a RAW as shown in Figure 3. We assume that a RAW group division scheme configured in the RPS divides a beacon interval into N_{RAW} RAW groups, with potentially different numbers of STAs, slots, and slot durations in each group. Based on the RPS settings, STAs with AIDs 1 to 8 are assigned to RAW-1 in order, with the first STA in RAW-2 being AID-9, and so on. In the RAW groups, STAs are sequentially assigned to different slots according to the mapping function. We assume the mapping offset $N_{offset} = 1$ and the number of slots $K = 1$. Consequently, in RAW-1, two STAs (with AID-3 and AID-6) are assigned to Slot-1, three STAs (with AID-1, AID-4, and AID-7) are assigned to Slot-2, and four STAs (with AID-2, AID-5, and AID-8) are assigned to Slot-3. The mapping function ensures a uniform distribution of STAs across slots.

(3) Cross slot boundary: The IEEE 802.11ah standard defines restrictions on channel access across slot boundaries. STAs can access the channel either in a cross-slot-boundary way or in a non-cross-slot-boundary (NCSB) way [4]. To alleviate the hidden nodes problem and facilitate performance analysis, it is generally advisable to employs the non-cross-slot-boundary mechanism [16]. Therefore, the holding time is defined to be $T_H \geq T_{TXOP}$, where T_{TXOP} is the time required for one successful data transmission, and its expectation can be obtained through statistical analysis. With this constraint, it can be ensured that the last data transmission in the current slot has been completed by the end of slot. If the time remaining in the current slot is not sufficient for one data transmission, the STAs cache their data and wait for the arrival of the next slot to which they belong.

3.1.2. RAW-Based Channel Access and Data Transmission

The channel access and data transmission process of STAs in an IEEE 802.11ah network with a RAW mechanism can be summarized as follows and is shown in Figure 3.

1. The STAs listen to the beacon frames broadcast by the AP, request association and authentication, and receive their AID. The AP periodically broadcasts beacon frames carrying the RPS element and informs the STAs of information including their RAW group, the slot count in a RAW, and the slot duration. The STAs are then assigned to different slots based on the mapping function (2).
2. The STAs contend for channel access following the enhanced distributed channel access (EDCA) mechanism when their slot arrives: the STAs perform carrier listening for a distributed inter-frame spacing (DIFS) time before initiating channel access. Once the channel is sensed to be idle, the STAs start decreasing their backoff counter, and they initiate channel access when their backoff counter reaches zero. If STA_a's backoff counter decreases to zero before STA_b's, STA_a initiates channel access, while STA_b suspends its backoff counter until the channel is sensed to be idle again.
3. If the backoff counters of two or more STAs in the network decrease to zero simultaneously, these STAs attempt to access the channel at the same time, which may result in collisions. Upon encountering a collision, the STAs increase and reset their backoff counter until they reach the maximum retry limit, at which point packet loss occurs.
4. STAs that successfully access the channel will transmit their data after waiting for a short inter-frame spacing (SIFS) time. A received acknowledgment (ACK) frame from

the AP indicates the completion of data transmission. The time taken for one data transmission is denoted as T_{TXOP}.

The operation of the RAW mechanism elaborated above can provide a preliminary explanation at the mechanism level for the significant impact of RAW parameters on network performance: Firstly, when the number of STAs in the network is given, the number of RAW groups and the number of slots in each RAW jointly determine the number of STAs contending for channel access in a slot. Constrained by the DCF mechanism, a large number of STAs contending for channel access in a slot will intensify collisions among STAs, thereby affecting system throughput. Moreover, the duration of a slot determines the maximum number of data transmissions that can occur in each slot. When the network size increases, inadequate slot duration will limit the amount of data that STAs can transmit per slot, consequently reducing overall throughput. Due to the limitation of the NCSB mechanism, an excessive number of slots can result in frequent slot boundary switches, which in turn increases the holding time overheard and the data buffering. In general, RAW parameters, including the number of RAW groups, the number of slots per RAW, and the duration of each slot in a RAW, significantly influence network throughput. In the next subsection, we will analyze the impact of RAW parameters on network throughput at the mathematical analysis level.

3.2. Performance Modeling for RAW Parameters Optimization

We assume that the number of RAW groups is denoted as N_{RAW}, the number of slots in each RAW group is represented by k_i, and the duration of a slot in each RAW group is denoted as t_i, where $i \in [1, N_{RAW}]$. Thus, the set of the number of STAs in each RAW group, the set of the number of slots in each RAW, and the set of slot durations for RAW groups are represented as $\boldsymbol{N}_{STA} = \{n_1, \ldots, n_i, \ldots, n_{N_{RAW}}\}$, $\boldsymbol{K}_{RAW} = \{k_1, \ldots, k_i, \ldots, k_{N_{RAW}}\}$, and $\boldsymbol{T}_{RAW} = \{t_1, \ldots, t_i, \ldots, t_{N_{RAW}}\}$, respectively.

The correlation between RAW parameters and network throughput can be derived based on the analytical model proposed in [14]. Given that the STAs are uniformly distributed among slots in a RAW, the number of STAs in each slot can be approximated as $x_i = \frac{n_i}{k_i}$, and the intensity of contention in each slot is considered to be the same. Consequently, for the STAs in each slot of the i-th RAW, the probability of STAs suspending their backoff counter is defined as $p_{f,i}(\tau_i, x_i, t_i)$, indicating that the suspending probability is related to the transmission probability τ_i, the number of STAs in each slot x_i, and the slot duration t_i. The collision probability is denoted as $p_{c,i}$.

The backoff process of an STA's backoff counter can be analyzed using a two-dimensional Markov chain [14]. Each state during the backoff process can be represented as a probability, and the steady-state probability of each state can be further determined. According to the normalization formula, a closed-form expression for the steady-state probability of the backoff counter decreasing to zero can be obtained as $b_{i,0}(p_f, p_c, CW_{min}, m)$, indicating that the steady-state probability at state-0 is dependent on the suspending probability $p_{f,i}$, the collision probability p_c, the given minimum size of the contention window CW_{min}, and the retry limit m. Subsequently, the transmission probability can be computed as

$$\tau_i = \frac{1 - (p_{c,i})^{m+1}}{1 - p_{c,i}} b_{i,0}. \tag{3}$$

The collision probability is given by $p_{c,i} = 1 - (1 - \tau_i)^{x_i - 1}$, and the probability that at least one STA transmits data in a slot is denoted as $P_{tr,i} = 1 - (1 - \tau_i)^{x_i}$. Furthermore, the successful transmission probability can be represented as

$$P_{suc,i} = \frac{x_i \tau_i (1 - \tau_i)^{x_i - 1}}{P_{tr,i}}. \tag{4}$$

The normalized slot throughput can be calculated as

$$u_i = \frac{P_{tr,i} P_{suc,i} E(D)}{(1 - P_{tr,i})\sigma + P_{suc,i} T_{suc} + p_{c,i} T_c}, \tag{5}$$

where $E(D)$ represents the average payload size of a data frame and σ is the time of a mini-slot in the contention window. The time for a successful data transmission and the time spent due to collision are denoted as T_{suc} and T_c, respectively, and are calculated in [14]. The effective time for data transmissions in a slot is $t_i^{'} = t_i - T_H$. Finally, the normalized throughput of the network can be denoted by

$$U = \sum_{i=1}^{N_{RAW}} \frac{u_i k_i t_i^{'}}{T_{BI}}, \tag{6}$$

where the duration of the beacon interval T_{BI} is dependent on the total duration of RAWs in one beacon interval.

According to (6), network throughput is related to successful transmission probability, which in turn depends on collision probability and transmission probability. These probabilities are influenced by the number of STAs in a slot and the slot duration. Moreover, the number of RAW groups and the number of slots in a RAW jointly determine the number of STAs in a slot. Intuitively, the increasing number of RAW groups and slot divisions reduces the number of STAs per slot, thereby decreasing the collision probability. Increasing the slot duration, on the other hand, allows more time for data transmission in a slot, thereby reducing data buffering. Therefore, increasing the number of RAW groups, dividing more slots in a RAW, and extending the slot duration can greatly enhance network throughput. However, excessive RAW divisions may cause more STAs to remain idle, leading to data buffering. Similarly, an excessively long slot duration may result in wasted time in networks with low traffic loads. There is a trade-off in adjusting the RAW parameters. Hence, by jointly optimizing the number of RAW groups N_{RAW}, RAW slot counts $k_i \in \boldsymbol{K}_{RAW}$, and slot durations $t_i \in \boldsymbol{T}_{RAW}$ with $i \in [1, N_{RAW}]$, we can formulate the network throughput maximization problem as follows:

$$\begin{aligned} \max_{N_{RAW},\, \boldsymbol{K}_{RAW},\, \boldsymbol{T}_{RAW}} \quad & U \\ s.t. \quad & (1),\ (3),\ (4),\ (5),\ \text{and}\ (6) \\ & \sum_i k_i t_i \le T_{BI}. \end{aligned} \tag{7}$$

The existing studies prefer to construct complicated analytical models of RAW, and they further propose optimization methods to find the optimal RAW parameters to improve network throughput. However, solving RAW parameters optimization problems based on analytical models may lead to a high level of computational complexity or even impracticality in dynamic networks. On the one hand, these analytical models require a series of assumptions, including saturated network traffic, ideal channel conditions, and packet loss solely caused by collisions. Moreover, the analytical models do not comprehensively consider details about the RAW mechanism and channel conditions. Although some studies have refined the analytical models and taken more complex network conditions into account, this has made the analysis process more cumbersome. On the other hand, because the mathematical or heuristic methods often involve complex rules and have not been validated in different network scenarios, their generalization ability in complex and dynamic network conditions needs to be improved.

To investigate practical network states, the IEEE 802.11ah network simulation environment was developed based on a widely used network simulator called NS-3 [5]. NS-3 is used to create simulation environments that closely resemble real-world network environments. The partial mechanisms of the PHY and MAC layers including the RAW mechanism

are also implemented. While analytical results serve as references for optimizing RAW parameters, the simulation environment implemented by NS-3 undoubtedly provides more accurate results and can serve as a benchmark for validating these analytical results. Additionally, with the capability of handling complex and dynamic environments, DRL-based methods are well-suited for addressing RAW parameters optimization problems and demonstrate strong generalization ability across various scenarios.

To determine the preferable RAW parameters that improve network throughput in complex network environments resembling real-world scenarios, we construct network environments using an NS-3 simulator, and employ the DRL-based method to optimize the RAW parameters for enhanced network throughput. The specific methodology will be elaborated in the following section.

4. DRL for RAW Parameters Optimization

In this section, we propose a learning framework for optimizing RAW parameters. As depicted in Figure 4, we set up network simulation environments in NS-3 and execute agent training in the DRL environment. The PPO algorithm [34] is employed as the specific implementation algorithm in the DRL framework for optimizing RAW parameters in NS-3, achieving enhanced learning efficiency and policy update stability. During training, the DRL agent receives network observations from the NS-3 simulation environment, serving as inputs to the DNNs. Each learned action (i.e., the RAW parameters) is then applied as the configuration parameters for the RAW mechanism in NS-3, and a new simulation is executed to obtain an updated reward (i.e., the network throughput). The DRL agent continues to receive observations from the network environment for a new training episode. Interactions between the DRL agent and the NS-3 environment continue until the DRL agent learns the preferable RAW parameters and achieves enhanced network throughput. To utilize PPO for optimizing the RAW parameters to maximize network throughput, we first reformulate problem (7) as an MDP.

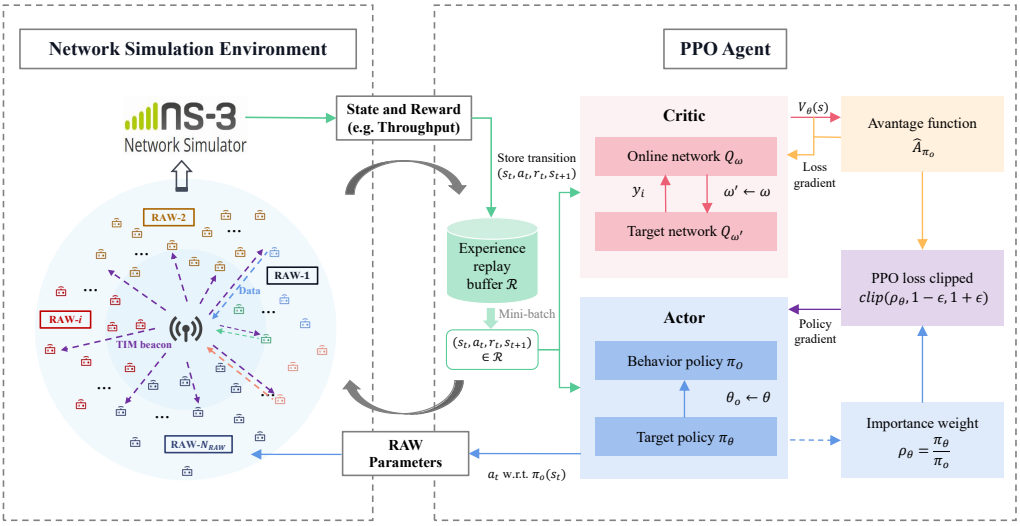

Figure 4. DRL framework for optimizing RAW parameters in NS-3.

4.1. MDP Reformulation

Given the network conditions, we aim to optimize the RAW parameters (i.e., the number of RAW groups, the number of slots in each RAW, and the slot duration in each RAW) to reduce contentions among the STAs and consequently improve the network throughput. To facilitate the problem formulation, the following assumptions are made for the IoT network:

1. The AP collects information about the network (e.g., network size N and traffic arrival of the STAs) through management frames. Based on received packets from

the STAs, the network performance, such as throughput and packet loss ratio, can be statistically determined.
2. To alleviate hidden nodes issues and collisions, the STAs obey the NCSB mechanism when accessing the channel among slots.

In RL, the interaction between the agent and the environment is typically modeled as an MDP, which can be represented by a tuple $(\boldsymbol{S}, \boldsymbol{A}, P, r, \gamma)$, where $\boldsymbol{S}$ represents the state space, $\boldsymbol{A}$ represents the action space, the transition probability function $P(s^{'}|s,a)$ represents the probability of transitioning from state s to state $s^{'}$ when action a is taken, the reward function $r(s,a)$ represents the reward obtained after taking action a in state s, and $\gamma \in [0,1]$ is a constant discount factor. Specifically, the definitions of state, actions, reward, and observations are given as follows:

1. State: The state at the current time step is defined as the throughput obtained from the current simulation statistics, denoted as $s_t = U_t$. During the simulation, the AP collects the number of packets received and the payload size of each packet at the current time step to calculate the network throughput at the end of the current step.
2. Action: The actions in the MDP are defined as the RAW parameters, including the number of RAW groups, the number of slots in each RAW group, and the slot duration in each RAW group. Thus, the action at step t is denoted as $a_t = (N_{RAW}, \boldsymbol{K}_{RAW}, \boldsymbol{T}_{RAW})$.
3. Reward: According to the optimization objective, the reward is defined as the throughput obtained at each time step, represented as $r_t = U_t$.
4. Observation: The observation set is defined as the network information observable by the AP, including network size N, the set of traffic loads $\boldsymbol{D}$, and the set of traffic intervals $\boldsymbol{I}$, which can be represented as $o_t = (N, \boldsymbol{D}, \boldsymbol{I})$.

In the following subsection, we elaborate on the PPO algorithm for RAW parameters optimization.

4.2. PPO for Optimizing RAW Parameters

Given a policy approximator $\pi_\theta(a|s)$ with parameters θ, policy-based policy gradient (PG) algorithms find the optimal θ to maximize the reward or value function [35]. For a given input state s_t, the policy network directly outputs either the action or the probability associated with the action. It then selects the appropriate action based on the probability, allowing the output action to be a continuous value. The expected value function in PG algorithms can be represented in terms of the policy parameters as

$$J(\theta) = \sum_s d^{\pi_\theta}(s) V^{\pi_\theta}(s) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a) Q^{\pi_\theta}(s,a), \tag{8}$$

where d^{π_θ} is the stationary distribution of the Markov chain for π_θ, and $Q^{\pi_\theta}(s,a)$ denotes the Q-value of the state–action pair (s,a) following the policy π_θ. The goal of PG is to find parameters θ that maximize $J(\theta)$ by ascending the gradient of the policy. The evaluation of the policy gradient $\nabla_\theta J(\theta)$ can be simplified as [36]

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[Q^\pi(s,a) \nabla_\theta \ln \pi_\theta(a|s))] = \mathbb{E}_\pi[\sum_{t=1}^{T} Q^\pi(s_t,a_t) \nabla_\theta \ln \pi_\theta(a_t|s_t))], \tag{9}$$

where the expectation is taken over all possible state–action pairs following the same policy π_θ. The policy gradient $\nabla_\theta J(\theta)$ can be evaluated by sampling historical decision-making trajectories.

For each episode, all the $(s,a,r,s^{'})$ tuples acquired by the agent can be collectively represented as a state–action trajectory resulting from the agent's interaction with the environment over the current episode, which is denoted as $\tau = (s_0, a_0, r_1, s_1, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim (\pi_\theta, P(s_{t+1}|s_t,a_t))$. Let $G_t = \sum_{k=t}^{T} r(s_k,a_k)$ be the reward for a trajectory τ, and estimate the Q-value $Q^\pi(s_t,a_t)$ in (9) by G_t. Therefore, the policy gradient in each time step can be approximated by randomly sampling $G_t \nabla_\theta \ln \pi_\theta(a_t|s_t))$, and the policy parameters can be

updated as $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$, where α denotes the step size for the gradient update. To reduce prediction variability and improve learning efficiency, the value function $V_\pi(s)$ can be used as the baseline, and the advantage function $A_\pi(s,a) \triangleq Q^\pi(s,a) - V_\pi(s)$ is further introduced to replace G_t.

To address high-dimensional state and action spaces while stabilizing the learning process, actor–critic (AC)-based DRL algorithms introduce a DNN with weight parameters ω to approximate the Q value. AC algorithms update both the policy network and the Q-value network. Specifically, at each learning step t, the actor updates the policy network by updating the policy parameters $\theta \leftarrow \theta + \alpha_\theta Q_\omega(s,a) \nabla_\theta \ln \pi_\theta(a|s))$, while the critic updates the Q network by minimizing a loss function and updates the parameters $\omega \leftarrow \omega + \alpha_\omega \delta_t \nabla_\omega Q_\omega(s,a)$ by gradient ascent, where $\delta_t = r_t + \gamma Q_\omega(s^{'}, a^{'}) - Q_\omega(s,a)$ denotes the TD error. To further stabilize the training process, the deep deterministic gradient policy (DDPG) algorithm [36] utilizes two DNNs with different parameters, i.e., the online Q-network $Q_\omega(s,a)$ and the target Q-network $Q_{\omega'}(s,a)$. The TD error is rewritten as $\delta_t = r_t + \gamma Q_{\omega'}(s^{'}, a^{'}) - Q_\omega(s,a)$.

To facilitate the agent's utilization of past experiences and improve sample efficiency, PG can be transformed into off-policy learning through the utilization of importance sampling [37]. Sample collections can be conducted under a behavior policy $\pi_o(s,a)$ distinct from the target policy $\pi_\theta(a|s)$.

To mitigate the effects of improper step size in policy optimization on training stability, the off-policy trust region policy optimization (TRPO) algorithm imposes an additional constraint on the gradient update [37], ensuring that the old and new policies do not diverge significantly. Let $\rho_{theta} = \frac{\pi_\theta(s,a)}{\pi_o(s,a)}$ denote the probability ratio of the divergence between the old and new policies. TRPO maximizes the objective by applying conservative policy iteration without limiting the probability ratio to an appropriate range. This could lead to an excessively large policy update. Intuitively, a smaller deviation between the behavior policy and the target policy is better. Hence, the PPO algorithm [34] modifies the objective by constraining ρ_θ in a region $[1-\epsilon, 1+\epsilon]$ and penalizing changes to the policy that move ρ_θ away from 1. The objective in PPO_{CLIP} is

$$\begin{aligned} \max_{\theta} \quad & L^{CLIP}(\theta) = \tilde{J}(\theta) = \mathbb{E}_{\pi_o}[\min\{\rho_\theta \hat{A}_{\pi_o}, clip(\rho_\theta, 1-\epsilon, 1+\epsilon)\hat{A}_{\pi_o}\}] \\ s.t. \quad & D_{KL}(\pi_o, \pi_\theta) \leq \delta_{KL}, \end{aligned} \tag{10}$$

where $D_{KL}(P_1, P_2) \triangleq \int_\infty^\infty P_1(x) \log(P_1(x)/P_2(x))\, dx$ denotes a distance measure in terms of the Kullback–Leibler (KL) divergence between two different probability distributions. The advantage function A_{π_o} in the objective of problem (10) is the approximation of the actual advantage A_{π_θ} corresponding to the target policy π_θ. PPO constrains the parameters search within a region by introducing the inequality constraint in problem (10), which ensures that the KL convergence between π_θ and π_o is bounded by δ_{KL}. The clip function returns $\rho_\theta \in [1-\epsilon, 1+\epsilon]$ and the hyper-parameter $\epsilon = 0.2$ by default.

During the training of the DNNs, PPO employs fixed-length (e.g., T time steps) trajectory segments. A truncated advantage estimation is computed to replace the advantage function in problem (10) as

$$\hat{A}_t^{\pi_o} = \delta_t + \gamma \delta_{t+1} + \cdots + \gamma^{T-t+1} \delta_{T-1}, \tag{11}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. The loss function of PPO is

$$L_t(\theta) = \mathbb{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \tag{12}$$

where $L^{VF} = V_\theta(s_t) - V_t^{target}$ is the mean squared error loss, c_1, c_2 are coefficients, and S is an entropy bonus.

The PPO algorithm for optimizing RAW parameters in networks built in NS-3 is summarized in Algorithm 1. PPO utilizes two DNNs to approximate the policy networks.

In each learning episode, the PPO agent runs the old/behavior policy π_{θ_o} (i.e., RAW parameters), observes network throughput obtained from the NS-3 network simulations environment for T time steps, and stores T transition tuples $(s_t, a_t, r_t, s_{t+1}), t \in T$ in the experience replay buffer. Then, it samples mini-batches of transition tuples from the replay buffer and computes advantage estimates $\hat{A}_1^{\pi_o}, \cdots, \hat{A}_T^{\pi_o}$. Subsequently, the weight parameters of the target policy network π_θ are updated by using mini-batches randomly sampled from the replay buffer through importance sampling and by optimizing the surrogate loss in (12). The weight parameters of the behavior policy network are updated by $\theta_o \leftarrow \theta$. Limiting the probability ratio of the two policies $\rho_\theta \in [1-\epsilon, 1+\epsilon]$ ensures that the probability distribution of the output actions from the two policy networks remains similar.

Algorithm 1 PPO for RAW parameters optimization

```
Initialize target policy π_θ and behavior policy π_o
Initialize online critic Q_ω and target critic Q_ω'
Initialize clipping threshold ϵs
for episode= 1, ..., M do
  while t ≠ T do
    Observe the system state s_t from NS-3
    Select an action a_t according to behavior policy π_o(s_t)
    Execute action a_t = (N_RAW, K_{N_RAW}, T_{N_RAW}) in NS-3, obtain network throughput
    reward r_t = U_t, evaluate V_θ(s_t) and next state s_{t+1}
    Store transition tuple (s_t, a_t, r_t, s_{t+1}) and V_θ(s_t, a_t) in R
    t ← t + 1
  end while
  Sample mini-batch of transitions (s_i, a_i, V_θ(s_i, a_i), s_{i+1}) from R
  Estimate advantage Â_{π_o} using advantage according to (11)
  Update target policy by solving problem (10)
  Update behavior policy π_{θ_o} ← (1 − ϵ)π_{θ_o} + ϵπ_θ
  Update online and target critic by minimizing the value loss in (12) using gradient
  descent
end for
```

The uniform grouping scheme has been verified to perform better in homogeneous networks [38]. Considering the networks with periodic and random traffic in this paper, we employ the uniform grouping scheme, where STAs are evenly distributed in each RAW. Consequently, the slot duration and number of slots in each RAW group are considered to be equal. As a result, the actions of the MDP can be further simplified to the number of RAW groups, the number of slots in one RAW group, and the slot duration in one RAW group.

5. DRL-Guided NS-3 Simulation

In this section, we investigate the performance of the proposed PPO-based DRL algorithm for RAW parameters optimization in networks with periodic or random traffic, which are set up in the NS-3 simulator. We firstly demonstrate the learning performance of the PPO algorithm on finding preferable RAW parameters to enhance network throughput. Then, we investigate the adaptive capability of RAW parameters under dynamic network conditions such as traffic load and network size. Finally, we compare the performance of the PPO-based slot-division scheme with the equal-slot-division scheme (i.e., one STA per slot) and no-slot-division scheme (i.e., only one 'slot' in a RAW).

5.1. Simulation Setup

We set up the training environment for DRL on the Linux operating system. Specifically, we set up the DRL agent in a Python environment based on the PyTorch framework, and set up the network topology in the NS-3 simulator as depicted in Figure 3. Network conditions and simulation results are input into the PPO agent as environment states. The

RAW parameters are configured based on the actions learned by the PPO agent and used for subsequent simulations in the NS-3 simulator. Throughout the training process, the PPO agent interacts numerous times with the simulated network environment set up in the NS-3 simulator.

For the two different network environments established in NS-3, the two scenarios are primarily differentiated based on the network size, the traffic load of the STAs, and the traffic interval of the STAs, and are denoted as N, $\boldsymbol{D} = \{d_1, \ldots, d_N\}$, and $\boldsymbol{I} = \{i_1, \ldots, i_N\}$, respectively. These serve as the main environmental characteristics in the observations of the MDP established in Section 4.1.

The environment settings in the periodic traffic networks are summarized in Table 1. Specifically, we set small network sizes ($N < 100$). Each STA has the same traffic load, e.g., $d = 0.005$ Mbps. During data transmission, the packet transmission interval of each STA follows a fixed time interval, such as $i = 0.001$ ms. Parameters settings for the PPO agent are shown in Table 2. For analytical and simulation design purposes, we define the time step t as each fixed-duration simulation iteration performed in NS-3, where each episode consists of only one step. When performing simulations, we collect statistical information regarding network performance after every simulation iteration with a duration of 10 s. Note that the number of slots K_{RAW} is in the range of $[1, 63]$ according to the restriction in (1). When $K_{RAW} \in [1, 7]$, the maximum slot duration is 246.14 ms, and when $K_{RAW} \in [8, 63]$, the maximum slot duration is 31.1 ms. This can serve as a constraint for the agent during learning.

Table 1. Parameter settings in periodic traffic networks.

Parameters	Settings
Wi-Fi channel configuration	MCS 0, 2 MHz
coverage radius	300 m
data rate	650 kbit/s
traffic type	UDP
payload size	100 bytes
network size N	small, 60 (basic setting)
number of RAW group	1 (basic setting)
traffic load of the STAs	same
set of traffic loads $\boldsymbol{D}$	$d_1 = \ldots = d_N$
packet transmit interval of the STAs	periodic (same)
set of traffic intervals $\boldsymbol{I}$	$i_1 = \ldots = i_N$

5.2. Learning Performance in Periodic Traffic Networks

We first consider the RAW parameters optimization in networks with periodic traffic, where all STAs are assigned to one RAW group. The action of the MDP is $a_t = (K_{N_{RAW=1}}, T_{N_{RAW=1}})$. In each iteration during the training process, the PPO agent observes throughput s_t and other information o_t from the wireless environment and employs policy π to determine the RAW parameters setting a_t for the next time step. The effectiveness of the RAW parameters is evaluated at the subsequent time step with the reward obtained from NS-3, and the RAW parameters for the following time step are determined accordingly. The PPO agent is trained through numerous interactions with network simulation environments built in NS-3.

Table 2. Parameter settings for DRL training.

Parameters	Settings
state dimension	1
action dimension	$100 \times 63 \times 2047$
discount factor	0.99
generalized advantage estimation factor	0.95
PPO clip rate	0.2
PPO update times/epochs	10
max training episodes	1×10^4 (default)
hidden net width	256
learning rate of actor	3×10^{-4}
learning rate of critic	3×10^{-4}
L2 regularization coefficient for critic	1×10^{-3}
types of probability distribution of actor	beta
length of sliced trajectory of actor	64
length of sliced trajectory of critic	64
entropy coefficient of actor	0
decay rate of entropy coefficient of actor	0.9998

5.2.1. Convergence to the Preferable RAW Parameters

We first validate the convergence performance of the PPO algorithm on a basic network topology. In the periodic traffic network, the traffic interval of all STAs is fixed (e.g., 0.1 ms). We train the PPO agent though numerous interactions with the network environment built in NS-3. The convergence performance of the PPO algorithm is shown in Figure 5, and Figure 6 demonstrates the convergence process of the PPO agent interacting with the network simulation environment in the NS-3 simulator. As the training iteration proceeds, the PPO agent learned better actions, leading to significantly increased normalized rewards obtained from interacting with the NS-3 simulation environment. After 10,000 training episodes, the reward stabilized at its maximum value. This indicates that the PPO agent has learned the preferable RAW parameters by the end of training and has achieved the optimized network throughput in the periodic traffic network.

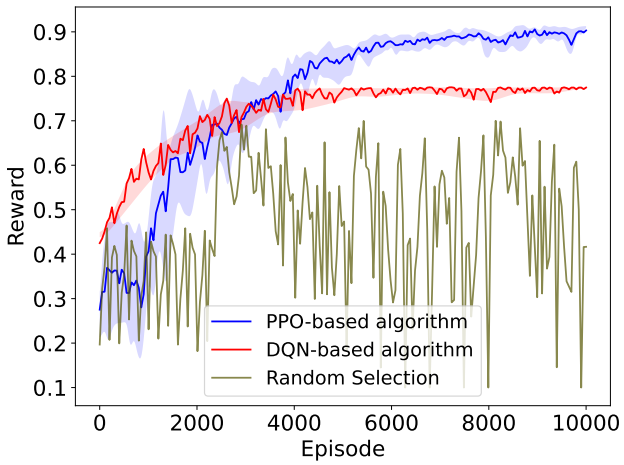

Figure 5. Convergence performance of the PPO-based algorithm compared with the DQN-based algorithm and the random selection scheme.

We also observe the improvement of network throughput with the NS-3 simulator during the PPO agent's training process. It can be seen in Figure 6 that the network throughput is ascending when the training process proceeds. Compared to the network throughput obtained with default settings ($K_{RAW} = 1$, $TBI = 100$ ms), the network throughput with the preferable RAW parameters obtained by the PPO agent is improved by about 70%. It is evident that employing the DRL method for optimizing RAW parameters is feasible, and that the RAW parameters derived from learning lead to a significant enhancement in network throughput compared to default settings.

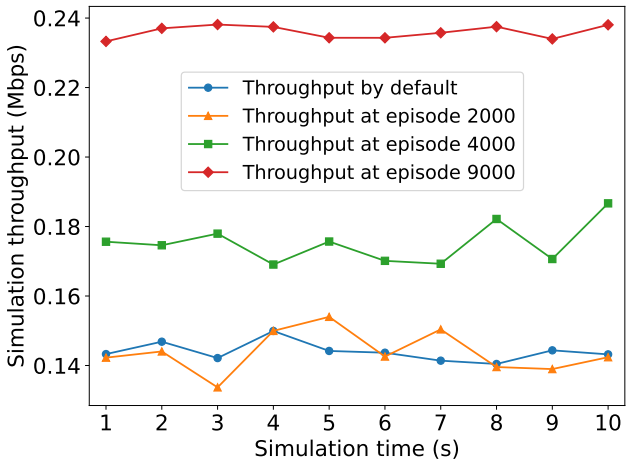

Figure 6. Throughput ascending in NS-3 simulation environment during training episodes.

To further validate the effectiveness of the proposed PPO-based algorithm, we compare it with the value-based Deep Q-Network(DQN) algorithm and the random RAW parameters selection scheme. DQN is suitable for discrete action learning but struggles with high-dimensional action spaces like RAW parameters. Therefore, we apply interval sampling to reduce the action space. As shown in Figure 5, compared to random selection, both DQN and PPO can converge to stable rewards through training, outperforming the random selection scheme. This observation highlights the ability of DRL methods to optimize RAW parameters and improve network throughput. Moreover, reducing the action space accelerates the convergence of DQN, requiring 50% fewer training episodes than PPO. However, this also leads to DQN's inferior performance, with a 20% lower stabilized reward than PPO.

Additionally, we depict the convergence performance of the slot count within a RAW and the slot duration with different numbers of STAs in the network. To provide a more straightforward demonstration, we calculate the approximate duration of a beacon interval as $T_{BI} = N_{RAW} \cdot K \cdot T_{slot}$, and we use beacon interval dynamics to represent variations in slot duration in the following subsections. As shown in Figures 7 and 8, both parameters converge to stable values for different network sizes, further validating the algorithm's convergence. As the network size is relatively small, the number of slots is similar when the number of STAs in the network is 40, 50, and 60, respectively. The duration of the beacon interval increases by about 40% when the number of STAs in the network increases from 40 to 60, indicating that the slot duration is adaptively adjusting to the network size with DRL.

5.2.2. Throughput Performance with Different Traffic Loads

In this section, we analyze the adaptive adjustment of RAW parameters obtained through the PPO method with different network loads. We assume a homogeneous traffic load of 0.05 Mbps for each STA. Therefore, the traffic load in the network increases as the number of STAs in the network increases. We observe the adaptive adjustments of RAW parameters and changes in network throughput as the number of STAs increases from 30 (10) to 90.

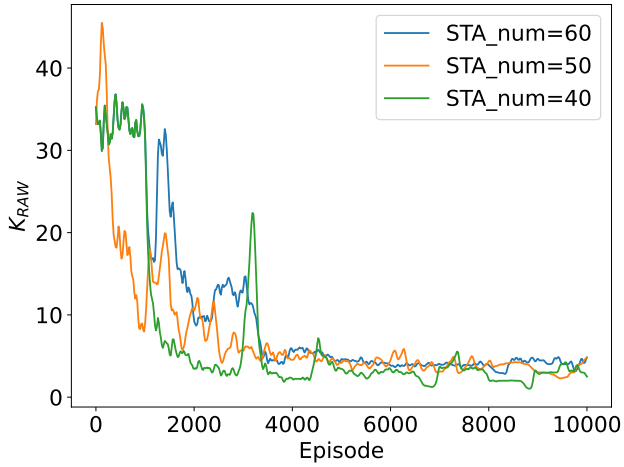

Figure 7. Convergence performance of K_{RAW}.

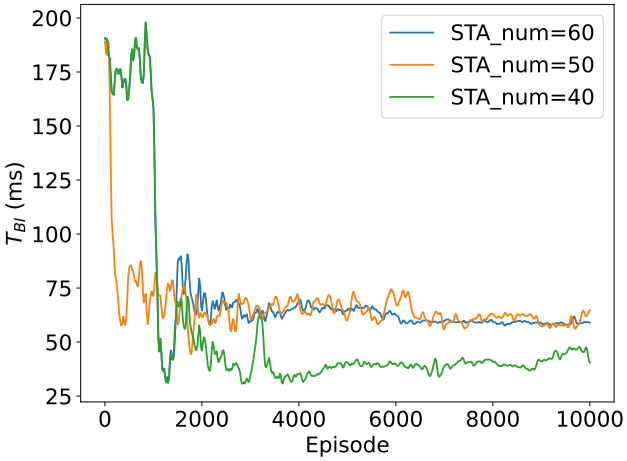

Figure 8. Convergence performance of T_{BI} (*w.r.t.* T_{slot}).

As shown in Figure 9, both the slot count and slot duration increase with the growing number of STAs and the traffic load in the network. The number of slots in a RAW increases stepwise with the network size and traffic load. Specifically, the slot count remains constant when the number of STAs is between 50–70 and 80–90. This is attributed to the fact that dividing fewer slots in a RAW significantly reduces contentions when the network size is small. Overall, the RAW mechanism ensures that the number of STAs in each RAW slot is not excessive. When the number of STAs in the network is less than 50, the slot duration increases significantly from 10 ms to about 50 ms, approximately 4 times longer, with the increasing network size and traffic load.

This trend is consistent with the changes in network throughput depicted in Figure 10. When the number of STAs in the network is less than 40, the network throughput remains unsaturated with few STAs and low traffic load in the network. As the number of STAs increases from 10 to 40, along with the ascending network traffic load, the network throughput obtained by PPO subsequently increases from 0.076 Mbps to 0.248 Mbps, approximately 4 times larger. At a certain point, with the number of STAs = 40, the network traffic load reaches its maximum capacity, leading to saturated network throughput under current network conditions. As the number of STAs in the network continues to increase from 40 to 90, contentions intensify, leading to a higher probability of transmission collisions. Meanwhile, when the network traffic load exceeds its capacity, constrained by the data transmission rate and the duration of a single slot, the AP cannot handle all the traffic from the STAs, resulting in a slight decrease by about 7% in network throughput. This trend

is consistent with the variation of throughput with the number of STAs in IEEE 802.11ah networks [39].

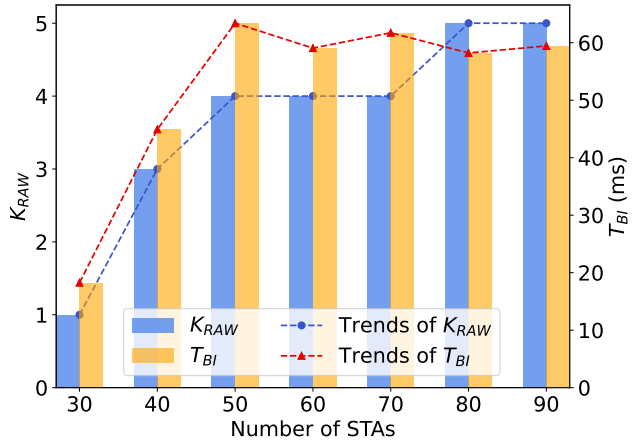

Figure 9. Adaptive adjustment of RAW parameters with varying network traffic loads.

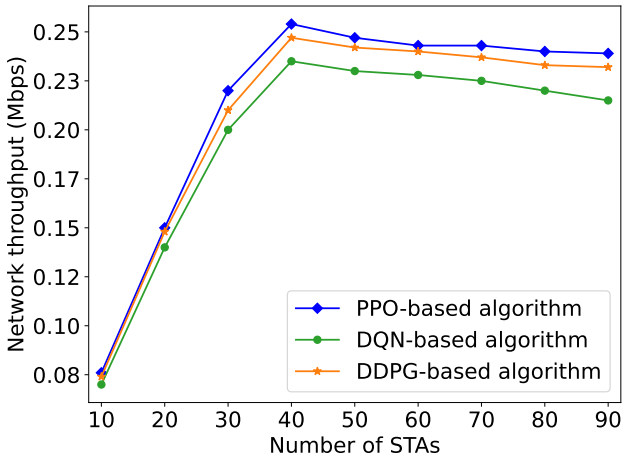

Figure 10. Network throughput obtained by the PPO-based algorithm with varying network traffic loads compared with the DQN-based and DDPG-based algorithms.

We have also compared the PPO-based algorithm with DQN-based and DDPG-based algorithms. As shown in Figure 10, when the number of STAs exceeds 40, the network throughput becomes saturated. Given 90 STAs, PPO obtains 11.2% and 3.1% higher network throughput compared to DQN and DDPG, respectively. Additionally, PPO and DDPG outperform DQN in small-size networks with periodic traffic. This is because DQN is designed for discrete actions, while PPO and DDPG are for continuous actions that perform better in high action spaces.

5.3. Learning Performance in Random Traffic Networks

To further validate the generalization ability of the proposed DRL algorithm, in this section, we modify the network conditions. While in the previous subsection, packets are transmitted at identical intervals, we now adjust the packet transmission intervals for each STA in the network.

The environment settings are shown in Table 3. In the random traffic network, the network size is set larger ($N_{max} \approx 300$) to emulate real-world network scales. The traffic load of each STA is random, following a normal distribution with mean μ and standard deviation σ, e.g., $\mu = 50$, $\sigma = 0.1$. As depicted in Figure 11, during data transmission the packet transmission interval of each STA follows a Poisson distribution with mean λ,

e.g., $\lambda = 100$. Additionally, we increase the maximum training iterations of the PPO agent to 20,000.

Table 3. Parameter settings in random traffic network.

Parameters	Settings
Wi-Fi channel configuration	MCS 0, 2 MHz
coverage radius	300 m
data rate	650 Kbit/s
traffic type	UDP
payload size	100 bytes
network size N	large, 150 (basic setting)
traffic load of the STAs	random
set of traffic loads $\boldsymbol{D}$	$d_n \sim \mathcal{N}(\mu, \sigma^2),\ n \in [1, N]$
packet transmission interval of the STAs	random
set of traffic intervals $\boldsymbol{I}$	$i_n \sim \pi(\lambda),\ n \in [1, N]$

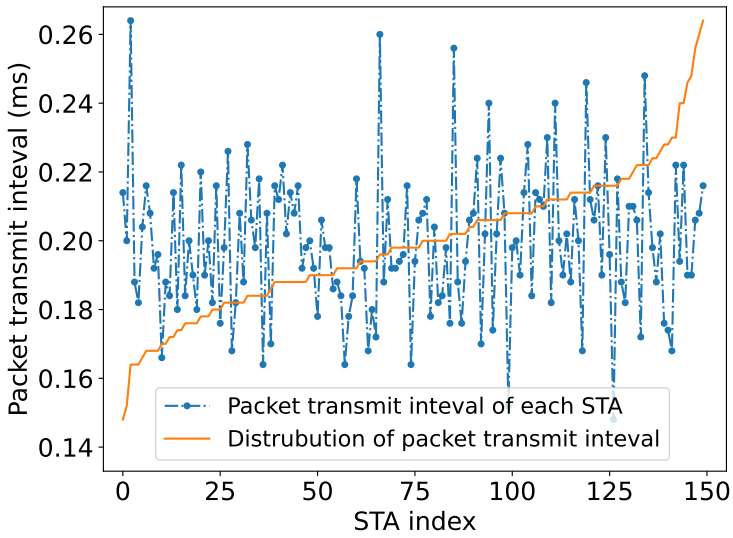

Figure 11. Illustration of the packet transmission interval distribution of STAs in random traffic networks.

5.3.1. Convergence to the Preferable RAW Parameters

We first validate the convergence performance of the PPO algorithm in the new network conditions. In the random traffic network implemented in NS-3, all the STAs transmit packets at random intervals following a Poisson distribution. Additionally, the network size is larger than that in the periodic traffic network, necessitating the division of STAs into more RAW groups. Therefore, the RAW parameters to be learned include RAW group count, slot count in one RAW, and slot duration in one RAW.

As shown in Figure 12, the normalized reward obtained by the PPO agent from interacting with the NS-3 simulation environment increases significantly as the training iterations progress, stabilizing at its maximum value after 17,000 training episodes. This indicates that the PPO agent can still learn the preferable RAW parameters and achieve optimized network throughput in the new network environment, i.e., the random traffic network. We also observe that as the network conditions become more complex, such as an increase in network size and random traffic arrivals, the PPO agent requires more interactions with the network simulation environment set up in NS-3. It needs to learn

the optimal action selection strategy over twice as many training iterations in the random traffic network compared to the periodic traffic network.

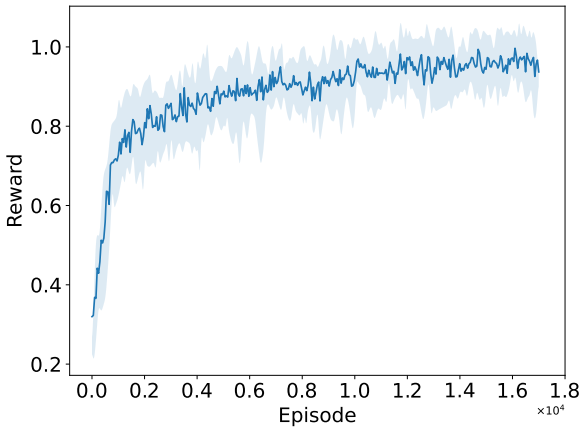

Figure 12. Convergence performance of the PPO algorithm in random traffic networks.

As depicted in Figures 13–15, when the number of STAs in the network is 150, both the number of RAW groups and the slot duration significantly increase compared to those in a small network size, while the number of slots remains small. This implies that the PPO agent tends to divide more RAW groups rather than more slots at this network size. The convergence performance to the preferable RAW parameters obtained by the PPO agent further demonstrates the generalization ability of the proposed DRL algorithm in complex networks environment with random traffic.

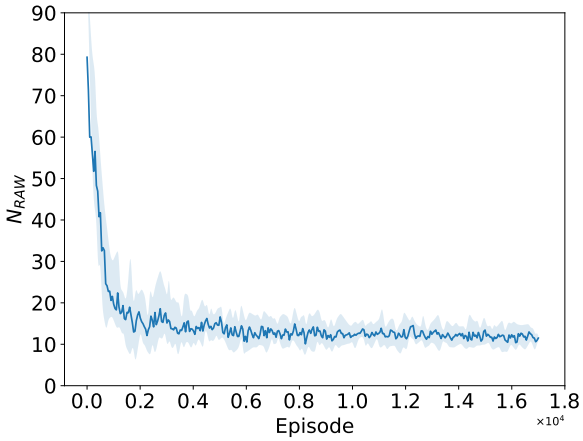

Figure 13. Convergence performance of RAW group count N_{RAW}.

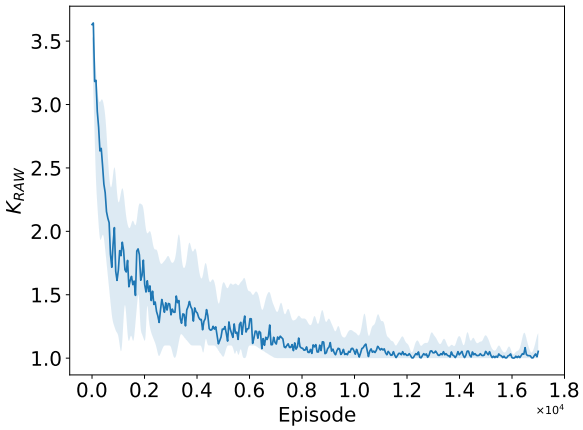

Figure 14. Convergence performance of slot count K_{RAW}.

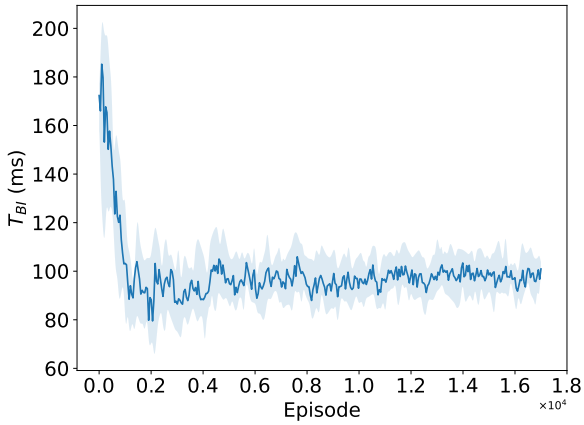

Figure 15. Convergence performance of BI duration T_{BI} *w.r.t. Tslot*.

5.3.2. Throughput Performance with Different Network Sizes

In this subsection, we validate and analyze the adaptive adjustment of RAW parameters and the network throughput obtained using the PPO algorithm under different network sizes, which is reflected by changes in the number of STAs. We increase the number of STAs from 150 to 300, a sufficiently large number to achieve saturated or oversaturated traffic load, which is suitable for validating the adjustment capability of RAW parameters and the network throughput of the proposed DRL framework. We observe the adaptive changes in the RAW parameters learned by the PPO agent and the network throughput obtained from the NS-3 simulation environment as the number of STAs increases. As shown in Figure 16, given that traffic load reaches saturation in large-scale networks, with the number of STAs in the network increasing from 150 to 300, the network throughput decreases by about 13%. It is evident that the increasing network size leads to intensified contention and collisions, thereby resulting in a significant decline in network throughput.

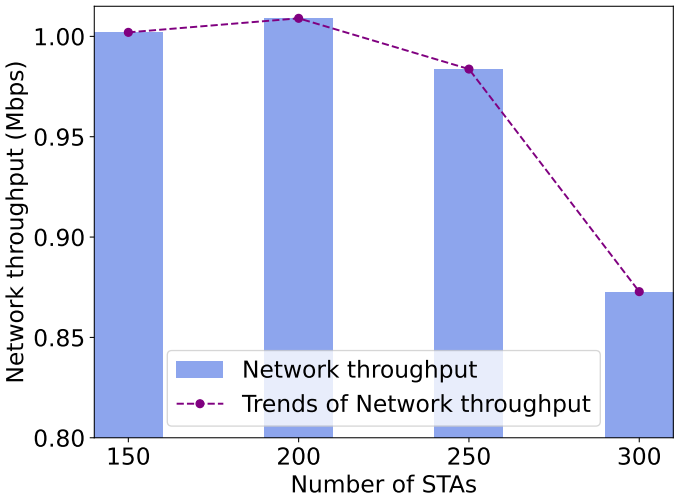

Figure 16. Network throughput obtained by the DRL algorithm with varying network sizes.

We also provide figures to show how the RAW parameters are adjusted to maintain certain network throughput in different network sizes with over-saturated traffic loads, emphasizing the importance of adjusting preferable RAW parameters to enhance network throughput with varying network sizes. As shown in Figure 17, we observe that when the number of STAs ranges from 150 to 200, the PPO agent tends to divide STAs into roughly 3 times more RAW groups. However, when the number of STAs increases to 250–300, the agent leans towards dividing more slots (from 1 to 5) in each RAW group. We analyze that within a certain range of network sizes, simply dividing RAW groups is capable of handling the current traffic load and mitigating contention. However, as the network size grows, it

becomes necessary to both divide RAW groups and more slots within each RAW group. The adaptive adjustment strategy learned by the PPO agent reduces the contention among STAs per slot, thus ensuring network throughput. Additionally, as the network size increases, the agent prefers to shorten the slot duration, consequently reducing the beacon interval duration by about 10%. We analyze that shortening the beacons broadcasting period allows the AP to schedule STAs more frequently for uplink data transmissions, thereby maintaining network throughput in intensified network conditions. We also notice that as the number of STAs increases from 60 to 150, the BI duration obtained by DRL increases from 60 ms to 100 ms, approximately by 66%, and the slot count increases significantly by about 3 times compared to the network size in the periodic traffic network. It is evident that as the network size scales up, contentions between STAs in the network intensify. To alleviate collisions and ensure network throughput, the PPO agent tends to dividing more slots, leading to an overall increase in the BI duration.

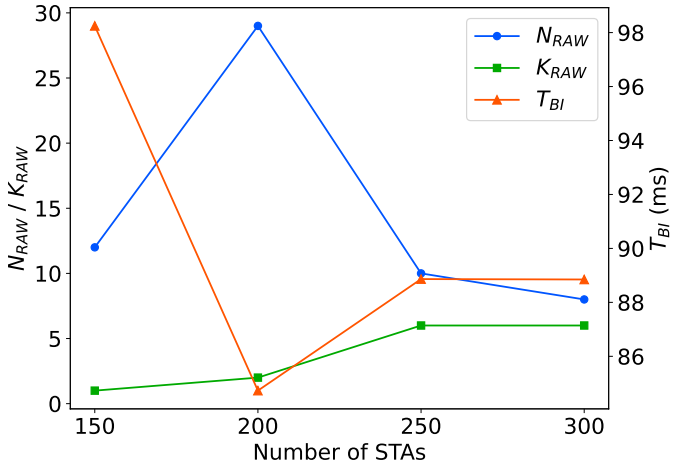

Figure 17. Adaptive adjustment of RAW parameters with varying network sizes.

5.4. Throughput Comparison of Different Slot Division Schemes

To further demonstrate the improvement in network performance achieved by the PPO-based algorithm, we compare the network throughput obtained from the PPO-based slot division scheme with two baseline slot division schemes. In the no-slot division scheme, all STAs contend for channel access in the same RAW group without slot division. Conversely, in the equal-slot division scheme, each STA is allocated one slot in every RAW group, ensuring non-contention-based access where only one STA can access the channel in a slot.

The overall BI durations are the same among different slot division schemes, as determined by the BI duration learned by the PPO agent. In this case, the slot durations vary among different schemes due to different slot division methods. As depicted in Figure 18, the throughput performance obtained from the NS-3 simulation environment with the RAW slot division scheme learned by the PPO agent significantly outperforms the two basic schemes, as the number of RAW slots and slot duration are adaptively adjusted according to the network size. In the worst case, where the number of STAs in the network is 300, the network throughput obtained from the PPO-based slot division scheme is still improved by about 80% and 1.3 times, respectively, compared to the two slot division schemes. This further illustrates the effective adjustments made by PPO in Figure 17, emphasizing that the optimization of RAW parameters can significantly improve network throughput, highlighting the necessity of RAW parameters optimization.

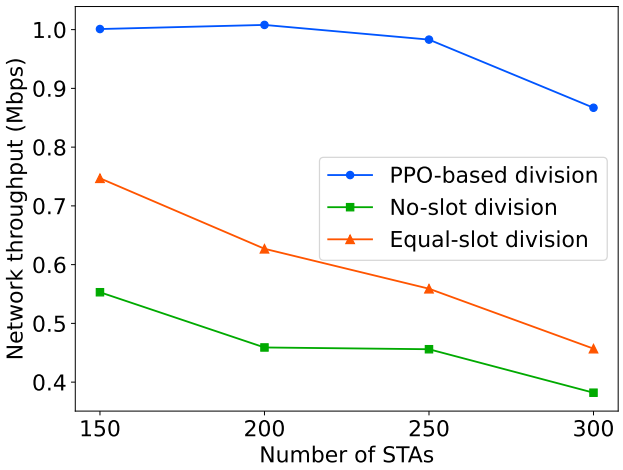

Figure 18. Comparison of network throughput between the PPO-based slot division scheme and the baseline slot division schemes.

It can be observed that as the network size increases, contentions and collisions among STAs in the network intensify, leading to a decrease in network throughput for all three slot division schemes. However, when the number of STAs in the network is between 150 and 250, the decrease in network throughput obtained from the PPO-based slot division scheme in the NS3 simulation environment is much smaller than that of the other two schemes. This indicates that the learning-based RAW slot division scheme can maintain better network throughput than the basic slot division schemes in deteriorating network conditions. It can be validated that the PPO agent can learn the preferable RAW parameters and effectively improve the network throughput, especially in scenarios with high contention. We also observe that as the network size increases, the network performance obtained by the division scheme that allocates one slot to each STA is significantly better than that of the scheme that does not divide slots within a RAW. This further validates the necessity of using the RAW grouping mechanism in large-scale networks and its improvement on network performance.

6. Conclusions

In this paper, we have proposed a PPO-based DRL algorithm for optimizing RAW parameters in the IEEE 802.11ah-based IoT network. Necessary analysis was first provided to emphasize the significant impact of RAW parameters on network throughput, and the RAW parameters optimization problem was formulated. A DRL framework interacting with the NS-3 simulator was then proposed, in which the optimization problem was reformulated as an MDP, and a PPO-based algorithm for RAW parameters optimization was proposed. In network environments with periodic and random traffic built in the NS-3 simulator, the performance of the proposed DRL algorithm was evaluated. The simulation results show that the PPO-based DRL scheme can adaptively adjust RAW parameters under different network conditions and achieve significantly improved network throughput compared to that of the baseline slot division schemes.

The proposed DRL and NS-3 simulation framework can be extended to different IEEE 802.11ah IoT network scenarios and optimization problems, such as the design of channel access mechanisms. Channel access optimization is particularly important in complex scenarios involving diverse traffic types, expanding network scales, and dynamic network topologies. In addition, complex network conditions hinder rapid environment reconstruction in NS-3, reducing learning and interaction efficiency in DRL. To reduce interaction overhead and enable real-time application, it is beneficial to develop a more accurate and comprehensive channel access analytical model. However, analytical models constructed solely relying on mathematical methods is limited when solving problems involving the joint optimization of multiple mechanisms. To address this limitation, a lightweight "surro-

gate" model can be constructed by collecting test data from real deployed IEEE 802.11ah IoT scenarios and fitting them using statistical methods and AI techniques. This model would be adaptable to diverse scenarios and capable of efficiently interacting with DRL algorithms. Moreover, in networks with time-varying network sizes and heterogeneous traffic, RAW grouping problems involve categorizing STAs into different RAW groups, requiring optimizing grouping strategies. In this paper, we have demonstrated the ability of the DRL approach to effectively determine the preferable RAW parameters across different network environments. Consequently, the DRL approach can be extended to address the challenge of finding the optimal RAW grouping strategy.

Author Contributions: Conceptualization, X.J., S.G., L.L. and B.G.; Methodology, X.J., S.G., C.D. and L.L.; Software, X.J. and C.D.; Validation, X.J., S.G., C.D., L.L. and B.G.; Investigation, X.J., S.G. and L.L.; Writing—original draft, X.J., S.G. and C.D.; Writing—review & editing, X.J., S.G., L.L. and B.G.; Supervision, S.G., L.L. and B.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62202506, Guangdong University Featured Innovation Program Project, grant number No. 2023KTSCX004, National Natural Science Foundation of China under Grant 62372488, and the Shenzhen Fundamental Research Program under Grant JCYJ20220818103201004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Vailshery, L.S. Internet of Things (IoT) and non-IoT Active Device Connections Worldwide from 2010 to 2025. 2022. Available online: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/ (accessed on 14 March 2024).
2. Wi-Fi Alliance. Wi-Fi CERTIFIED HaLow™: Wi-Fi® for IoT Applications (2021). 2021. Available online: https://www.wi-fi.org/file/wi-fi-certified-halow-wi-fi-for-iot-applications-2021 (accessed on 14 March 2024).
3. Wi-Fi Alliance. Wi-Fi CERTIFIED HaLow™ Technology Overview. 2021. Available online: https://www.wi-fi.org/file/wi-fi-certified-halow-technology-overview-2021 (accessed on 14 March 2024).
4. IEEE Standard for Information technology–Telecommunications and information exchange between systems—Local and metropolitan area networks–Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation. In *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as Amended by IEEE Std 802.11ai-2016)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–594. [CrossRef]
5. Tian, L.; Famaey, J.; Latré, S. Evaluation of the IEEE 802.11 ah restricted access window mechanism for dense IoT networks. In Proceedings of the 2016 IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, Coimbra, Portugal, 21–24 June 2016; pp. 1–9.
6. Taramit, H.; Barbosa, L.O.; Haqiq, A. Energy efficiency framework for time-limited contention in the IEEE 802.11 ah standard. In Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps), IEEE, Madrid, Spain, 7–11 December 2021; pp. 1–6.
7. Wang, Y.; Chai, K.K.; Chen, Y.; Schormans, J.; Loo, J. Energy-delay aware restricted access window with novel retransmission for IEEE 802.11 ah networks. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), IEEE, Washington, DC, USA, 4–8 December 2016; pp. 1–6.
8. Seferagić, A.; De Poorter, E.; Hoebeke, J. Enabling wireless closed loop communication: Optimal scheduling over IEEE 802.11 ah networks. *IEEE Access* **2021**, *9*, 9084–9100. [CrossRef]
9. Lakshmi, L.R.; Sikdar, B. Achieving fairness in IEEE 802.11 ah networks for IoT applications with different requirements. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), IEEE, Shanghai, China, 20–24 May 2019; pp. 1–6.
10. Mahesh, M.; Harigovindan, V. Throughput enhancement of IEEE 802.11 ah raw mechanism using ANN. In Proceedings of the 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), IEEE, Kurukshetra, India, 24–26 June 2020; pp. 1–4.
11. Oliveira, E.C.; Soares, S.M.; Carvalho, M.M. K-Means Based Grouping of Stations with Dynamic AID Assignment in IEEE 802.11 ah Networks. In Proceedings of the 2022 18th International Conference on Mobility, Sensing and Networking (MSN), IEEE, Guangzhou, China, 14–16 December 2022; pp. 134–141.

12. Yan, M.; Xiong, R.; Wang, Y.; Li, C. Edge Computing Task Offloading Optimization for a UAV-Assisted Internet of Vehicles via Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 5647–5658. [CrossRef]
13. Yan, M.; Zhang, L.; Jiang, W.; Chan, C.A.; Gygax, A.F.; Nirmalathas, A. Energy Consumption Modeling and Optimization of UAV-Assisted MEC Networks Using Deep Reinforcement Learning. *IEEE Sens. J.* **2024**, *24*, 13629–13639. [CrossRef]
14. Bianchi, G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Sel. Areas Commun.* **2000**, *18*, 535–547. [CrossRef]
15. Soares, S.M.; Carvalho, M.M. Throughput analytical modeling of IEEE 802.11 ah wireless networks. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, Las Vegas, NV, USA, 11–14 January 2019; pp. 1–4.
16. Zheng, L.; Ni, M.; Cai, L.; Pan, J.; Ghosh, C.; Doppler, K. Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 6180–6192. [CrossRef]
17. Sangeetha, U.; Babu, A. Performance analysis of IEEE 802.11 ah wireless local area network under the restricted access window-based mechanism. *Int. J. Commun. Syst.* **2019**, *32*, e3888.
18. Taramit, H.; Camacho-Escoto, J.J.; Gomez, J.; Orozco-Barbosa, L.; Haqiq, A. Accurate analytical model and evaluation of Wi-Fi HaLow based IoT networks under a Rayleigh-fading channel with capture. *Mathematics* **2022**, *10*, 952. [CrossRef]
19. Zhao, Y.; Yilmaz, O.N.; Larmo, A. Optimizing M2M energy efficiency in IEEE 802.11 ah. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), IEEE, San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
20. Nawaz, N.; Hafeez, M.; Zaidi, S.A.R.; McLernon, D.C.; Ghogho, M. Throughput enhancement of restricted access window for uniform grouping scheme in IEEE 802.11 ah. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), IEEE, Paris, France, 21–25 May 2017; pp. 1–7.
21. Tian, L.; Khorov, E.; Latré, S.; Famaey, J. Real-time station grouping under dynamic traffic for IEEE 802.11 ah. *Sensors* **2017**, *17*, 1559. [CrossRef] [PubMed]
22. Tian, L.; Santi, S.; Latré, S.; Famaey, J. Accurate sensor traffic estimation for station grouping in highly dense IEEE 802.11 ah networks. In Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems, Delft, The Netherlands, 5 November 2017; pp. 1–9.
23. Khorov, E.; Lyakhov, A.; Nasedkin, I.; Yusupov, R.; Famaey, J.; Akyildiz, I.F. Fast and reliable alert delivery in mission-critical Wi-Fi HaLow sensor networks. *IEEE Access* **2020**, *8*, 14302–14313. [CrossRef]
24. Ahmed, N.; De, D.; Hussain, M.I. A QoS-aware MAC protocol for IEEE 802.11 ah-based Internet of Things. In Proceedings of the 2018 Fifteenth International Conference on Wireless and Optical Communications Networks (WOCN), IEEE, Kolkata, India, 2–4 February 2018; pp. 1–5.
25. Tian, L.; Mehari, M.; Santi, S.; Latré, S.; De Poorter, E.; Famaey, J. IEEE 802.11 ah restricted access window surrogate model for real-time station grouping. In Proceedings of the 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), IEEE, Chania, Greece, 12–15 June 2018; pp. 14–22.
26. Hasi, M.A.A.; Haque, M.D.; Siddik, M.A. Traffic Demand-based Grouping for Fairness among the RAW Groups of Heterogeneous Stations in IEEE802. 11ah IoT Networks. In Proceedings of the 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), IEEE, Gazipur, Bangladesh, 24–26 February 2022; pp. 1–6.
27. Chang, T.C.; Lin, C.H.; Lin, K.C.J.; Chen, W.T. Traffic-aware sensor grouping for IEEE 802.11 ah networks: Regression based analysis and design. *IEEE Trans. Mob. Comput.* **2018**, *18*, 674–687. [CrossRef]
28. Garcia-Villegas, E.; Lopez-Garcia, A.; Lopez-Aguilera, E. Genetic algorithm-based grouping strategy for IEEE 802.11 ah networks. *Sensors* **2023**, *23*, 862. [CrossRef] [PubMed]
29. Tian, L.; Lopez-Aguilera, E.; Garcia-Villegas, E.; Mehari, M.T.; De Poorter, E.; Latré, S.; Famaey, J. Optimization-oriented RAW modeling of IEEE 802.11 ah heterogeneous networks. *IEEE Internet Things J.* **2019**, *6*, 10597–10609. [CrossRef]
30. Bobba, T.S.; Bojanapally, V.S. Fair and Dynamic Channel Grouping Scheme for IEEE 802.11 ah Networks. In Proceedings of the 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT), IEEE, Shah Alam, Malaysia, 9–11 November 2020; pp. 105–110.
31. Mahesh, M.; Pavan, B.S.; Harigovindan, V. Data rate-based grouping using machine learning to improve the aggregate throughput of IEEE 802.11 ah multi-rate IoT networks. In Proceedings of the 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), IEEE, New Delhi, India, 14–17 December 2020; pp. 1–5.
32. Ibrahim, A.; Hafez, A. Adaptive IEEE 802.11 ah MAC protocol for Optimization Collision Probability in IoT smart city data traffic Based Machine Learning models. **2023**, *preprint*. [CrossRef]
33. Pavan, B.S.; Harigovindan, V. GRU based optimal restricted access window mechanism for enhancing the performance of IEEE 802.11 ah based IoT networks. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 16653–16665. [CrossRef]
34. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
35. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS '99), NeurIPS, Denver, CO, USA, 29 November–4 December 1999; pp. 1057–1063.
36. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.

37. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 1889–1897.
38. Kim, Y.; Hwang, G.; Um, J.; Yoo, S.; Jung, H.; Park, S. Throughput performance optimization of super dense wireless networks with the renewal access protocol. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 3440–3452. [CrossRef]
39. Tian, L.; Deronne, S.; Latré, S.; Famaey, J. Implementation and Validation of an IEEE 802.11ah Module for ns-3. In Proceedings of the 2016 Workshop on Ns-3 (WNS3 '16), Seattle, WA, USA, 15–16 June 2016; pp. 49–56.