Traffic-driven Fast RAW Grouping in Wi-Fi HaLow Heterogeneous Network

Zirui Li*†, Chengyi Deng*†, Yusi Long*†, Shimin Gong*†
*School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, China
†Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology, China

2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring) | 979-8-3315-3147-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/VTC2025-SPRING65109.2025.11174734

***Abstract*—In this paper, we consider a large-scale Wi-Fi HaLow heterogeneous network in a real-world Internet of Things (IoT) environment, where numerous devices are distributed around an access point (AP). These devices collect and transmit data using the restricted access window (RAW) mechanism. They exhibit varying traffic characteristics. Moreover, dependencies among the devices often exist. We aim to maximize the overall throughput by adjusting the RAW grouping decision. The heterogeneity of real network data and dependencies between devices make the RAW grouping process more complicated. To overcome this challenge, we propose a novel traffic-driven RAW grouping approach. Specifically, we build a simulation environment based on real IoT data and NS-3. Then, we analyze the traffic characteristics of each IoT device. This analysis allows us to fully explore the dependencies and cooperation relationships among these devices. Hence, we can aggregate devices with these relationships into clusters. Each cluster is then treated as a supernode which is used as a basic unit for RAW grouping. Then, we use proximal policy optimization (PPO) algorithm to optimize the RAW grouping process via interacting with the environment. Numerical results indicate that the proposed traffic-driven algorithm significantly achieves faster convergence and improves grouping efficiency in large-scale heterogeneous networks compared to baselines.**

***Index Terms*—Wi-Fi HaLow, RAW grouping, clustering, deep reinforcement learning**

I. Introduction

Internet of Things (IoT) devices have been widely integrated into electronic products such as televisions, smartphones, and smart home systems. These devices generate substantial amounts of heterogeneous network traffic, either periodically or in response to specific events. The massive network traffic produced by numerous devices leads to intense channel contention, which poses significant challenges to network protocols [1]. The IEEE 802.11ah protocol (Wi-Fi HaLow) has been specifically designed for the IoT network [2]. Its physical and media access control (MAC) layers meet the requirements of extended range and large-scale connectivity in IoT networks. Wi-Fi HaLow operates in a frequency band below 1 GHz, offering low power consumption and supporting a variety of modulation and coding schemes (MCS). It facilitates data rates from 150 Kbps to 78 Mbps, with a transmission range of up to 1 kilometer. Furthermore, Wi-Fi HaLow can connect up to 8191 devices to a single access point (AP). Wi-Fi HaLow implements a mechanism known as the Restricted Access Window (RAW) to reduce channel contention and transmission interference in environments with dense device deployments [3]. This mechanism utilizes carrier sense multiple access with collision avoidance (CSMA/CA) to organize devices into groups, allowing each RAW group to transmit data during designated times within the beacon interval (BI). The RAW mechanism enables the devices to flexibly configure the RAW parameters. Some studies proposed modeling the channel contention process of devices in RAW grouping as a Markov chain [4]. Some studies proposed to use graph neural network methods to represent device interactions and make RAW grouping decisions [5], [6].

This work was supported in part by National Natural Science Foundation of China under Grant 62372488, the Shenzhen Fundamental Research Program under Grant JCYJ20220818103201004. (Corresponding author: Shimin Gong)

However, these methods typically operate under the assumption that all devices have similar traffic characteristics and task requirements, which overlooks the heterogeneity of real-world IoT devices. These devices often exhibit diverse data generation behaviors and operational characteristics. Specifically, some generate periodic sensing data with predictable traffic, while others produce event-driven data with bursty and unpredictable behavior. The authors in [7] showed that the data of IoT devices exhibit significant variation in packet sizes and transmission intervals, not only among different brands and types of devices but also within the same device. The variation in traffic patterns makes RAW grouping more complex and challenging. Existing RAW grouping methods fail to fully capture such heterogeneity. The authors in [8] determined RAW grouping by estimating the next transmission interval of devices. However, they assumed that devices are homogeneous. It means that the devices in [8] have identical packet sizes and predictable dynamic transmission frequencies. The authors in [9] derived the expected channel time required for a group of sensors to meet their traffic demands and formulated an optimization problem to maximize the worst-case channel utilization across groups. However, the data in these devices cannot change dynamically. The above studies only consider partial heterogeneity in device traffic, assuming ideal conditions. The data in real-world IoT environments are heterogeneous, making the implementation or deployment of RAW grouping methods difficult or suboptimal.

To address the challenges posed by device heterogeneity and diverse traffic patterns in IoT networks, we first introduce real IoT data into the simulation environment to model the traffic patterns of devices in various application scenarios. Specifically, we extend the NS-3 simulation environment of Wi-Fi HaLow network [10], allowing each device to define its packet size, transmission time, and other functions. By combining real data, we can accurately evaluate the impact of different devices and traffic patterns on the RAW grouping strategy. Then we propose a traffic-driven method by thoroughly analyzing the traffic characteristics of devices before RAW grouping. Specifically, we extract the traffic characteristics of devices and fully explore the dependencies and collaborations between the devices. Based on these relationships, we group related devices into supernodes. In the subsequent RAW grouping, the supernodes are used as units to make grouping more efficient. We use deep reinforcement learning (DRL) to optimize the grouping process. Our approach aims to maximize network throughput by leveraging the interaction between DRL and the simulation environment to continuously adjust the RAW grouping strategy. Specifically, the NS-3 evaluates the network throughput under different RAW groupings and provides it to the DRL as a reward. Then DRL agent continuously adjusts the grouping strategy based on the reward to improve network performance. Experimental results show that the proposed traffic-driven RAW grouping method is significantly better than the conventional method in terms of grouping efficiency. Compared with the baselines, the proposed method can efficiently capture the dependencies and collaborations between devices and achieve more efficient grouping efficiency in a real IoT environment.

II. System Model

We consider a single AP in a randomly accessed Wi-Fi HaLow network that is connected to multiple devices. These devices generate a significant amount of heterogeneous traffic based on their specific task requirements. Different devices exhibit heterogeneous traffic patterns, with some actively transmitting information frequently, while others may remain silent for extended periods. Moreover, the amount of transmitted packet sent by the devices varies significantly, which ranges from small packets containing only a few bytes to larger ones with substantial amounts of data. We assume that all devices and the AP operate on the same channel, which allows for adjustable MCS for each device. Each device follows the CSMA/CA mechanism to access the channel and transmit its packets, which helps to avoid collisions with other devices' transmissions. Specifically, each device first checks whether other devices are transmitting on the channel. If the channel is busy, the device continues to listen until it detects that the channel is idle. Then it performs a random backoff to minimize the chances of colliding with other devices that are also waiting to transmit. If some devices fail to detect others' transmissions, they may transmit simultaneously, which causes interference. When a packet transmission fails, no successful acknowledgment is received. The device will retry transmitting the same packet until it receives an acknowledgment or reaches the maximum number of retry attempts.

A. Traffic-driven Clustering

Due to the heterogeneity of traffic characteristics among devices, directly optimizing RAW grouping can lead to inefficient resource allocation and increased complexity. Through the analysis of real IoT datasets, we observed that devices

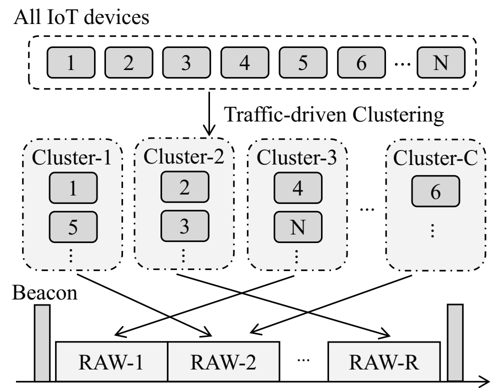

Fig. 1: Clustering-based RAW Grouping.

with dependency and collaboration relationships often exhibit similar traffic patterns. To enhance the efficiency of RAW grouping and reduce system complexity, we first cluster the devices with dependency and collaboration relationships based on their traffic characteristics. The traffic characteristics of each device include the overall characteristics of the data flow, or can be specific to the data packet size, data transmission interval, etc. Here we analyze the data packet size, data transmission interval and whether the data flow is bursty. We have recorded the historical data generated by each device in the network model over a period of time, including the packet size and transmission timestamps for each device. Let H_i denote the transmission record of device i within a time period as follows:

$$H_i = \{(p_{i,1}, t_{i,1}), (p_{i,2}, t_{i,2}), \ldots, (p_{i,n_i}, t_{i,n_i})\}^T, \quad (1)$$

where $p_{i,j}$ and $t_{i,j}$ denote the packet size and the timestamp of the j-th packet of device i, respectively. The factor n_i indicates the total number of packets transmitted by device i. Then, we conduct traffic feature analysis for all devices using the historical data. We extract three types of features: periodic features, packet features, and event-driven features.

We first calculate the time intervals between the data packets transmitted by each device to extract the temporal features of the device's traffic. The transmission interval of i-th device between two adjacent packets is denoted as $\Delta t_{i,j} = t_{i,j} - t_{i,j-1}$. The periodicity features of device i, i.e., mean $\mu_{\Delta t_i}$ and variance $\sigma_{\Delta t_i}$ of $\Delta t_{i,j}$ are denoted as follows:

$$\mu_{\Delta t_i} = \frac{1}{n_i - 1} \sum_{j=2}^{n_i} \Delta t_{i,j}, \quad (2a)$$

$$\sigma_{\Delta t_i} = \sqrt{\frac{1}{n_i - 1} \sum_{j=2}^{n_i} (\Delta t_{i,j} - \mu_{\Delta t_i})^2}. \quad (2b)$$

The packet size reflects the intensity of the device's traffic. Similarly, its features for device i are computed as follows:

$$\mu_{p_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} p_{i,j}, \quad (3a)$$

$$\sigma_{p_i} = \sqrt{\frac{1}{n_i} \sum_{j=1}^{n_i} (p_{i,j} - \mu_{p_i})^2}. \quad (3b)$$

We also aim to analyze whether a device is event-driven, i.e., whether the device belongs to the category that generates traffic with high burstiness due to certain random events. We introduce the sliding window method to analyze the time intervals. We define a window of device i as follows:

$$\mathcal{W}_{i,k} = \{\Delta t_{i,j} \mid j = k, k+1, \ldots, k+W-1\}, \tag{4}$$

where $k \in \{1, 2, \ldots, n_i - W + 1\}$ represents the starting position of the window and W represents the window size. For $\forall i, k$, we use E_i to represent whether device i exhibits event-driven characteristics as follows:

$$E_i = \begin{cases} 1, & \text{if} \max_k \dfrac{\lambda_{i,k}}{W} > \alpha, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $\lambda_{i,k}$ denotes the number of packet intervals in $\mathcal{W}_{i,k}$ that are smaller than a threshold. Thus, $\lambda_{i,k}$ quantifies the traffic density within the sliding window. If this ratio exceeds α, the traffic in the given time window is considered bursty. When burstiness is detected, the device is classified as exhibiting event-driven characteristics. After extracting the traffic features of the devices, we classify them based on similar traffic patterns. We denote the set of cluster-k as $\mathcal{C}_k$. The traffic features of a device is represented as follows:

$$f_i = \mathbb{F}(H_i) = [\mu_{\Delta t_i}, \sigma_{\Delta t_i}, \mu_{p_i}, \sigma_{p_i}, E_i], \tag{6}$$

where $f_i \in \mathcal{C}_k$. We utilize K-means for device classification. The goal of the K-means algorithm is to partition the devices into C clusters while minimizing the squared Euclidean distance between the feature vectors of the devices. This objective is expressed as follows:

$$\min_C \sum_{i=1}^{I} \sum_{k=1}^{C} \|f_i - c_k\|^2, \quad \text{s.t. } (1) - (6), \tag{7}$$

where c_k is the centroid of k-th cluster, and I is the number of devices. Then We conduct RAW grouping for each cluster.

B. Clustering-based RAW Grouping

As depicted in Fig.1, we first extract and analyze the device traffic features and then aggregate the devices into clusters. Then, we perform RAW grouping for each cluster utilizing the RAW mechanism. We assume that the devices are divided into R RAW groups. The transmission time is divided into T periodic transmission cycles, where $t = 1, 2, , T$. The RAW transmission resources within each group are allocated to the devices in the clusters belonging to that group. Let $u_k(t)$ be the throughput of k-th cluster during the t-th transmission cycle. Then, the averaged network throughput is expressed as follows:

$$U = \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{C} u_k(t). \tag{8}$$

We maximize network performance by optimizing the RAW grouping $\boldsymbol{z}$ within the network clustering state $\boldsymbol{c}$ as follows:

$$\max_{\boldsymbol{z}} \mathbb{E}[U|\boldsymbol{c}, \boldsymbol{z}], \quad \text{s.t.}(7) - (8). \tag{9}$$

Deriving system throughput from RAW grouping (9) presents a significant challenge. This complexity arises from the difficulty of mapping the current clustering results $\boldsymbol{c}$ and RAW grouping decision $\boldsymbol{z}$ to the device throughput U, while also making optimal grouping decisions. In simpler terms, obtaining a closed-form solution for device throughput through mathematical computation is not practical. To make decisions for RAW grouping, we implement the following architecture.

III. Traffic-driven DRL for RAW Grouping

To tackle this issue, we propose a method based on DRL. This method interacts with the NS-3 simulation system to learn how to make effective grouping decisions based on the current network state. Within this framework, the algorithm engages with the simulation system to receive feedback on throughput after each grouping decision, allowing it to gradually adjust its strategy and enhance overall system performance. We evaluate network performance using the NS-3 simulator, which demonstrates high accuracy in wireless network simulations. The work presented in [10], [11] implements the Wi-Fi HaLow protocol within the NS-3 simulator. In real-world IoT networks, the packet sizes of different devices are not always uniform, even the same device may transmit packets of varying sizes under different conditions. Additionally, the transmission times of devices are not constant. However, existing simulation systems typically support fixed packet sizes with fixed intervals, which does not reflect the real conditions of IoT networks. In our work, we enhance the NS-3 simulation system for the HaLow protocol, allowing users to freely define the transmission time and packet size of devices in the network. This improvement makes the simulation more realistic and better aligned with actual IoT scenarios. After analyzing the historical data, we randomly select a time window of data based on the device's traffic characteristics. This data represents the device's behavior within a specific time period and aligns with its traffic features. Subsequently, we embed this data into the device nodes within the NS-3 network simulation environment, generating additional device nodes. To ensure that the device behavior in the simulation environment matches the traffic patterns of real devices, we strictly adhere to the specific data characteristics of the devices when embedding the data.

In the DRL framework, the need for an explicit throughput function is eliminated, which significantly enhancing the flexibility and adaptability of the algorithm. In traditional network optimization methods, accurately modeling the relationship between network states and throughput is often necessary, which usually involves complex mathematical derivations and prior assumptions. In contrast, the DRL framework allows the agent to interact with the environment and adjust its strategy based on the rewards it receives after each decision. It means that the agent progressively optimizes its policy through continuous trial and feedback, without needing to define the relationship between throughput and states in advance. This reward-based learning approach enables DRL to effectively manage highly dynamic and complex network environments. As a result, it can

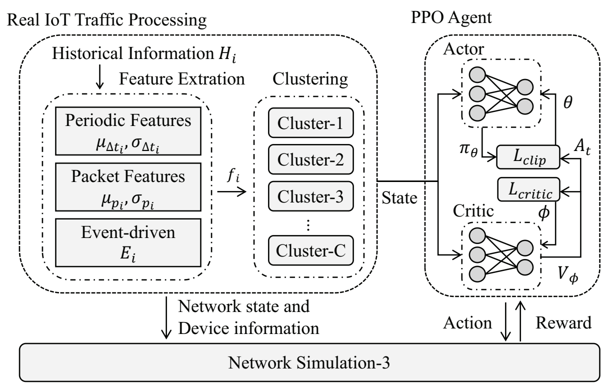

Fig. 2: Traffic-driven DRL for RAW grouping.

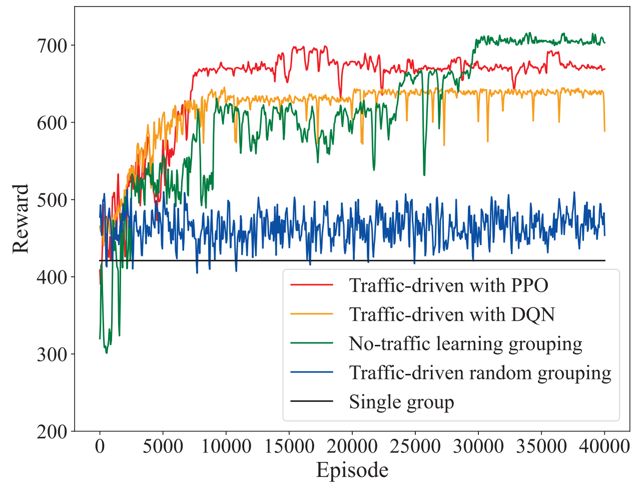

Fig. 3: The comparison of reward.

discover the optimal decision strategy through learning, even when precise modeling is challenging.

We utilize the Proximal Policy Optimization (PPO) algorithm, which improves the stability of the learning process by constraining the size of policy updates. This feature makes PPO especially suitable for tackling high-dimensional control problems in complex environments. Additionally, PPO employs action sampling to effectively explore the parameter space, which helps avoid premature convergence to local optima. We utilize the markov decision process to model the interaction between the agent and the NS-3 in our approach. This interaction is mathematically characterized by a tuple consisting of state space S, action space A, reward function R, and discount factor γ, where R guides the agent's exploration, and γ is used to balance immediate and long-term rewards. For our specific problem (9), the state S refers to the current network configuration, while the action A corresponds to the current RPS. Throughout the transmission process, the states and actions are represented as $s = [s_1, s_2, \ldots, s_T]$ and $a = [a_1, a_2, \ldots, a_T]$. Here, s_i describes the network device configuration during the i-th transmission period, and a_i represents the RPS during that period. In our framework, the reward function can be constructed based on the feedback from NS-3.

PPO enhances the sample efficiency of Advantage Actor-Critic by incorporating importance sampling. Additionally, PPO imposes a constraint on the magnitude of parameter updates to ensure the stability of the policy gradient. The policy network $\pi_\theta(a|s)$ with parameter θ outputs an action a_i based on the input state s_i. The optimization objective is to maximize the expected cumulative reward, which means maximizing the value function of the policy: $J(\theta) = \mathbb{E}_{s_t \sim \pi_\theta, a_t \sim \pi_\theta}\left[Q^\pi(s_t, a_t)\right]$. Let the ratio of updated policy to the previous policy be denoted as r_t. The optimized clipped surrogate objective function of PPO can be expressed as:

$$L_{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right],$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, and $\text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)$ function is used to restrict r_t to a range $[1-\epsilon, 1+\epsilon]$. The hyperparameter ϵ defines the limits for this clipping operation. This ensures that the ratio between the new and old policies stays within the interval $[1-\epsilon, 1+\epsilon]$, preventing excessive differences between the two policies.A_t is the advantage function.

The Critic network is used to estimate the state value function, denoted as $V_\phi(s)$, which represents the expected return for a given state. It evaluates the actions chosen by the Actor by calculating the value of state. The output of the Critic network is the estimated value V for state s. The model is optimized by minimizing the error $L_{\text{critic}}(\phi) = \mathbb{E}_t\left[(V_\phi(s_t) - R_t)^2\right]$ between the estimated value and target value.

IV. Simulation

In this part, we consider a large-scale Wi-Fi HaLow heterogeneous network, where numerous devices executing various tasks are distributed within a 1 km radius around AP. The beacon interval and the channel bandwidth of the AP are set to TBI = 102400 μs and 2 MHz. We conducted feature analysis and processing on the real device data captured from the CIC IoT Dataset 2022 provided by the Canadian Institute for Cybersecurity [7]. This dataset includes data transmission histories from 60 different brands and categories of IoT devices, collected within a lab environment over three months. The data gathering was divided into four phases for each device: booting, idle, active and interaction. In addition to the data captured in a controlled lab environment, they also recorded device traffic throughout the day in a more flexible setting. During the experiment, personnel were permitted to enter the lab at any time, resulting in devices either passively or actively interacting with users. This interaction generated network traffic that closely resembles real-life usage scenarios. During clustering process, we control the number of devices in each cluster to $[3, 7]$ to prevent too many devices from being aggregated together and ensure the ultimate performance.

In Fig. 3, we compare the throughput performance of the proposed Traffic-driven with PPO algorithm with four baselines: No-traffic learning grouping, Traffic-driven random grouping, Single group schemes, and a deep Q-network (DQN) implementation under the same grouping framework, where the proposed method demonstrates superior final performance and stability. The traffic-driven Random grouping scheme means

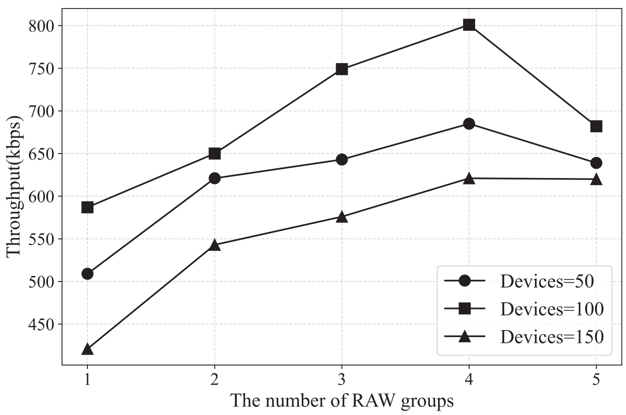

Fig. 4: The performance of Traffic-driven DRL with different numbers of devices and RAW groups.

that devices are first clustered based on traffic analysis, and the resulting clusters are then randomly assigned to groups for further processing. Single group scheme illustrates that all devices transmit in a single time slot without any clustering or grouping. Utilizing cluster optimization, the proposed method decreases the action space in model-free PPO, resulting in quicker convergence with a 70% reduction in convergence time and only a 4% performance gap. The enhanced stability of policy updates in PPO, enabled by its proximal policy clipping mechanism, allows PPO to achieve superior final performance and robustness compared to other reinforcement learning algorithms. This advantage effectively mitigates the drastic policy fluctuations in DQN which stem from Q-value estimation errors. In the Single group scheme, all devices transmit in the same time slot without any grouping optimization. This means that the probability of collisions and interference within each time slot remains constant, regardless of changes in network load. The competition among devices is random and uncoordinated, leading to relatively stable throughput, which is represented as a flat line. This behavior indicates that, without grouping optimization, the system's performance is constrained by collisions and interference, preventing any significant improvement in throughput. Therefore, the throughput of the Single group scheme can serve as a lower bound, which reflects the worst-case performance.

Fig. 4 illustrates the relationship between the number of RAW groups and network throughput under different device quantities. Specifically, as the number of RAW groups increases, the throughput gradually improves because the channel contention is reduced. However, the throughput decreases as the number of groups reaches 5. As an excessive number of groups results in overly short time slots for each RAW group, which underutilizes the available channel resources. Regarding the device quantity, throughput increases with the number of devices. In particular, the throughput with 100 devices is higher than that with 50 devices, because, although the number of data packets increases with more devices, the more efficient use of channel resources results in improved throughput. However, the throughput with 150 devices is significantly lower than with 50 and 100 devices, primarily because the network channels are approaching their capacity limit, leading to severe channel contention and a subsequent reduction in throughput.

V. CONCLUSION

In this paper, we have presented a novel traffic-driven RAW grouping approach to optimize the RAW mechanism in large-scale Wi-Fi HaLow heterogeneous networks. By analyzing the traffic characteristics of IoT devices, we fully explore the dependency and cooperation among devices, and cluster devices based on this relationship. Then we treat each cluster as a supernode and utilize PPO to dynamically adjust the RAW grouping decisions for supernodes. Through this method, we effectively solve the difficulties of RAW grouping caused by device heterogeneity in large-scale IoT. The simulation results demonstrate that our approach substantially improves the grouping efficiency compared to conventional methods. The proposed solution offers a practical and scalable optimization framework for real-world IoT environments, where diverse device behaviors and different traffic patterns are prevalent. Future work will focus on addressing the challenge of long training times caused by frequent interactions with NS-3.

REFERENCES

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6g internet of things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Aug. 2022.

[2] IEEE, "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," *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1–594, Oct. 2017, doi:10.1109/IEEESTD.2017.7920364.

[3] X. Xiao, X. Wang, and W. Lin, "Wi-fi halow for the internet of things: An up-to-date survey on ieee 802.11ah research," *J. Netw. Comput. Appl.*, vol. 182, p. 103036, May 2021.

[4] E. Khorov, A. Lyakhov, and R. Yusupov, "Two-slot based model of the ieee 802.11ah restricted access window with enabled transmissions crossing slot boundaries," in *Proc. WoWMoM 2018*, Chania, Greece, Jun. 2018, pp. 1–9.

[5] Z. Gu, B. Vucetic, K. Chikkam, P. Aliberti, and W. Hardjawana, "Graph representation learning for contention and interference management in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2479–2494, Jan. 2024.

[6] Q. Chen, "An energy-efficient channel access with target wake time scheduling for overlapping 802.11ax basic service sets," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18 973–18 986, Mar. 2022.

[7] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multidimensional iot profiling dataset," in *Proc. PST*, Fredericton, Canada, Aug. 2022, pp. 1–11.

[8] L. Tian, E. Khorov, S. Latr, and J. Famaey, "Real-time station grouping under dynamic traffic for ieee 802.11ah," *Sensors*, vol. 17, no. 7, p. 1559, Jul. 2017.

[9] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, and W.-T. Chen, "Traffic-aware sensor grouping for ieee 802.11ah networks: Regression based analysis and design," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 674–687, May 2019.

[10] L. Tian, A. Šljivo, S. Santi, E. De Poorter, J. Hoebeke, and J. Famaey, "Extension of the ieee 802.11ah ns-3 simulation module," in *Proc. WNS3*, Surathkal, India, Jun. 2018, p. 53C60, doi:10.1145/3199902.3199906.

[11] L. Tian, S. Deronne, S. Latré, and J. Famaey, "Implementation and validation of an ieee 802.11ah module for ns-3," in *Proc. WNS3*, Seattle, WA, USA, Jun. 2016, p. 49C56.