ODGMAC: On-Demand Grouping-Based MAC for Dense IoT Networks

Yusen Zhou, Wenjia Wu, Hui Lv, Ming Yang, Feng Shan, *Member, IEEE*, and Junzhou Luo, *Member, IEEE*

***Abstract*—In recent years, the Internet of Things (IoT) has rapidly advanced, with applications ranging from smart homes to industrial manufacturing, often involving densely deployed nodes such as temperature and humidity sensors. Since these nodes have limited computation and energy, the use of stuffed Wi-Fi management frames for data transmission has emerged as a promising way to avoid the association overhead of the traditional transmission mode. However, this unassociated data transmission mode continues to encounter significant channel contention in dense deployments. To this end, we propose ODGMAC, an on-demand grouping-based MAC solution that dynamically groups transmission-awaiting nodes and allocates time slots on a per-group basis, thereby enabling intra-group contention to improve transmission efficiency and reduce node energy consumption. Firstly, we present a fuzzy control-based algorithm at the access point (AP) to dynamically identify nodes with transmission demands in the current beacon period. On this basis, we then propose a hierarchical group-based time slot allocation methodology. Specifically, the nodes are initially clustered according to their per-packet airtime requirements. Within each cluster, we evenly partition nodes into multiple groups and assign each group to a unique time slot for channel contention, where the optimal slot count is determined by a renewal-theory–based analytical model with a discrete search over candidate counts. Finally, we implement the ODGMAC testbed with one AP and 100 IoT nodes, and conduct real-world experiments in a dense environment. The experimental results show that our solution outperforms existing methods in terms of both data delivery rate and node power consumption. Specifically, under severe channel collision conditions, our solution achieves an average increase of 14.77% in data delivery rate and an average reduction of 8.21% in node power consumption, while maintaining excellent fairness. Moreover, extended simulations show that our solution scales to 1000 nodes and maintains excellent performance under node mobility.**

***Index Terms*—Grouping-based MAC, dense IoT nodes, transmission demand estimation, energy efficiency.**

Received 18 February 2025; revised 6 February 2026; accepted 9 March 2026. Date of publication 13 March 2026; date of current version 6 August 2026. This work was supported in part by the National Key R&D Program of China under Grant 2023YFC3605800, in part by the National Natural Science Foundation of China under Grant 62132009, Grant 62232004, and Grant 62472090, in part by the National Natural Science Foundation of Jiangsu under Grant BK20242026, in part by the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201, and in part by the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under Grant 93K-9. Recommended for acceptance by A. Abdrabou. *(Corresponding author: Wenjia Wu.)*

Yusen Zhou, Wenjia Wu, Ming Yang, Feng Shan, and Junzhou Luo are with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: yszhou@seu.edu.cn; wjwu@seu.edu.cn; yangming2002@seu.edu.cn; shanfeng@seu.edu.cn; jluo@seu.edu.cn).

Hui Lv is with the College of Software Engineering, Southeast University, Nanjing 211189, China (e-mail: lvhui@seu.edu.cn).

Digital Object Identifier 10.1109/TMC.2026.3673836

I. Introduction

The rapid development of Internet of Things (IoT) technologies has significantly accelerated the transformation and upgrading of traditional economies and societies toward digitalization and intelligence. According to GSMA statistics, the total number of IoT connections worldwide is expected to reach 38.5 billion by 2030 [1]. It is clear that IoT applications have become widespread, infiltrating various fields of production and everyday life. For example, the traditional manufacturing industry utilizes the data sensing and interconnectivity of numerous IoT nodes to facilitate digital transformation, with smart factories serving as a key component that can automatically run entire production lines without human intervention [2], [3]. As shown in Fig. 1, a large number of IoT nodes are deployed densely in the smart factory, such as temperature and humidity sensors, smoke sensors, and acceleration sensors. These sensors collect environmental information, machine status, and other data in real time, and then upload it to the cloud platform through wireless access points (APs) for analysis and processing. Hence, a wireless network that carries these dense IoT nodes is fundamental and crucial for such IoT applications.

As one of the most popular wireless technologies today, Wi-Fi has been widely deployed and applied in IoT scenarios [4], [5]. A report by IoT Analytics indicates that 31% of global IoT connections use Wi-Fi as their communication mechanism [6]. Since most IoT nodes are lightweight with limited computational and energy resources, a low data-rate transmission mode, that is, utilizing the information elements of the management frames to stuff custom data to achieve unassociated data transmission [7], is more suitable than the traditional Wi-Fi transmission mode. The unassociated transmission mode can reduce the overhead caused by the node maintaining the association state, but still faces the challenges of channel contention in dense scenarios. Therefore, it is of great significance to propose the corresponding media access control (MAC) layer solution for dense IoT nodes that use such unassociated transmission mode.

Since dense IoT nodes access the wireless network at the same time, their concurrent traffic intensifies the contention of channel resources and increases the probability of data packet transmission collisions, which in turn leads to a series of problems, such as loss of data packets, increased delay, and reduced transmission rate in the network [8], [9], [10]. The currently widely used Wi-Fi MAC protocol is carrier-sense multiple access with collision avoidance (CSMA/CA), which cannot meet

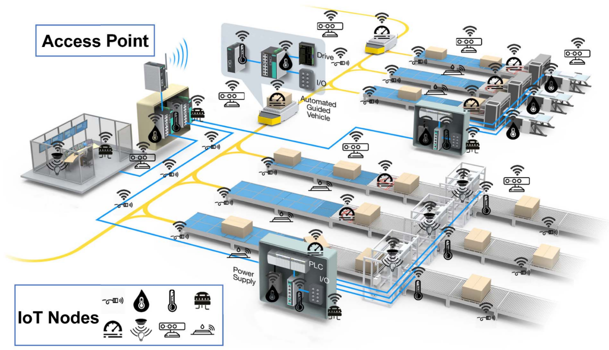


Fig. 1. High IoT node density in a smart factory. There are a large number of IoT nodes deployed here, responsible for regularly sampling the environmental information and machine status, and transmitting it to the cloud platform via wireless technologies.

the needs of concurrent communication of a large number of nodes, and frequent contention can also lead to additional energy consumption of nodes, thereby reducing their service life. To this end, prior studies introduce the time division multiple access (TDMA) mechanism for time slice division to strictly regulate channel use [11]. At the same time, some researchers propose the idea of group competition, which reduces the probability of collisions of IoT nodes and reduces frequent channel contention by setting a restricted access window (RAW) [12]. Meanwhile, existing reinforcement learning (RL)-based MAC protocols typically employ a centralized training and decentralized execution (CTDE) framework. By optimizing node rate control, adjusting contention windows, or allocating channel resources, they enhance network performance and node fairness metrics [13].

However, existing MAC schemes still cannot perfectly address the issues of dense IoT scenarios under unassociated transmission mode. Since the AP does not maintain the association status of the nodes, it becomes difficult to accurately estimate their transmission demands, and thus cannot make reasonable channel resource allocation. Although there are some traffic prediction methods [14], [15], [16], [17], [18], [19], [20], their estimation strategies are conservative. In the face of deviations caused by channel contention, they cannot converge quickly and have poor anti-interference capabilities. Furthermore, existing optimizations for channel access control in MAC protocols are inadequate for the performance demands of dense scenarios. In particular, RL-based solutions require complex state observations and substantial computations, making them difficult to deploy on resource-constrained APs and IoT nodes [13], [21], [22]. Moreover, in dense scenarios, their decision-making latency is also challenging to guarantee. Although various RAW-based node grouping strategies have been explored to mitigate collisions [12], [23], [24], the prohibitive overhead of dynamically grouping massive dense nodes, combined with the challenges of accurate modeling and conditional limitations, has resulted in most validations being conducted only through simulations, lacking real-world implementation and testing.

In this paper, we investigate the MAC mechanism for densely deployed IoT nodes with Wi-Fi unassociated transmission mode, and design an on-demand grouping-based MAC solution called ODGMAC to improve transmission efficiency and reduce node energy consumption. First, we present a fuzzy control-based algorithm at the AP to dynamically identify nodes with transmission demands in the current beacon period. Then, we propose a hierarchical group-based time slot allocation methodology. We cluster nodes by their per-packet airtime requirements and then optimize the number of slots for each group to maximize energy efficiency while mitigating contention and collisions. The main contributions of this paper are summarized as follows:

- We propose a fuzzy control-based transmission demand estimation algorithm, which models node transmission demands according to the characteristics of the unassociated transmission mode and utilizes a corresponding fuzzy proportional-integral-derivative (PID) control framework to predict node transmission demands in real time.
- We present a hierarchical group-based time slot allocation methodology. Nodes are first clustered by their per-packet airtime requirements. Within each cluster, we evenly partition nodes into multiple groups and assign each group to a unique time slot for channel contention. The optimal number of time slots is determined by a renewal-theory–based analytical model via a discrete search over candidate counts.
- We implement the corresponding MAC solution called ODGMAC and conduct real-world experiments in a dense environment with one AP and 100 IoT nodes. The experimental results show that our solution is superior to existing methods in terms of both data delivery rate and node energy consumption. Specifically, under severe channel collision conditions, the average data delivery rate increases by 14.77% and the average power consumption is reduced by 8.21% , with excellent fairness maintained. Moreover, extended simulations show that our solution scales to 1000 nodes and maintains excellent performance under node mobility. Our code is available at https://github.com/ilabot/ODGMAC.

II. Related Work

Recent advances in wireless networks have driven intensive research on dense IoT deployments, targeting higher channel utilization and improved energy efficiency under severe contention and collision. We first review Wi-Fi for IoT with unassociated transmission, then survey traffic prediction methods, and finally discuss MAC-layer channel access control and grouping strategies.

A. *Wi-Fi for IoT Networks*

Wi-Fi technology relies on complex modulation and coding techniques, along with the high-overhead mechanisms of the 802.11 protocol, to achieve high throughput. However, these features make it less suitable for IoT nodes with limited computational and energy resources. To enable lightweight and low-power Wi-Fi data transmission, some researchers have dedicated significant efforts to developing various solutions. Since WiFi requires nodes to wake up regularly to receive the AP's beacon frames and maintain association, which inevitably increases their energy consumption. To mitigate the additional overhead caused by association, some researchers have proposed optimized node wake-up mechanisms [25], [26], [27], [28]. Meanwhile, other studies have shifted their focus to management frame stuffing, introducing an innovative unassociated transmission mode.

Chandra et al. [29] first propose three stuffing schemes for beacon frames, which operate through the use of the service set identifier (SSID), the basic service set identifier (BSSID) and the vendor-specific information element (VSIE) fields. By stuffing these fields with customized content, many applications such as information publishing and passive indoor positioning can be realized [30], [31], [32], [33]. Later, Jaisinghani et al. [7] propose a response mechanism of probe request and probe response to achieve reliable data transmission and communication, providing an effective unassociated transmission mode for low-power IoT nodes. On this basis, they take into account the complex conditions of the channel and propose a solution for channel adaptability [34]. Through reasonable channel selection and dynamic transmission interval adjustment, they realize an adaptive transmission mechanism to cope with changes in channel conditions, and it can coexist with existing Wi-Fi networks in a scalable manner. In addition, Abedi et al. [35] conduct a fine-grained analysis and research on the power consumption characteristics of the existing power saving mechanism of Wi-Fi and the proposed unassociated communication protocol Wi-LE and low-power Bluetooth (BLE). The evaluation results show that the power consumption of Wi-LE is similar to that of BLE.

B. Traffic Prediction

Resource allocation relies on accurate, real-time prediction of end-to-end network traffic. Traditional statistical model-based traffic prediction methods have been proposed to track the linear and nonlinear characteristics of end-to-end network traffic [14]. However, these methods are not suitable for scenarios with dense deployment of IoT nodes. The primary reason is that IoT traffic in such environments exhibits significant irregular fluctuations due to channel congestion, making it difficult for a single model to fully account for these uncertainties. Meanwhile, machine learning has been widely applied to traffic prediction, enabling the capture of various traffic characteristics. Qiu et al. [15] focus on the spatiotemporal traffic characteristics of wireless networks, and design a recursive neural network for traffic prediction, considering both short-term and long-term dependencies. However, these approaches are also unsuitable for IoT networks due to the need for extensive infrastructure deployment and their reliance on prior information for sampling. Furthermore, deep learning-based methods [16] and RL-based methods [17] have emerged to address the traffic prediction problem. However, these methods primarily adopt passive estimation strategies that depend on historical data collection, making them particularly unreliable under channel congestion. As a result, they fail to effectively handle the complexities of such IoT traffic, where severe channel contention significantly affects prediction accuracy.

Therefore, some researchers propose traffic prediction approaches based on active allocation, which can mitigate the impact of channel contention and dynamically adapt to changing channel conditions. These approaches require real-time adjustment of the receiving frequency to align with the traffic patterns of IoT nodes, preventing excessive data accumulation at the nodes and reducing the risk of packet loss. For example, Shehab et al. [18] propose a traffic prediction framework for IoT nodes, which takes advantage of the traffic correlation in event and time dimensions, and employs a forward algorithm within a hidden Markov model (HMM) to predict the activation probability of each IoT node, allowing resource allocation to the node with the highest transmission probability. However, this algorithm requires sufficient historical data. For IoT nodes with low correlation and high variability in traffic patterns, it is necessary to increase the dimensions of the HMM model to address the issue, which significantly increases computational complexity. To simplify the complexity of model learning, Tian et al. [19] adopt the strategies of additive increase and multiplicative decrease to adjust the predicted transmission period. When a transmission failure occurs (i.e., a transmission opportunity is scheduled but no data is received), the predicted value is increased by twice the number of consecutive failures, allowing the interval to expand more rapidly. Conversely, when a transmission is successful (i.e., a transmission opportunity is scheduled and the data is received), the predicted value decreases by one, thereby reducing the interval. On this basis, they further utilize the "More Data" header field and the cross-slot boundary feature to gain a deeper understanding of traffic status and accelerate traffic prediction [20]. However, this approach still has certain limitations. It relies heavily on predefined rules and employs a conservative adjustment strategy. Consequently, it exhibits slow convergence in the presence of channel-interference-induced deviations, thereby weakening its anti-interference capability.

C. Channel Access Control

With the popularity of IoT nodes and the development of wireless technologies, the vast majority of IoT nodes transmit data using a wireless network, making the already scarce channel resources even more constrained. The traditional MAC layer technology, CSMA/CA, is clearly unable to solve this problem. As the number of nodes increases, the probability of transmission collisions increases dramatically. For traditional 802.11b/g/n networks, some researchers have conducted experiments to evaluate network performance under dense node conditions, and have verified its impact on channel contention [36], [37], [38]. In addition, Ganji et al. [10] build a real IoT testbed to perform comprehensive measurements on the performance of 802.11n/ac's MAC, and clarify the impact of the density and type of IoT traffic.

Currently, some researchers have improved the CSMA/CA's performance to meet the requirements of dense nodes. For example, Cordeschi et al. [39] propose a new CSMA strategy with conflict avoidance, which balances the contention probability and channel access time, named Delay-Collision CSMA. It is a time-slot non-persistent CSMA/CA with a non-uniform contention probability distribution, aiming to simultaneously reduce the delay of competitors and maintain a high access success probability. Modad et al. [11] introduce TDMA to solve the delay problem caused by frequent collisions, and propose a multi-channel hybrid CSMA/CA with adaptive TDMA enhancement, where one channel is only used for user association using CSMA/CA and other channels are used for user data transmission using TDMA to avoid interfering with the associating users. Furthermore, Fami et al. [40] propose a slicing solution for enterprise Wi-Fi networks, and perform dynamic user association to maximize the total throughput of the network with respect to different IoT requirements, which employs a RL algorithm to calculate the best matching strategy in near real time. The aforementioned research works have only been implemented and evaluated in simulation experiments, while implementation in real-world environments remains challenging. Moreover, these optimized CSMA/CA solutions increase the protocol complexity and lead to higher energy consumption in network nodes, making them unsuitable for energy-constrained IoT nodes.

For wireless transmission in IoT networks, the IEEE 802.11 working group has introduced the 802.11ah protocol, operating below 1 GHz. This protocol incorporates several innovative MAC layer technologies, including RAW, hierarchical association identification (AID), and traffic indication map (TIM). The RAW mechanism involves the concept of node grouping, which aims to reduce the probability of contention collisions between nodes by limiting the same group of nodes to compete for channels in the allocated time slots. The 802.11ah protocol does not specify a clear grouping rule, and performs equal grouping based on the association order. However, since different nodes have varying traffic demands, such simple grouping cannot effectively reduce the probability of contention collisions. Therefore, some researchers investigate the grouping strategy of the RAW mechanism. Tian et al. [19] propose a real-time grouping strategy to optimize RAW parameters by analyzing the current traffic conditions. However, this scheme only considers uniform and saturated network conditions. In addition, they use the proxy model method [23], [24] to improve the model to support different modulation and coding schemes (MCS) of the node. Compared with the traditional CSMA/CA, they increase throughput by approximately 43% . Chang et al. [12] believe that the performance of grouping is closely related to the different traffic requirements of the nodes, and propose a load-balanced grouping algorithm to improve the channel utilization of each group. By assuming that the traffic demand of the node is known and limiting the number of RAW groups, the node distributes evenly to different RAW groups according to the time the traffic occupies the channel. Furthermore, they propose a traffic-aware IoT node grouping algorithm [41]. Based on the throughput results of two extreme cases (i.e., saturation state and each RAW group only sends one data packet), they derive a regression model of competition success probability under different traffic demands, thereby determining the number and duration of RAW groups under any traffic. Similarly, Marcelli et al. [42] propose a Multi-way Karmarkar-Karp (KK) based grouping algorithm, which takes into account the heterogeneity of sensor traffic demands and minimizes the difference in transmission time sums across groups, thereby achieving a more balanced load compared to greedy approaches. Although these schemes effectively address the channel contention model under different traffic demands, such fixed-group designs, combined with equal slot durations and prior-known demand assumptions, limit scalability and adaptability in dense IoT scenarios, as the optimal grouping must be recalculated when the scenario changes. In addition, in a real environment, the traffic demand of the node changes due to channel conditions and time, and the assumption of known traffic demand becomes too idealistic. To solve the complex modeling problem of the avoidance and concession process of node competition channels, some researchers propose using deep reinforcement learning (DRL) methods [43], [44] to self-learn and optimize the RAW grouping parameters, and some researchers also propose using probabilistic statistical methods [45], [46] to simplify the complex modeling of the channel contention process. For example, Taramit et al. [46] use the renewal theory to solve the grouping solution for maximizing channel utilization, but it depends on homogeneous traffic patterns.

Furthermore, some studies employ DRL to design MAC protocols that mitigate channel contention in dense networks. Yu et al. [21] propose DLMA, a DRL-based MAC protocol for heterogeneous time-slotted networks. It adopts a central gateway to coordinate the nodes as a virtual big agent for near-optimal performance without prior knowledge of coexisting MACs. Although effective, its assumptions of saturated traffic and reliance on centralized coordination limit its applicability in resource-constrained or lightweight IoT scenarios. Nisioti et al. [22] design a coordinated RL-based MAC protocol under irregular repetition slotted ALOHA (IRSA), where the global Q-function is decomposed over a coordination graph and joint actions are selected via max-sum message passing on the IRSA-induced bipartite graph. Although pruning strategies are employed to reduce computational cost, the protocol's per-frame coordination and convergence checks still incur substantial latency, hindering its deployment on resource-constrained devices. Moon et al. [13] present Neuro-DCF, a multi-agent reinforcement learning based CSMA controller that uses centralized training with a graph neural network critic. Although its decentralized actors achieve gains over standard protocols without control-plane messaging, the framework suffers from high computational overhead and is constrained by parameter sharing in heterogeneous environments. Consequently, lightweight grouping strategies that avoid computationally intensive learning are preferable for resource-constrained IoT networks.

In contrast to computationally intensive DRL methods and probabilistic models that assume homogeneous traffic, appropriate node grouping strategies can adapt to traffic heterogeneity in dense deployments, thereby maximizing channel utilization and reducing node energy consumption.

III. System Model and Problem Formulation

We consider a dense IoT network consisting of an AP and a large number of IoT nodes, where the transmission demands of the IoT nodes are heterogeneous. We first present a transmission demand estimation model and an energy consumption model.

TABLE I
SUMMARY OF KEY NOTATIONS

Notations	Definition	Notations	Definition
I	set of dense IoT nodes	t	beacon-period index
N	total number of nodes	$\eta_i(t)$	data sampling interval
$T_i(t)$	node i's transmission demand at beacon t	$T_i'(t)$	node i's transmission estimate at beacon t
$S_i(t)$	node i's packet receiving state at beacon t	$M_i(t)$	node i's "More Data" flag at beacon t
l_i	node i's packet length	R_i	node i's transmission rate
D_i	node i's per-packet airtime requirement	T_{B}	duration of a beacon period
E_{SUCC}	energy consumption in the successful time units	E_{COLL}	energy consumption in the collision time units
E_{IDLE}	energy consumption in the idle time units	EE	energy efficiency
K	total number of time slots in a beacon period	T_s	duration of time slot s
C	set of clusters	EE_c^{SLOT}	energy efficiency of each time slot in cluster $c \in C$
T_c	duration of cluster c	T_c^{SLOT}	duration of each time slot in cluster c

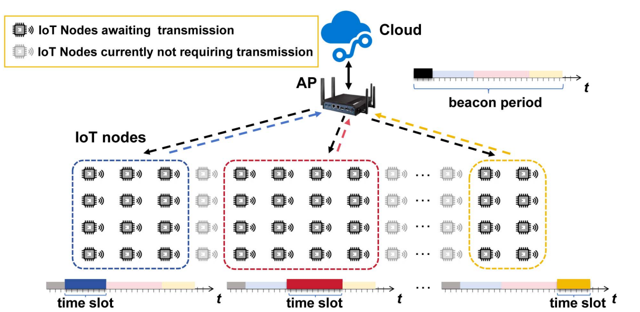


Fig. 2. A dense IoT network with an AP and a large number of IoT nodes, where the AP allocates transmission opportunities based on the current transmission demands of the nodes, and each node awaiting transmission uses its uplink to transmit data to the AP.

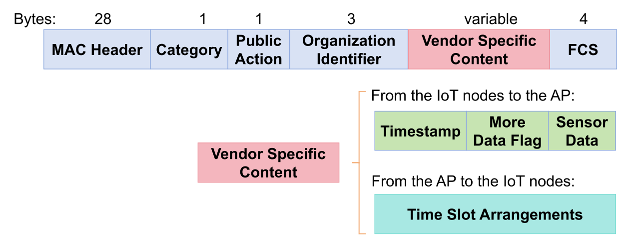


Fig. 3. Frame format for the unassociated transmission mode. We select a vendor-specific public action frame for custom data stuffing, where the Vendor Specific Content field carries sensor data for IoT nodes and time slot arrangement information for the AP.

Based on these models, we then formulate the corresponding transmission demand estimation problem and the channel resource allocation problem. Table I summarizes the key notations used in this paper.

A. Network Model

As shown in Fig. 2, we consider a dense IoT network with a single AP and a large number of IoT nodes. Densely deployed IoT nodes transmit their sensed data through wireless AP in the network, and the AP aggregates uplink traffic from each node to the cloud platform for processing. Due to differences in node functionalities, the sampling frequencies of their sensed data vary, resulting in heterogeneous transmission demands. Hence, the AP needs to allocate channel resources according to the current transmission demands of the nodes, and does not allocate transmission opportunities to nodes that currently do not require transmission. To improve resource utilization efficiency under heterogeneous transmission demands, the AP performs slot-based channel resource allocation within each beacon period. Specifically, the beacon period is partitioned into multiple contention time slots, and nodes that currently have transmission demands are assigned to designated slots, limiting their competition to these specific intervals. This slotted arrangement can mitigate collisions and reduce energy consumption.

In addition, the AP and IoT nodes operate in the unassociated transmission mode. To ensure compliance with the 802.11 protocol specification, we utilize a vendor-specific public action frame for custom data stuffing. The specific format of the action frame is illustrated in Fig. 3, where the Vendor Specific Content field serves as the designated stuffing area, its length being restricted by the maximum allowed MMPDU size. The IoT nodes stuff sensor data, while the AP stuffs the information about time slot arrangements.

B. Transmission Demand Estimation Model

In order to efficiently allocate transmission opportunities to IoT nodes, it is necessary to predict the transmission demand of each node in real-time on the AP side. Therefore, the AP estimates the transmission demand of each node once per beacon period, and allows nodes with current demands to transmit data in the next period. The set of dense IoT nodes is defined by I, and the beacon-period index is defined by t. Assume that the total number of nodes is N, i.e., $|I| = N$. For current beacon period t, each node $i \in I$ has a relatively fixed data sampling interval, denoted by the integer $\eta_i(t)$. For any node i, its current transmission demand (i.e., transmission interval) is defined as $T_i(t)$, where $1 \leq i \leq N$. Ideally, the transmission demand $T_i(t)$ should equal its data sampling period $\eta_i(t)$, i.e., $T_i(t) = \eta_i(t)$. The AP estimates the node's transmission demand based on the reception status $S_i(t)$ and the "More Data" flag $M_i(t)$ from the data packet header, and the predicted value is denoted by $T_i'(t)$.

As shown in Fig. 4, we present the transmission demand estimation model. The packet receiving state $S_i(t) \in \{0, 1\}$,

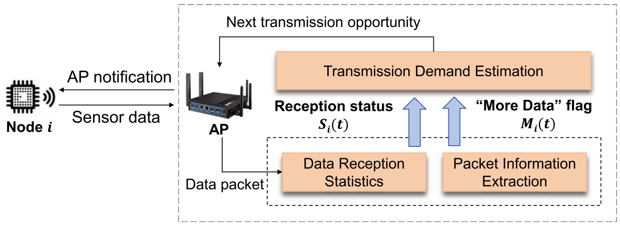


Fig. 4. Transmission demand estimation model. For node i in beacon period t, the reception status $S_i(t) \in \{0, 1\}$, where 1 indicates a successful data packet reception and 0 indicates failure. Failures occur either due to no new data packets in the node's queue or interference. The "More Data" flag $M_i(t) \in \{0, 1\}$ indicates whether there are additional data packets awaiting transmission.

the value of 1 means that the data packet sent by the target node is received within the time of the scheduled transmission opportunity, while the value of 0 is the opposite, indicating that the transmission opportunity is scheduled but the target data packet is not received. Reception failure mainly occurs in the following two situations: the first is that there is no new data packet generated in the node transmission queue and the packet cannot be sent, which means that the current estimated value $T_i'(t)$ is too small and needs to be increased over time; the second is that there is a data packet sent on the node side but the AP cannot receive it due to interference factors such as channel contention. The occurrence of this situation is regarded as a disturbance factor for the stability of the system. The "More Data" flag $M_i(t) \in \{0, 1\}$ is determined by the corresponding flag bit in the header of the received data packet. When the flag bit value is 1, it indicates that there is additional data waiting for transmission in the queue on the current node side, and the current estimated value $T_i'(t)$ should be reduced as soon as possible to clear the node queue; otherwise, it indicates that the node's transmission queue is empty at the current moment, which is an ideal state, and the current estimated value $T_i'(t)$ should be maintained. This enables the AP to estimate the node's transmission demands in real time.

C. Energy Consumption Model

IoT nodes with data to transmit in the current beacon period are active nodes. For active node i, its traffic pattern is characterized by the packet length l_i and the transmission rate R_i. We use D_i to represent the time that the node needs to occupy the channel to complete this transmission, also referred to as its per-packet airtime requirement. According to the IEEE 802.11 standard, D_i is defined as:

$$D_i = (l_i + l_{\text{ACK}})/R_i + t_{\text{SIFS}} + 2t_{\text{PLCP}} \tag{1}$$

where l_{ACK} is the length of the acknowledgment frame, t_{SIFS} is the short inter-frame space, and t_{PLCP} is the preamble time.

To better characterize the node's energy consumption, it is necessary first to model its process of accessing the channel. Discrete time is adopted, and a time unit represents the basic time duration in the channel. For a set of g nodes, the nodes need to compete for the channel to transmit data at the same time, so the mean analysis method [47] can be used to represent their transmission probability. According to the CSMA/CA protocol, the maximum number of retransmissions $R_{\max}$, the minimum contention window size $CW_{\min}$ and the maximum contention window size $CW_{\max}$ can be obtained. Therefore, the probability τ that a node transmits in an idle channel can be defined as:

$$\tau = \frac{E[R]}{E[B] + E[R]} \tag{2}$$

where $E[B]$ and $E[R]$ are the average number of backoffs and the average number of attempts to transmit a packet, respectively. For each data packet, given the conditional collision probability p, it follows a truncated geometric distribution with a success probability of $1 - p$. Therefore, $E[R]$ and $E[B]$ can be obtained as follows:

$$E[R] = \sum_{r=1}^{R_{\max}-1} r(1-p)p^{r-1} + R_{\max} p^{R_{\max}-1} = \sum_{r=1}^{R_{\max}} p^{r-1} \tag{3}$$

$$\begin{aligned} E[B] &= \frac{1}{2} \sum_{r=1}^{R_{\max}-1} \min(2^{r-1} CW_{\min}, CW_{\max})(1-p)p^{r-1} \\ &\quad + \frac{1}{2} \min(2^{R_{\max}-1} CW_{\min}, CW_{\max}) p^{R_{\max}-1} \\ &= \frac{1}{2} \sum_{r=1}^{R_{\max}} \min(2^{r-1} CW_{\min}, CW_{\max}) p^{r-1} \end{aligned} \tag{4}$$

In the case of network saturation, for a total of g nodes, the transmission probability of each node is τ, and the conditional collision probability can be defined as $p = 1 - (1-\tau)^{g-1}$. Furthermore, combined with (2), the probability of the channel being idle, successful, and failed (i.e., collision) when g nodes compete for the channel in a time unit can be defined as:

$$P_{\text{IDLE}} = (1-\tau)^g \tag{5}$$

$$P_{\text{SUCC}} = g\tau(1-\tau)^{g-1} \tag{6}$$

$$P_{\text{COLL}} = \sum_{k=2}^{g} \binom{g}{k} \tau^k (1-\tau)^{g-k} \tag{7}$$

The sum of the three probabilities is always equal to 1. Therefore, according to the probability that the time units are in different states, the time for the three states can be calculated within a contention time slot, that is, the number of successful-transmission time units U_{SUCC}, the number of collision time units U_{COLL}, and the number of idle time units U_{IDLE}. α denotes the length of one time unit.

In addition, the power consumption of the node in different states is defined, such as the power consumption of channel listening e_{LISTEN}, the power consumption of data transmission e_{TX}, and the power consumption of idle waiting e_{IDLE}. In this way, the energy consumption corresponding to all competing nodes in different states of time units within the specified duration is calculated, which are defined as the energy consumption E_{SUCC} in the successful time units, the energy consumption E_{COLL} in the collision time units, and the energy consumption E_{IDLE} in the idle time units. The specific definitions are as follows:

$$E_{\text{SUCC}} = U_{\text{SUCC}} \cdot \alpha \cdot (e_{\text{TX}} + (g-1) \cdot e_{\text{LISTEN}}) \tag{8}$$

$$E_{\text{COLL}} = U_{\text{COLL}} \cdot \alpha \cdot g \cdot (\tau \cdot e_{\text{TX}} + (1-\tau) \cdot e_{\text{LISTEN}}) \quad (9)$$

$$E_{\text{IDLE}} = U_{\text{IDLE}} \cdot \alpha \cdot g \cdot e_{\text{IDLE}} \quad (10)$$

D. Problem Formulation

The design of an on-demand grouping-based MAC mechanism focuses on two fundamental problems: transmission demand estimation and channel resource allocation. These problems are formally formulated as follows.

1) Transmission Demand Estimation (TDE) Problem: We consider a dense IoT network with a set of nodes denoted by $I = \{1, 2, \ldots, N\}$. The data transmission of the nodes is periodic due to the sampling period of the sensors on the nodes. This transmission cycle represents the node's transmission demand, which exhibits a certain level of stability. At the same time, due to the functional differences of the sensors on the nodes, the transmission demands of the different nodes also vary. The TDE problem requires the AP to predict the transmission demand of each node $i \in I$ based solely on the header information of the received packet, that is, to accurately estimate the data sampling interval of the node. Meanwhile, the estimated value should converge quickly.

For node i in beacon period t, the AP adjusts its estimated transmission demand $T_i'(t)$ using decision variable $x_i(t)$:

$$T_i'(t) = T_i'(t-1) + x_i(t) \quad (11)$$

where the adjustment strategy for the decision variable $x_i(t)$ is based on the following conditions:

1) $S_i(t) = 1$ and $M_i(t) = 1$: The AP receives a data packet from node i. The value of the "More Data" flag in its header is 1, indicating that the previous estimate is too large, causing the node data packet to accumulate in the transmission queue. $x_i(t)$ should be a negative value to reduce the estimated value.
2) $S_i(t) = 1$ and $M_i(t) = 0$: The AP receives a data packet from node i. The value of the "More Data" flag in its header is 0, indicating that the previous estimate is already quite close, but there will still be estimation deviations. $x_i(t)$ should be 0 to maintain the estimated value.
3) $S_i(t) = 0$: The AP does not receive the data packet from node i as expected, indicating that the previous estimate is too small and it is not yet time to schedule the transmission opportunity, that is, the transmission queue of node i has no packet to send. $x_i(t)$ should be a positive value to increase the estimated value.

When $x_i(t)$ remains 0 for d consecutive beacon periods, we consider the estimated value to have converged to the true value. Therefore, we formulate the objective of the TDE problem as finding a prediction strategy $X_i = \{x_i(1), x_i(2), \ldots, x_i(r)\}$ for each node i, minimizing the index of convergence beacon period r to enable $T_i'(r)$ to reach $T_i(r)$, i.e.,

$$\min_X \{r \mid \forall j \in [0, d),\, T_i'(r-j) = T_i(r-j)\} \quad (12)$$

where d is a tunable parameter representing the required number of consecutive beacon periods for which the condition $T_i'(r-j) = T_i(r-j)$ must be satisfied in order to confirm convergence.

2) Channel Resource Allocation (CRA) Problem: In a dense scenario where a large number of active nodes with saturated traffic demands compete for channel access, we define the set of these nodes as I_{A}, which exhibit heterogeneous traffic patterns, i.e., different packet sizes and transmission rates. The CRA problem requires the AP to perform an allocation strategy within the limited channel resources to reasonably arrange the saturated heterogeneous traffic of the nodes.

This allocation strategy is executed once per beacon period. For each beacon period, it involves the division of time slots and the scheduling of active nodes in each time slot. Multiple nodes are scheduled in a time slot to compete for the channel. The set of time slots $S = \{1, 2, \ldots, K\}$ is defined, where K is the total number of time slots in a beacon period, that is, $|S| = K$. The duration of each time slot $s \in S$ is defined as T_s, and the set of nodes allocated in the time slot is defined as I_s. Therefore, the time period allocation strategy for channel resources can be defined as: $F = \{(T_1, I_1), (T_2, I_2), \ldots, (T_K, I_K)\}$, where each active node must be allocated to a time slot, i.e.,

$$\bigcup_{s=1}^{K} I_s = I_{\text{A}} \quad (13)$$

$$I_{s1} \cap I_{s2} = \emptyset, \forall s1, s2 \in S,\ s1 \neq s2 \quad (14)$$

In addition, the division of time slots must also meet the channel resource constraints:

$$\sum_{s=1}^{K} T_s = T_{\text{B}} \quad (15)$$

where T_{B} is the duration of a beacon period.

In addition, for any node i assigned to the time slot s, its per-packet airtime requirement D_i is also satisfied, that is, this transmission can be completed within T_s. Therefore, the minimum transmission duration constraint is expressed as follows.

$$T_s \geq D_i \quad \forall i \in I_s \quad (16)$$

Combined with the above energy consumption model, the energy efficiency EE_s for time slot s in a beacon period can be defined, considering the duration of T_s and the set of nodes I_s.

$$EE_s = \frac{E_{\text{SUCC}}^s}{E_{\text{SUCC}}^s + E_{\text{COLL}}^s + E_{\text{IDLE}}^s} \quad (17)$$

The objective is to find an allocation strategy F that maximizes the mean value of energy efficiency in all contention time slots.

$$\max_F \frac{1}{K} \sum_{s=1}^{K} EE_s$$

$$\text{subject to: } (13), (14), (15), (16) \quad (18)$$

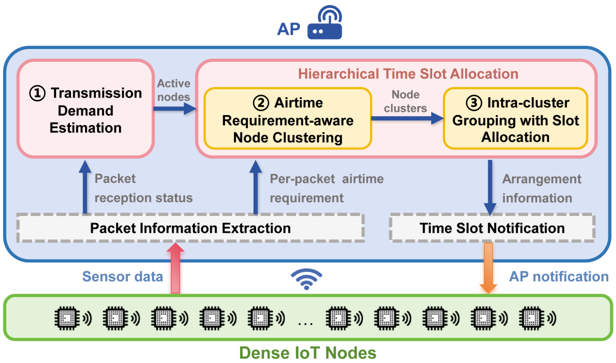


Fig. 5. Overall framework of ODGMAC. On the AP side, it predicts the transmission demands of IoT nodes according to their transmission status. On this basis, it clusters the IoT nodes and further divides each cluster's allocated period into multiple equallength time slots, with nodes evenly assigned so that each group corresponds to one slot. Hence, the ODGMAC mainly consists of three parts: transmission demand estimation, airtime requirement-aware node clustering, and intra-cluster grouping with slot allocation.

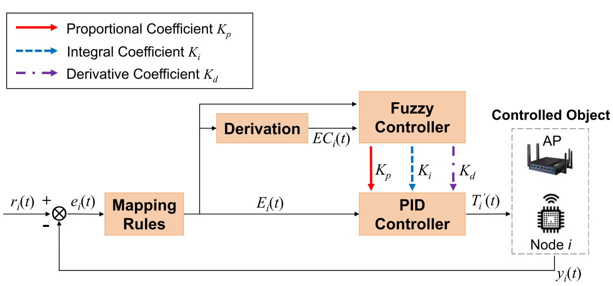


Fig. 6. Process of fuzzy PID-based transmission demand estimation algorithm. The PID controller is responsible for adjusting the output value in time according to feedback, the fuzzy controller guides the control parameters of the PID controller, and the mapping rules increase the change range of the feedback value.

IV. ODGMAC

To address the aforementioned problems, we propose our MAC solution called ODGMAC for dense IoT nodes in wireless networks.

A. Framework of ODGMAC

In this paper, we need to address the TDE problem and the CRA problem sequentially. After further analysis of the problems, we find that the CRA problem is difficult to solve directly. This is because as the number of nodes increases, the solution space expands dramatically, making the cost of obtaining an optimal solution prohibitively high. In order to meet the real-time requirements of resource allocation, we adopt a hierarchical strategy. First, we cluster nodes according to their per-packet airtime requirements, since allocating nodes with similar airtime requirements to compete for the channel is a reasonable way to maintain contention fairness. Subsequently, time slots are allocated to the nodes within each cluster. Given that their airtime requirements are identical, the solution space for this allocation problem is significantly simplified. Therefore, we design the overall framework of ODGMAC as shown in Fig. 5. The AP receives data from the IoT nodes, and then extracts the reception status and the "More Data" flag from the data packets to predict their transmission demands. Transmission demand estimation determines the active nodes, which are then subject to further processing. The per-packet airtime requirement of each active node is obtained from its packet length, and these nodes are clustered to form traffic-homogeneous groups. Within each cluster, nodes are then evenly partitioned into per-slot groups. Finally, we optimize only the per-cluster slot count via a renewal-theory-based analytical model with a discrete search over candidate counts, which reduces contention and improves energy efficiency while keeping the decision space deployable on resourceconstrained APs.

Based on the above, the core process of ODGMAC includes transmission demand estimation, airtime requirement-aware node clustering, and intra-cluster grouping with slot allocation, as detailed below.

1) *Transmission Demand Estimation:* The AP predicts the transmission cycle of each node based on the data packet reception status and the "More Data" flag.
2) *Airtime Requirement-aware Node Clustering:* The AP first calculates the airtime required by each node. Then, it organizes nodes with similar requirements into clusters. Finally, it allocates a time period to each cluster in proportion to the cluster's total requirement.
3) *Intra-cluster Grouping with Slot Allocation:* For each cluster, the AP determines the optimal number of time slots and groups the cluster's nodes accordingly, all guided by the energy efficiency metric.

B. Transmission Demand Estimation

According to the transmission demand estimation model in Fig. 4, we design the corresponding transmission demand estimation algorithm to achieve a fast and accurate estimation of node transmission demand based on the packet reception status information on the AP side.

Inspired by the idea of the PID controller and fuzzy control strategy, the transmission cycle of each node can be accurately predicted. The process is presented in Fig. 6. The input of the controlled object is the estimation value of node i's transmission cycle $T_i'(t)$ calculated by the algorithm based on historical feedback, and the output of the controlled object is the state feedback $y_i(t)$ obtained by the AP side based on $S_i(t)$ and $M_i(t)$, which is defined as follows:

$$y_i(t) = \begin{cases} 1 & \text{if } S_i(t) = 1 \text{ and } M_i(t) = 1 \\ 0 & \text{if } S_i(t) = 1 \text{ and } M_i(t) = 0 \\ -1 & \text{if } S_i(t) = 0 \end{cases} \tag{19}$$

Specifically, $r_i(t)$ is the expected feedback value for system stability. According to the state feedback $y_i(t)$, when $y_i(t) = 0$, the AP successfully receives the node data packet, and the node does not have cached packets, indicating that the current estimation value is close to the true value, i.e., $r_i(t) = 0$. Based on the difference between the expectation and feedback, the current deviation value $e_i(t)$ can be obtained, i.e., $e_i(t) = r_i(t) - y_i(t)$. However, due to the small range of change in $e_i(t)$, the feedback information that can be transmitted to the PID controller is

TABLE II
FUZZY RULE TABLE

E	EC						
	NB	NM	NS	ZO	PS	PM	PB
NB	PB/NB/PS	PB/NB/NS	PM/NM/NB	PM/NM/NB	PS/NS/NB	ZO/ZO/NM	ZO/ZO/PS
NM	PB/NB/PS	PB/NB/NS	PM/NM/NS	PS/NS/NM	PS/NS/NM	ZO/ZO/NS	NS/PS/ZO
NS	PM/NB/ZO	PM/NM/NS	PM/NS/NM	PS/NS/NM	PS/NS/NM	ZO/ZO/NS	NS/PS/ZO
ZO	PM/NM/ZO	PM/NM/NS	PS/NS/NM	ZO/ZO/NS	PS/NS/NS	NM/PM/ZO	NM/PB/ZO
PS	PS/NM/ZO	ZO/ZO/ZO	ZO/ZO/ZO	ZO/ZO/ZO	NS/PS/ZO	NM/PB/PS	NM/PB/ZO
PM	PS/ZO/PB	ZO/ZO/PM	NS/PS/PM	NM/PM/PS	NM/PB/PS	NM/PB/PB	NB/PB/PS
PB	NB/PB/PB	NB/PB/PB	NB/PB/PB	NB/PB/PB	NB/PB/PB	NB/PB/PB	NB/PB/PB

insufficient to improve the control response speed. Therefore, a mapping rule is defined based on the original $e_i(t)$ to obtain the new deviation value $E_i(t)$, i.e.,

$$E_i(t) = \begin{cases} e_i(t) & \text{if } e_i(t) \times e_i(t-1) \leq 0 \\ e_i(t) + E_i(t-1) & \text{if } e_i(t) \times e_i(t-1) > 0 \end{cases} \quad (20)$$

The fuzzy controller calculates appropriate control parameters under different combinations of deviation values and deviation change rates, specifically the proportional coefficient K_p, integral coefficient K_i, and derivative coefficient K_d. This is a process of fuzzification, fuzzy inference, and defuzzification, requiring the establishment of corresponding input-output membership functions and fuzzy rule tables. First, the input deviation value $E_i(t)$ and deviation change rate $EC_i(t)$ are fuzzified. In fuzzy control, triangular membership functions are often used for fuzzifying input variables, mapping actual continuous input values to fuzzy subsets, thus providing clear and operable inputs for fuzzy inference and control strategies. The triangular membership function divides the input into seven different fuzzy subsets: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). Then, fuzzy inference is performed based on the fuzzy rule base, where specific fuzzy rules are shown in Table II, defining the fuzzy values of control parameters K_p, K_i, and K_d under the fuzzy combination of deviation value E and deviation change rate EC. This fuzzy rule-based control strategy allows the system to make reasonable control decisions in the face of uncertainty and fuzziness, thereby improving the robustness and adaptability of the system. Finally, the weighted average method is used for defuzzification to obtain the specific control parameter values K_p, K_i, and K_d under this input.

The PID controller adjusts the output according to the combination of the proportional, integral, and derivative values of the deviation. Hence, the above three control parameters K_p, K_i, and K_d will affect the output of the PID controller for the estimation value of transmission cycle $T_i'(t)$, i.e.,

$$T_i'(t) = K_p E_i(t) + K_i \sum_{0}^{t} E_i(t) + K_d(E_i(t) - E_i(t-1)) \quad (21)$$

The detailed process of fuzzy PID-based transmission demand estimation is shown in Algorithm 1.

Algorithm 1: PID-Based Transmission Demand Estimation.

Input: $S_i(t)$, $M_i(t)$
Output: $T_i'(t)$
begin
  if $S_i(t) = 1$ **then**
    if $M_i(t) = 1$ **then**
      $y_i(t) \leftarrow 1$
    else
      $y_i(t) \leftarrow 0$
    end
  else
    $y_i(t) \leftarrow -1$
  end
  $e_i(t) \leftarrow 0 - y_i(t)$
  if $e_i(t) \cdot e_i(t-1) \leq 0$ **then**
    $E_i(t) \leftarrow e_i(t)$
  else
    $E_i(t) \leftarrow e_i(t) + E_i(t-1)$
  end
  $EC_i(t) \leftarrow E_i(t) - E_i(t-1)$
  Perform fuzzy calculation on $E_i(t)$ and $EC_i(t)$ to obtain K_p, K_i, K_d
  $T_i'(t)$ is obtained using Equation (21)
  return $T_i'(t)$
end

C. Airtime Requirement-Aware Node Clustering

The high density and massive scale of the IoT nodes lead to a vast solution space for the CRA problem. Hence, we adopt a practical density-plus-mean clustering heuristic to make the problem more tractable. For node clustering, we evaluate the per-packet airtime requirement of each node according to (1), and cluster nodes with similar requirements using a clustering algorithm. If a cluster exhibits an overly wide airtime spread, secondary clustering is applied to refine its range.

Specifically, the density-based clustering algorithm OPTICS is initially employed. Known for its robustness against noise and minimal sensitivity to input parameters, OPTICS achieves superior clustering performance and effectively identifies outliers. By defining two key parameters, that is, the domain radius eps and the neighbor number threshold $minPts$, it enables the identification of clusters ranging from dense to sparse regions. However, because of the closely distributed data densities, the algorithm may produce excessively wide ranges. To address this issue, a clustering range threshold $scopeLimit$ is introduced. Clusters exceeding this threshold undergo secondary clustering

Algorithm 2: Airtime Requirement-Aware Node Clustering.

Input: $I = \{1, 2, \ldots, N\}$, l_i, R_i, ξ_i^{LAST}, T_i', ξ_{CUR}, eps, $minPts$, $scopeLimit$
Output: C
begin
  $Q \leftarrow \emptyset$
  for $i \leftarrow 1$ **to** N **do**
    if $\xi_i^{\text{LAST}} + T_i' \leq \xi_{\text{CUR}}$ **then**
      $D_i \leftarrow (l_i + l_{\text{ACK}})/R_i + t_{\text{SIFS}} + 2t_{\text{PLCP}}$
      $Q \leftarrow Q \cup \{(i, D_i)\}$
    end
  end
  Using parameters eps and $minPts$, apply OPTICS clustering on Q to get set X
  for $x \in X$ **do**
    for $i \in x$ **do**
      $D_x^{\min} \leftarrow \min(D_x^{\min}, D_i)$
      $D_x^{\max} \leftarrow \max(D_x^{\max}, D_i)$
    end
    $scope \leftarrow D_x^{\max} - D_x^{\min}$
    if $scope > scopeLimit$ **then**
      $k \leftarrow \lceil scope/scopeLimit \rceil$
      Using parameter k, apply K-Means clustering on x to obtain set Y
      $C \leftarrow C \cup Y$
    end
    else
      $C \leftarrow C \cup \{x\}$
    end
  end
  return C
end

using the mean-based K-Means algorithm, which refines the clustering process and ensures a more constrained range for each cluster.

Additionally, due to the varying transmission cycles of nodes, only a subset of nodes have data to transmit during a beacon period. For node i in a beacon period, its next transmission opportunity is calculated using the last successful transmission timestamp ξ_i^{LAST} and the estimated transmission cycle T_i'. The set of nodes currently awaiting transmission, denoted as Q, is determined by comparing the current timestamp ξ_{CUR} with the estimated transmission time. Subsequently, the corresponding clustering algorithm is applied to the set Q to derive the node clustering for the current beacon period. The detailed process of airtime requirement-aware node clustering is shown in Algorithm 2.

D. Intra-Cluster Grouping With Slot Allocation

Based on node clustering, we first allocate channel resources to each cluster according to its current per-packet airtime requirements, dynamically partitioning the beacon period into per-cluster allocated periods. Specifically, the set of clusters is defined by C, with its size denoted as M, i.e., $C = \{1, 2, \ldots, M\}$. A beacon period T_{B} is preliminarily allocated for these groups. For each cluster $c \in C$, the duration of its allocated time period, denoted by T_c, is determined by the number of nodes in the cluster and their average per-packet airtime requirement. That is,

$$T_c = \frac{N_c D_c}{\sum_{c=1}^{M} N_c D_c} \times T_{\text{B}} \tag{22}$$

where N_c is the number of nodes in cluster c, and D_c is the average per-packet airtime requirement.

Given that nodes within a cluster exhibit similar requirement characteristics, they are evenly partitioned into multiple groups, with each group corresponding to a unique equal-length time slot. In particular, the time duration T_c needs to be evenly divided into multiple time slots, and the number of time slots is denoted by K_c. Hence, the length of each time slot is denoted by T_c^{SLOT}, i.e.,

$$T_c^{\text{SLOT}} = \frac{T_c}{K_c} \tag{23}$$

In this way, contention occurs only within the assigned slot. This clustering-then-grouping arrangement reduces collisions while keeping the solution space of the CRA problem computationally feasible.

To improve energy efficiency, it is also essential to optimize time slot division. Reducing the number of time slots increases the number of nodes competing for channel access within each slot, leading to more potential collisions. Conversely, dividing the time slots too finely results in fewer nodes competing per slot, which reduces the transmission probability per slot. Additionally, shorter time slots limit the number of transmissions that can occur within each slot, thereby decreasing channel utilization efficiency. Consequently, the number of time slots in a cluster directly influences the number of nodes competing within the same slot, which in turn determines the probability of transmission collisions among nodes.

In order to characterize the distribution of channel contention states within a time slot and determine the optimal parameters for energy efficiency, we employ a probabilistic statistical approach based on renewal theory. Renewal theory models a counting process for events whose occurrence probabilities follow an independent and identical distribution. Each event occurrence is termed a renewal, and the number of renewals within a specified time interval can be calculated. Since transmissions of nodes assigned to the same time slot can be approximated as independent and identically distributed, renewal theory can provide a suitable framework to characterize each time slot by estimating the expected number of transmission events occurring within the time slot [46].

In this paper, we use discrete time, that is, a time unit represents the basic time duration in the channel, typically 20 μs. As shown in Fig. 7, the duration of a transmission event consists of idle time units and busy time units, where the idle time is variable-length and the busy time is fixed-length. Hence, the duration of the transmission event can be expressed by the formula $y = x\alpha + \beta$, where x is a random number following a geometric distribution, α is the length of a time unit, and β is the time required for data transmission, that is, $\beta = D_c$ for

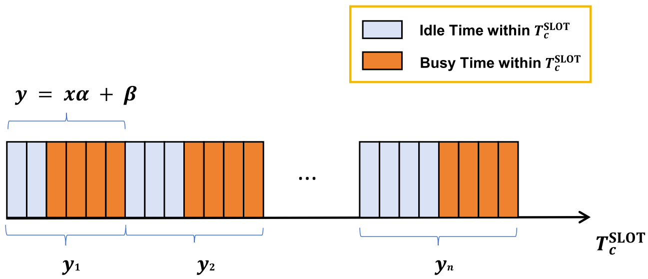


Fig. 7. In the duration of a time slot T_c^{SLOT}, there are n transmission events described by renewal theory, i.e., $y_1, y_2, \ldots, y_n$, where α is the length of the channel time unit, and β is the time required for data transmission.

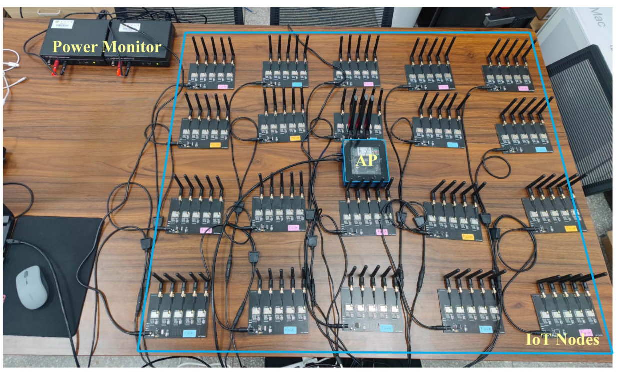


Fig. 8. We construct a dense experimental testbed comprising 100 IoT nodes and one AP. The IoT nodes are arranged in clusters of five on each board, with each node implemented as an energy-efficient ESP32C3 module. The AP is built on an Intel J4125 x86 architecture platform integrated with a Qualcomm QCA9880 wireless network interface card, running the OpenWrt open-source routing system.

the cluster c. The occurrence time of data transmission events follows a geometric distribution, and the renewal theory can be used to calculate the expected number of transmission-occupied in a time slot, i.e., $\mathrm{E}[U(T_c^{\text{SLOT}})]$. During an event, busy time units can be either time units with successful transmissions or time units with collisions, and thus the probability of the time units being in different states is calculated according to (5), (6), and (7), and the expected number of time units that are successful, collision, and idle within the T_c^{SLOT} time can be derived as follows:

$$\mathrm{E}[U_{\text{SUCC}}(T_c^{\text{SLOT}})] = \frac{\beta \cdot \mathrm{E}[U(T_c^{\text{SLOT}})] \cdot P_{\text{SUCC}}}{\alpha \cdot (1 - P_{\text{IDLE}})} \tag{24}$$

$$\mathrm{E}[U_{\text{COLL}}(T_c^{\text{SLOT}})] = \frac{\beta \cdot \mathrm{E}[U(T_c^{\text{SLOT}})] \cdot P_{\text{COLL}}}{\alpha \cdot (1 - P_{\text{IDLE}})} \tag{25}$$

$$\mathrm{E}[U_{\text{IDLE}}(T_c^{\text{SLOT}})] = \frac{T_c^{\text{SLOT}} - \mathrm{E}[U(T_c^{\text{SLOT}})] \cdot \beta}{\alpha} \tag{26}$$

The number of nodes in each group within cluster c is denoted by g, i.e., $g = N_c/K_c$, and the value of $\mathrm{E}[U(T_c^{\text{SLOT}})]$ can be obtained by probability calculation:

$$\mathrm{E}\left[U\left(T_c^{\text{SLOT}}\right)\right] = \sum_{i=1}^{\left\lfloor \frac{T_c^{\text{SLOT}}}{\beta} \right\rfloor} \sum_{j=0}^{\left\lfloor \frac{T_c^{\text{SLOT}} - i \cdot \beta}{\alpha} \right\rfloor} \binom{j+i-1}{j} \left(1 - P_{\text{IDLE}}\right)^i P_{\text{IDLE}}^j \tag{27}$$

The outer sum over $i = 1, \ldots, \lfloor T_c^{\text{SLOT}}/\beta \rfloor$ enumerates all feasible counts of renewals that can fit in the slot. For a given i, the inner sum over j accounts for all admissible totals of idle units preceding these i busy periods such that $i\beta + j\alpha \leq T_c^{\text{SLOT}}$. Since the per–time-unit idle probability is P_{IDLE}, each idle-run length is geometric with success parameter $(1 - P_{\text{IDLE}})$. Hence the total number of idle units in the i-th iteration follows a negative-binomial law with probability distribution $\binom{j+i-1}{j}(1 - P_{\text{IDLE}})^i P_{\text{IDLE}}^j$.

From the above analysis, we obtain the energy consumption for the three states in a time slot, i.e., $E_{\text{SUCC}}^{c,\text{SLOT}}$, $E_{\text{COLL}}^{c,\text{SLOT}}$, and $E_{\text{IDLE}}^{c,\text{SLOT}}$.

$$E_{\text{SUCC}}^{c,\text{SLOT}} = \mathrm{E}\left[U_{\text{SUCC}}\left(T_c^{\text{SLOT}}\right)\right] \cdot \alpha \cdot \left(e_{\text{TX}} + (g-1) \cdot e_{\text{LISTEN}}\right) \tag{28}$$

$$E_{\text{COLL}}^{c,\text{SLOT}} = \mathrm{E}\left[U_{\text{COLL}}\left(T_c^{\text{SLOT}}\right)\right] \cdot \alpha \cdot g \cdot \left(\tau \cdot e_{\text{TX}} + (1-\tau) \cdot e_{\text{LISTEN}}\right) \tag{29}$$

$$E_{\text{IDLE}}^{c,\text{SLOT}} = \mathrm{E}\left[U_{\text{IDLE}}\left(T_c^{\text{SLOT}}\right)\right] \cdot \alpha \cdot g \cdot e_{\text{IDLE}} \tag{30}$$

Thus, using (17), the energy efficiency for this time slot is given as follows.

$$EE_c^{\text{SLOT}} = \frac{E_{\text{SUCC}}^{c,\text{SLOT}}}{E_{\text{SUCC}}^{c,\text{SLOT}} + E_{\text{COLL}}^{c,\text{SLOT}} + E_{\text{IDLE}}^{c,\text{SLOT}}} \tag{31}$$

In addition, we find that an excessively large K_c value will lead to a significant reduction in transmission opportunities for a cluster, due to unusable time fragments in the slots. To this end, we incorporate a weighting coefficient Υ_c into the optimization objective, which is defined as the expected number of transmissions over the entire time period of cluster c. While this coefficient's influence is limited when K_c is small, its value drops significantly when K_c is excessively large. As a result, the optimization process penalizes overly large values of K_c.

$$\Upsilon_c = K_c \cdot \mathrm{E}[U(T_c^{\text{SLOT}})] \tag{32}$$

Therefore, the CRA problem for a single cluster can be formulated as follows.

$$\max_{K_c} \quad \Upsilon_c \cdot EE_c^{\text{SLOT}}$$
$$\text{s.t. } 1 \leq K_c \leq \left\lfloor \frac{T_c}{D_c} \right\rfloor \tag{33}$$

Clearly, the solution space of the problem is limited to discrete values of K_c, and the optimal value can be obtained via a simple search strategy. Algorithm 3 shows the detailed process to obtain the optimal number of time slots for cluster c.

In summary, by determining the number of time slots for each cluster to maximize energy efficiency, a reasonable allocation of channel resources is achieved. This approach not only mitigates channel collisions and reduces node energy consumption but also accommodates heterogeneous per-packet airtime requirements.

Algorithm 3: Intra-Cluster Grouping With Slot Allocation.

Input: N_c, D_c, T_c
Output: K_c
begin
  $k_{\text{max}} \leftarrow T_c/D_c$
  $O_{\text{max}} \leftarrow 0$
  for $k \leftarrow 1$ **to** k_{max} **do**
    $T_c^{\text{SLOT}} \leftarrow T_c/k$
    $g \leftarrow N_c/k$
    Obtain P_{IDLE}, P_{SUCC}, and P_{COLL} by Eq. (5), (6) and (7)
    Compute $E[U(T_c^{\text{SLOT}})]$ according to Eq. (27)
    Compute EE_c^{SLOT} according to Eq. (17)
    $\Upsilon_c \leftarrow k \cdot E[U(T_c^{\text{SLOT}})]$
    if $O_{max} < \Upsilon_c \cdot EE_c^{\text{SLOT}}$ **then**
      $O_{\text{max}} \leftarrow \Upsilon_c \cdot EE_c^{\text{SLOT}}$
      $k_{\text{best}} \leftarrow k$
    end
  end
  $K_c \leftarrow k_{\text{best}}$
  return K_c
end

TABLE III
PARAMETERS OF EXPERIMENTAL ENVIRONMENT

Parameter	Value
Wi-Fi Protocol Standard	802.11b
Beacon Period T_{B}	200 ms
Estimation Convergence Parameter d	3
Length of Channel Time Unit α	20 µs
ACK Frame Length l_{ACK}	14 B
Short Inter-Frame Spacing t_{SIFS}	10 µs
PLCP Preamble Length t_{PLCP}	96 µs
Node Idle Power Consumption e_{IDLE}	32 mW
Node Transmission Power Consumption e_{TX}	1100 mW
Node Listening Power Consumption e_{LISTEN}	270 mW
IoT Node Number N	100
IoT Node Sampling Period $\eta_i(t)$	10/30/50 T_B
IoT Node Data Packet Size $l_i(t)$	20/150/320 B
IoT Node Transmission Rate $R_i(t)$	1/2/5/11 Mbps

TABLE IV
ALGORITHM PARAMETERS

Parameter	Value
Default Proportional Coefficient K_p	1.8
Default Integral Coefficient K_i	1.2
Default Derivative Coefficient K_d	0
Neighborhood Radius ε	150 µs
Minimum Number of Neighbors $minPts$	4
Grouping Range Threshold $scopeLimit$	300 µs

V. PERFORMANCE EVALUATION

In this section, we present empirical evaluations conducted in real-world scenarios to assess the efficacy of the proposed ODGMAC solution, with particular emphasis on two critical performance metrics: data delivery rate and node power consumption. The experimental results confirm the superior performance of our proposed solution in real-world environments.

A. Experiment Setup

We construct a dense experimental testbed comprising 100 IoT nodes and one AP, as shown in Fig. 8. The IoT nodes are arranged in clusters of five on each board, with each node implemented as a low-power ESP32C3 module featuring a 160 MHz single-core RISC-V processor, 400 KB SRAM, and 4 MB flash memory. The AP is built on an Intel J4125 x86 architecture platform integrated with a Qualcomm QCA9880 wireless network interface card, running the open-source routing system OpenWrt.[1] The ODGMAC solution is implemented through secondary development of the Wi-Fi management frame data stuffing and the wireless user access software package hostapd.[2] The experimental setup is deployed in a 15 m^2 indoor environment, where the 20 boards (100 nodes) are arranged in a 5×4 grid pattern with the AP positioned at the center of the node array. Node power consumption is measured using two Monsoon AAA10F power monitors. The detailed parameter settings of the experimental environment and the algorithms are provided in Tables III and IV, respectively. Specifically, node power consumption parameters are obtained from platform measurements under node transmission/listening/idle states, while the remaining parameters are chosen based on IoT traffic characteristics and empirical tuning. For example, a shorter beacon period would markedly increase channel occupancy and energy usage, whereas a longer one would degrade responsiveness to queue dynamics.

Our proposed ODGMAC consists of three processes: transmission demand estimation, airtime requirement-aware node clustering, and intra-cluster grouping with slot allocation, which correspond specifically to the three algorithms described above. In the experiments, we compare it with the following three existing MAC solutions.

- *TaGA:* The traffic-aware grouping algorithm [41] derives a regression-based analysis model to estimate the probability of successful competition under heterogeneous traffic and uses this probability to balance channel utilization of each group. However, this algorithm relies on the assumption that the traffic demands of the nodes are known in advance. Therefore, to enable comparative analysis with our proposed solution, our transmission demand estimation algorithm is implemented in conjunction with the TaGA solution.
- *E-TAROA:* Tian et al. [20] adopt the strategies of additive increase and multiplicative decrease to adjust the predicted transmission period, and propose an adaptive grouping optimization algorithm that combines the "More Data" flag to accelerate traffic estimation.
- *MW-KK:* The Multi-way KK based grouping algorithm [42] sorts sensors in descending order based on their transmission time requirements and iteratively combines subsets to minimize the workload difference between the most and least loaded groups. Similar to the TaGA solution above, we also incorporate our transmission demand

[1] https://openwrt.org/
[2] https://w1.fi/hostapd/

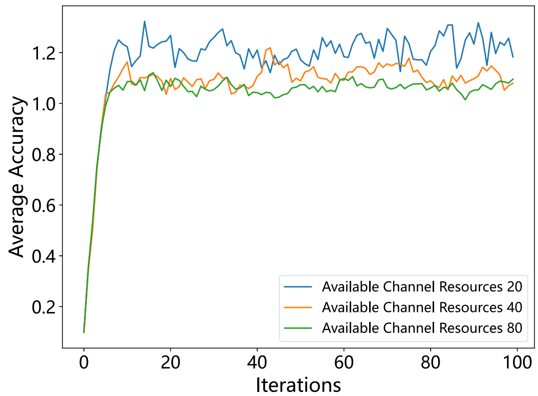


Fig. 9. Performance of transmission demand estimation. After 8 rounds of iterations, the transmission demand estimation of the node initially converges, and the prediction tends to be stable in subsequent iterations.

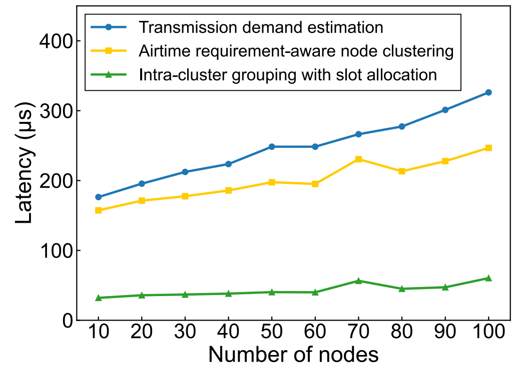


Fig. 10. Latency of the three phases with a varying number of nodes. The latency for all phases remains low across different number of nodes, demonstrating the system's efficiency for resource-constrained environments.

estimation algorithm into the solution implementation so as to facilitate the comparison of solutions.

B. Experiment Results

We first evaluate the performance of the transmission demand estimation algorithm in isolation, and then assess the performance of the overall ODGMAC solution.

1) Performance of Transmission Demand Estimation: To evaluate the accuracy of Algorithm 1 in predicting node transmission demand, the data sampling period of 100 IoT nodes is set to be evenly distributed from 5 to 54 beacon periods, and the average accuracy is calculated under different available channel resource conditions, where the available time for transmission in a single beacon period is set to 20 ms, 40 ms, and 80 ms. The shorter the time, the fewer the available channel resources. The accuracy is defined as the ratio of the predicted value $T_i'(t)$ to the true value $T_i(t)$.

As shown in Fig. 9, after 8 rounds of iterations, the nodes' transmission demand estimation initially converges, and the prediction tends to be stable in subsequent iterations. As the available channel resources decrease, channel collisions increase, and more packet loss will cause the predicted value to be greater than the true value. Overall, the algorithm is capable of rapid convergence and accurate prediction of the nodes' current transmission demands.

2) Analysis of System Overhead: To verify ODGMAC's suitability for resource-constrained platforms, we evaluate its computational overhead in terms of additional latency and CPU utilization. The experimental setup features a beacon period with an available transmission time of 80 ms. Each packet has a payload of 120B and is transmitted at a rate of 1 Mbps. The number of nodes is varied from 10 to 100 in steps of 10.

Fig. 10 presents the latency for the three phases of ODGMAC, that is, transmission demand estimation, airtime requirement-aware node clustering, and intra-cluster grouping with slot allocation. As shown in the figure, ODGMAC introduces very low additional latency. Even with 100 nodes, the total latency remains well below 1 ms, satisfying the real-time requirements of the MAC solution.

Additionally, the CPU utilization introduced by ODGMAC remains extremely low, consistently measuring below 0.1% . This confirms that even lightweight APs can handle the added computational burden without strain.

3) Impacts of Channel Conditions and Traffic Patterns: We evaluate the performance of ODGMAC in terms of data delivery rate and node power consumption. The data delivery rate is defined as the ratio of the number of packets received by the AP to the number of packets generated by the node. The node power consumption is obtained by monitoring the entire process using the power monitor. The average power consumption is obtained by sampling the average power performance of 10 IoT nodes (at different distances from the central AP) in the same period of time. These two metrics collectively form the basis for validating the energy efficiency (EE) objective in our mathematical model (31), where a higher delivery rate indicates larger proportion of energy consumption is used for effective transmission, and a lower node power reflects reduced total energy consumption.

To comprehensively evaluate the effects of diverse channel conditions and traffic patterns, systematic experiments are conducted on four critical dimensions: available channel resources, packet payload size, packet transmission rate, and data sampling period. In particular, the packet payload size is specifically configured based on statistical analysis of traffic patterns observed in real IoT nodes [48].

As illustrated in Fig. 11, the experimental results demonstrate a clear correlation between the available channel resources and the performance metrics. As available channel resources increase, the substantial reduction in channel collisions leads to two significant performance improvements, i.e., a corresponding improvement in the average delivery rate and a decrease in average power consumption. Notably, the system maintains robust performance even under severe channel collision conditions (the available channel resources are 20 ms), achieving an impressive average data delivery rate of 86.35% . Furthermore, the evaluation of various traffic patterns, including packet payload size, packet transmission rate, and data sampling period, reveals minimal impact on both the average delivery rate and power consumption metrics, indicating the system's resilience to diverse traffic patterns.

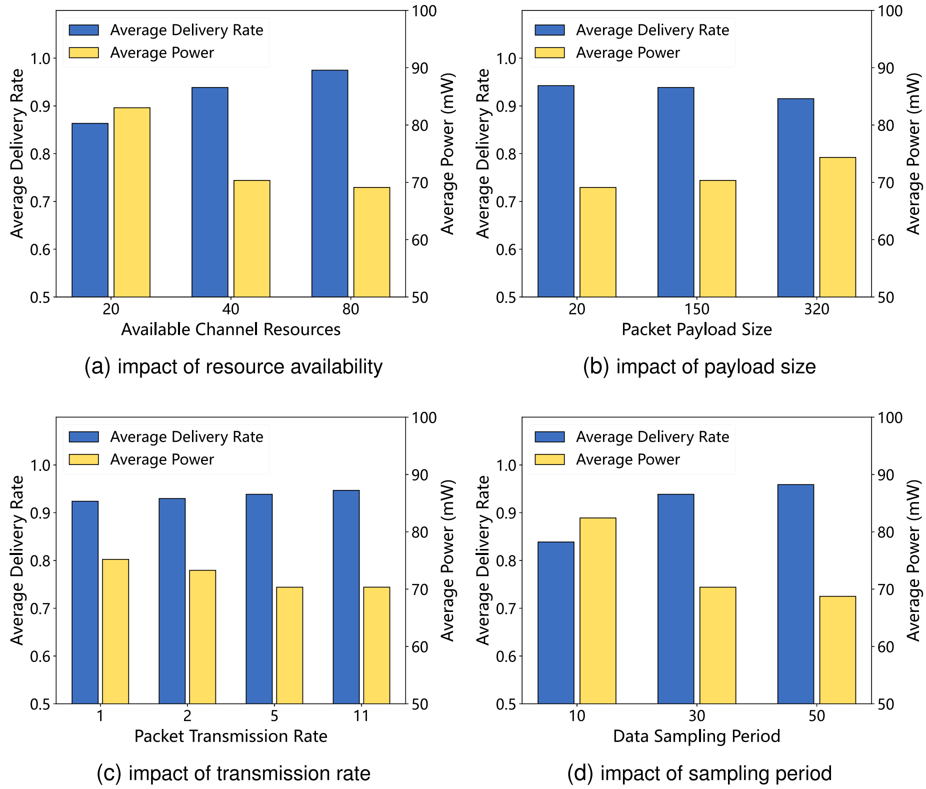


Fig. 11. Impacts of channel conditions and traffic patterns. Even in the case of severe channel collisions, the average delivery rate of the nodes can reach 86.35% . In addition, diverse traffic patterns have little effect on the average data delivery rate and node power consumption.

4) Impact of Heterogeneous Traffic: To validate the efficacy of node clustering, we conduct experiments under diverse heterogeneous traffic scenarios. The experiments incorporate three distinct traffic modes for IoT node configuration:

- *Model 1:* We set the packet payload to 20B, set the transmission rate to 5 Mbps, and set the sampling period to 40 beacon periods.
- *Model 2:* We set the packet payload to 150B, set the transmission rate to 2 Mbps, and set the sampling period to 30 beacon periods.
- *Model 3:* We set the packet payload to 320B, set the transmission rate to 1 Mbps, and set the sampling period to 20 beacon periods.

Based on these predefined modes, we establish five heterogeneous traffic scenarios by systematically varying the proportional distribution of traffic modes across IoT nodes.

As demonstrated in Fig. 12, the proposed solution maintains consistently robust performance across various heterogeneous traffic scenarios. The observed minor performance variations can be primarily attributed to the progressive increase in overall transmission demand within the network environment.

5) Performance Comparison: We compare our solution with three existing approaches, and evaluate the performance under diverse available channel resources (i.e., channel collision levels). The heterogeneous traffic configuration refers to the 1:2:2 scenario in the previous experiments.

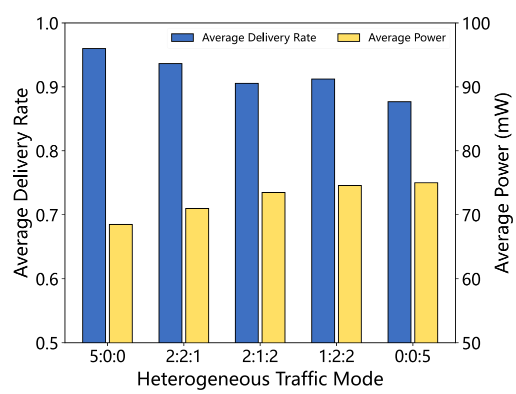
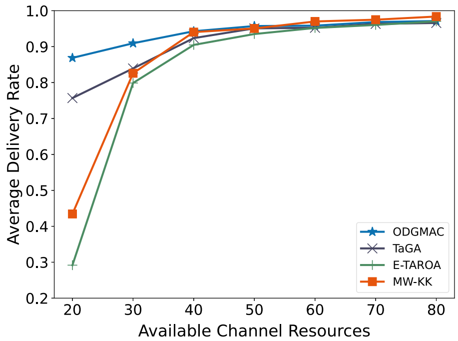


Fig. 12. Impact of heterogeneous traffic. Our solution can achieve good performance in different heterogeneous traffic scenarios. There is a slight difference in performance, it is due to the impact of overall transmission demand growth.

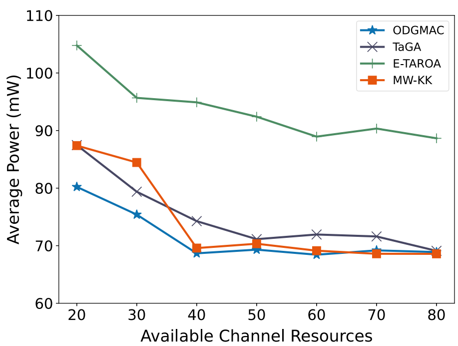
In addition to delivery rate and power consumption, we also use Jain's Fairness Index [49] to evaluate the fairness of node delivery rate. Specifically, let ϕ_i denote the delivery rate of node i, and the JFI over all N nodes is computed as

$$\mathrm{JFI} = \frac{\left(\sum_{i=1}^{N} \phi_i\right)^2}{N \sum_{i=1}^{N} \phi_i^2} \tag{34}$$

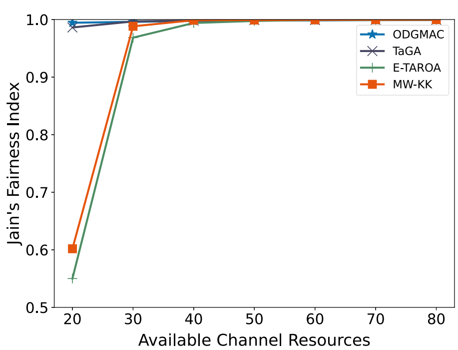
where a higher JFI value (ranging from 0 to 1) indicates greater fairness in the delivery rates allocated to each node.



(a) comparison of data delivery rate



(b) comparison of node power consumption



(c) comparison of delivery fairness

Fig. 13. Performance comparison. Especially in the case of severe channel congestion, our solution ODGMAC improves the average delivery rate by 14.77% , 197.70% and 100.02% respectively compared with TaGA, E-TAROA and MW-KK, and reduces the average power consumption by 8.30% , 23.45% and 8.21% , while maintaining high delivery fairness across nodes.

Fig. 13 illustrates the performance comparison under varying channel conditions. When available channel resources exceed 50 ms, indicating a low channel collision level, all four methods achieve an average delivery rate surpassing 95% . However, a distinct performance divergence emerges as channel resources diminish and collision frequency escalates. In particular, under severe channel collision conditions, the proposed solution ODGMAC demonstrates significant performance advantages, i.e., it improves the average delivery rate by 14.77% , 197.70% and 100.02% compared to TaGA, E-TAROA and MW-KK, and reduces average power consumption by 8.30% , 23.45% and 8.21% , respectively. Meanwhile, ODGMAC's JFI consistently approximates 1, indicating excellent fairness. This is especially evident under severe channel collision conditions, where it markedly surpasses E-TAROA and MW-KK. Overall, these performance improvements under challenging channel conditions validate the efficacy, robustness and fairness of the proposed approach.

Through further analysis of the experimental results, we find that E-TAROA exhibits significant limitations in processing cached data packets within node queues, primarily attributed to its slow convergence rate in transmission demand estimation and inadequate poor anti-interference ability. These technical constraints lead to substantial data accumulation, forcing nodes to maintain continuous channel monitoring and preventing them from entering sleep state, consequently resulting in elevated energy consumption. Although TaGA demonstrates improved overall performance by employing Algorithm 1 (proposed in this paper) for transmission demand estimation, it remains constrained by its reliance on a fixed packet number configuration. For MW-KK, balancing the aggregate transmission time across groups does not guarantee that contention intensity within each group is concurrently balanced. MW-KK handles heterogeneous traffic by concentrating numerous small-packet nodes with a few large-packet nodes to equalize transmission time. This results in some groups with intensely concentrated contention. Under limited channel resources, this concentration exacerbates collisions, backoffs, and retransmissions, severely undermining both the overall packet delivery rate and system fairness. These comparative results further substantiate the superior performance of our proposed solution in handling heterogeneous traffic patterns and operating in dense network environments.

VI. Extensive Simulation

To comprehensively evaluate the performance of our proposed ODGMAC solution, we conduct extensive simulations using the ns-3 network simulator, building upon an existing IEEE 802.11ah module [50]. The primary objective is to compare ODGMAC against the standard 802.11ah RAW mechanism in large-scale, dense IoT scenarios.

A. Simulation Setup

We consider a simulation scenario with one AP and 1000 IoT nodes, where the payload size for these nodes is varied among 32B, 64B, 128B and 256B to reflect common IoT data patterns. To evaluate network performance under varying levels of traffic saturation and channel contention, we construct multiple scenarios, each with a predefined total traffic demand. The total demand is then randomly distributed across all nodes to simulate sporadic and heterogeneous traffic patterns. For the baseline 802.11ah RAW configuration, we employ a single RAW Parameter Set, which divides the beacon period into 4 RAW slots, thereby allocating channel resources uniformly among the nodes. Simulations are performed under two distinct network scenarios: a static scenario, where all nodes remained stationary, and a mobile scenario, where nodes move according to a random-walk mobility model within a $1000\,\text{m} \times 1000\,\text{m}$ area, while moving from -500 m to 500 m relative to the centrally placed AP at the speed of 1 m/s.

TABLE V
SIMULATION PERFORMANCE COMPARISON

Network Scenario		Average Power Consumption (mW)		Average Delivery Rate (%)	
Type	**Traffic Demand**	**ODGMAC**	**ah-RAW**	**ODGMAC**	**ah-RAW**
Static	0.08	87.32	433.47	99.7	74.51
	0.1	101.82	561.67	99.54	68.65
	0.12	355.94	672.13	88.11	65.26
Mobile	0.04	61.32	81.08	99.48	91.6
	0.06	87.19	216.09	98.65	61.34
	0.08	278.84	366.65	55.48	42.71

B. Results and Analysis

The performance comparison between ODGMAC and the standard ah-RAW is summarized in Table V, focusing on two critical metrics: average node power consumption and packet delivery rate.

In static scenarios, ODGMAC exhibits a significant advantage. As the traffic demand increases from 0.08 Mbps to 0.12 Mbps, the power consumption of ah-RAW rises significantly from 433.47 mW to 672.13 mW, whereas ODGMAC maintains much lower consumption, peaking at 355.94 mW under the highest load. More importantly, ODGMAC sustains a high delivery rate of over 99.5% for demands of 0.08 and 0.1, only dropping to 88.11% at the 0.12 demand level. In contrast, the delivery rate for ah-RAW starts lower at 74.51% and further declines to 65.26% , highlighting its inefficiency in managing contention among a large number of static nodes.

ODGMAC's superiority is even more pronounced under mobile scenarios. At a low demand of 0.04 Mbps, both protocols perform well, yet ODGMAC consumes less power (61.32 mW vs. 81.08 mW). However, as demand increases to 0.06 and 0.08 Mbps, the performance of ah-RAW deteriorates sharply, with its delivery rate dropping sharply to 61.34% and 42.71% , respectively. In contrast, ODGMAC demonstrates significantly more robust performance. It maintains a delivery rate of 98.65% at 0.06 Mbps. Although experiencing a drop at 0.08 Mbps, it remains significantly more effective than ah-RAW. This confirms that our on-demand grouping and slot allocation strategy is highly effective in mitigating collisions and managing the challenges introduced by node mobility.

In summary, our simulations validate that ODGMAC significantly outperforms the standard 802.11ah RAW mechanism. It achieves a higher packet delivery rate and lower power consumption in various large-scale static and mobile IoT deployments. These results position ODGMAC as a highly adaptable and efficient MAC solution for dense IoT networks.

VII. DISCUSSION

The proposed ODGMAC solution is primarily designed for dense IoT scenarios characterized by periodic reporting, such as smart factories where massive sensors regularly upload environmental information or machine status. It introduces trade-offs in airtime efficiency, traffic adaptability and grouping optimality, as discussed below.

Data rate under unassociated transmission: The unassociated transmission mode is designed for lightweight IoT nodes with low-to-moderate data rates. It operates by using management-frame stuffing, which eliminates the overhead of association establishment and maintenance. However, because the IEEE 802.11 standard mandates that management frames be transmitted using mandatory basic rates and prohibits the use of advanced frame formats, this approach cannot leverage modern PHY/MAC enhancements available in association-based data transmission. Specifically, our ODGMAC solution is unable to utilize features such as MU-MIMO/OFDMA, frame aggregation, and Block ACK. These limitations stem from the compatibility-oriented design of management frames, thereby rendering the solution unsuitable for highrate devices.

Adaptability to traffic dynamics: Industrial IoT sensing and status reporting typically follow fixed sampling routines, which generate periodic traffic with relatively stable or slowly varying transmission demands over time. The fuzzy PID-based demand estimation is well-suited to this pattern by utilizing closed-loop feedback to effectively track gradual trends, suppress short-term fluctuations induced by channel contention, and avoid overly aggressive estimation updates, thereby providing stable and reliable demand estimations. However, under event-driven or highly bursty traffic, demand can shift abruptly. In such cases, the feedback-based estimation inherently requires several scheduling cycles to converge, leading to a transient degradation in the accuracy of airtime demand estimation until the controller re-converges.

Optimality in hierarchical grouping: The CRA problem is inherently complex and computationally challenging in dense deployments, primarily due to the vast combinatorial search space. To render this intractable problem manageable, ODGMAC decomposes CRA into a two-stage hierarchical methodology: it first clusters nodes by per-packet airtime demand and then determines each cluster's slot count using a renewal-theory-based mathematical model. By pre-grouping nodes into airtime-homogeneous clusters, this strategy substantially narrows the decision space, which in turn improves solving efficiency and leads to superior performance over existing methods. However, this two-stage decomposition, though necessary to manage complexity, can result in locally optimal solutions in certain cases. For instance, overly fine-grained clustering can restrict subsequent grouping options, potentially trapping the allocation in a local optimum.

VIII. Conclusion

In this paper, we investigate the channel access control problem for densely deployed IoT nodes, with the objective of reducing transmission collisions and improving energy efficiency. To solve this problem, we propose an on-demand grouping-based MAC solution called ODGMAC. First, we model node transmission demand by the characteristics of the unassociated transmission mode, and further develop a fuzzy PID-based algorithm to estimate node transmission demand, enabling real-time prediction. Then, we perform a hierarchical slot allocation methodology. The nodes are first clustered based on their per-packet airtime requirements, and then evenly partitioned within each cluster into per-slot groups for channel contention. Finally, experimental results show that our solution is superior to existing methods in terms of both data delivery rate and node power consumption. Specifically, under severe channel collision conditions, our solution achieves an average increase of 14.77% in the data delivery rate and an average reduction of 8.21% in node power consumption, without compromising fairness among nodes. Moreover, extended simulations confirm that ODGMAC effectively scales to 1000-node networks and maintains high performance under node mobility.

In the future, we plan to consider multi-AP scenarios, and investigate the joint AP coverage and MAC-layer resource allocation problem.

References

[1] M. Iji and R. Gurung, "IoT market outlook: Trends and drivers shaping connections growth through 2030," Jan. 2024. [Online]. Available: https://data.gsmaintelligence.com/research/research/research-2024/iot-market-outlook-trends-and-drivers-shaping-connections-growth-through-2030

[2] J. Wan, X. Li, H.-N. Dai, A. Kusiak, M. Martínez-García, and D. Li, "Artificial-intelligence-driven customized manufacturing factory: Key technologies, applications, and challenges," *Proc. IEEE*, vol. 109, no. 4, pp. 377–398, Apr. 2021.

[3] Y. Wu, H.-N. Dai, H. Wang, Z. Xiong, and S. Guo, "A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1175–1211, Secondquarter 2022.

[4] A. Khanna and S. Kaur, "Internet of things (IoT), applications and challenges: A comprehensive review," *Wireless Pers. Commun.*, vol. 114, no. 2, pp. 1687–1762, Sep. 2020.

[5] D. Glaroudis, A. Iossifides, and P. Chatzimisios, "Survey, comparison and research challenges of IoT application protocols for smart farming," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107037.

[6] "State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally," May 2023. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[7] D. Jaisinghani, G. Singh, H. Fulara, M. Maity, and V. Naik, "Elixir: Efficient data transfer in WiFi-based IoT nodes," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New Delhi, India:ACM Press, Oct. 2018, pp. 823–825.

[8] F. Rincon, Y. Tanaka, and T. Watteyne, "On the impact of WiFi on 2.4 GHz industrial IoT networks," in *Proc. IEEE Int. Conf. Ind. Internet*, Oct. 2018, pp. 33–39.

[9] L. Guegan and A.-C. Orgerie, "Estimating the end-to-end energy consumption of low-bandwidth IoT applications for WiFi devices," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci.* Sydney, Australia, Dec. 2019, pp. 287–294.

[10] A. Ganji, G. Page, and M. Shahzad, "Characterizing the performance of WiFi in dense IoT deployments," in *Proc. 28th Int. Conf. Comput. Commun. Netw.*, Valencia, Spain, Jul. 2019, pp. 1–9.

[11] B. A. A. Modad, E. Yaacoub, A. Kassir, and Z. Dawy, "A practical TDMA modification of IEEE 802.11 for ultra-dense IoT-health with fairness considerations," in *Proc. IEEE Int. Conf. Commun. Workshops*, Jun. 2020, pp. 1–6.

[12] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, and W.-T. Chen, "Load-balanced sensor grouping for IEEE 802.11ah networks," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2015, pp. 1–6.

[13] S. Moon, S. Ahn, K. Son, J. Park, and Y. Yi, "Neuro-DCF: Design of wireless MAC via multi-agent reinforcement learning approach," in *Proc. 22nd Int. Symp. Theory, Algorithmic Found., Protocol Des. Mobile Netw. Mobile Comput.*, Shanghai, China, Jul. 2021, pp. 141–150.

[14] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, Jun. 2019.

[15] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

[16] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.

[17] L. Nie et al., "A reinforcement learning-based network traffic prediction mechanism in intelligent Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2169–2180, Mar. 2021.

[18] M. Shehab, A. K. Hagelskjær, A. E. Kalør, P. Popovski, and H. Alves, "Traffic prediction based fast uplink grant for massive IoT," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–6.

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

[20] L. Tian, S. Santi, S. Latré, and J. Famaey, "Accurate sensor traffic estimation for station grouping in highly dense IEEE 802.11ah networks," in *Proc. 1st ACM Int. Workshop Eng. Reliable, Robust, Secure Embedded Wireless Sens. Syst.*, Delft Netherlands: ACM, Nov. 2017, pp. 1–9.

[21] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.

[22] E. Nisioti and N. Thomos, "Robust coordinated reinforcement learning for MAC design in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2211–2224, Oct. 2019.

[23] L. Tian, M. Mehari, S. Santi, S. Latre, E. De Poorter, and J. Famaey, "IEEE 802.11ah restricted access window surrogate model for real-time station grouping," in *Proc. IEEE 19th Int. Symp. "A World Wireless, Mobile Multimedia Netw."*, Chania, Greece, Jun. 2018, pp. 14–22.

[24] L. Tian, M. T. Mehari, S. Santi, S. Latré, E. De Poorter, and J. Famaey, "Multi-objective surrogate modeling for real-time energy-efficient station grouping in IEEE 802.11ah," *Pervasive Mobile Comput.*, vol. 57, pp. 33–48, Jul. 2019.

[25] J. Sheth and B. Dezfouli, "Enhancing the energy-efficiency and timeliness of IoT communication in WiFi networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9085–9097, Oct. 2019.

[26] M. Nurchis and B. Bellalta, "Target wake time: Scheduled access in IEEE 802.11ax WLANs," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 142–150, Apr. 2019.

[27] Q. Chen and Y.-H. Zhu, "Scheduling channel access based on target wake time mechanism in 802.11ax WLANs," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1529–1543, Mar. 2021.

[28] V. K. Ramanna, J. Sheth, S. Liu, and B. Dezfouli, "Towards understanding and enhancing association and long sleep in low-power WiFi IoT systems," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1833–1845, Dec. 2021.

[29] R. Chandra, P. Jitendra, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *Proc. 8th IEEE Workshop Mobile Comput. Syst. Appl.*, Tucson, AZ, USA, Mar. 2007, pp. 53–57.

[30] S. Zehl, N. Karowski, A. Zubow, and A. Wolisz, "LoWS: A complete open source solution for Wi-Fi beacon stuffing based location-based services," in *Proc. 9th IFIP Wireless Mobile Netw. Conf.*, Jul. 2016, pp. 25–32.

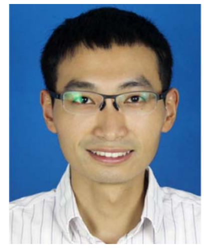
[31] S. Ham, J. Lee, and K. Lee, "QuickTalk: An association-free communication method for IoT devices in proximity," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 56:1–56:18, Sep. 2017.

[32] S. Zehl, A. Zubow, and A. Wolisz, "Demo: Stuffing Wi-Fi beacons for fun and profit," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Snowbird Utah USA:ACM, Oct. 2017, pp. 516–518.

[33] O. Waltari and J. Kangasharju, "Prongle: Lightweight communication over unassociated Wi-Fi," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Brno Czech Republic:ACM, Mar. 2020, pp. 994–1001.

[34] D. Jaisinghani, N. Gupta, M. Maity, and V. Naik, “Adaptive ViFi: A dynamic protocol for IoT nodes in challenged WiFi network conditions,” in *Proc.IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst.*, Delhi, India, Dec. 2020, pp. 147–155.
[35] A. Abedi, O. Abari, and T. Brecht, “Wi-LE: Can WiFi replace bluetooth?,” in *Proc. 18th ACM Workshop Hot Topics Netw.*, New York, NY, USA:ACM, Nov. 2019, pp. 117–124.
[36] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, “Characterizing user behavior and network performance in a public wireless LAN,” in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, New York, NY, USA:ACM, Jun. 2002, pp. 195–205.
[37] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, “Understanding congestion in IEEE 802.11b wireless networks,” in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, Berkeley, California:ACM, Oct. 2005, Art. no. 25.
[38] Z. Zhong, P. Kulkarni, F. Cao, Z. Fan, and S. Armour, “Issues and challenges in dense WiFi networks,” in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, Aug. 2015, pp. 947–951.
[39] N. Cordeschi, F. De Rango, and M. Tropea, “Exploiting an optimal delay-collision tradeoff in CSMA-based high-dense wireless systems,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2353–2366, Oct. 2021.
[40] F. Fami, N. Hammami, C. Pham, and K.-K. Nguyen, “Optimization of IoT slices in WiFi enterprise networks,” in *Proc. 37th ACM/SIGAPP Symp. Appl. Comput. Virtual Event: ACM*, Apr. 2022, pp. 1967–1974.
[41] 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. pp. 674–687, Mar. 2019.
[42] E. N. R. Marcelli, G. A. A. Blanco, A. R. S. Duarte, Y.-H. Chen, and T.-L. Chin, “Sensor grouping for efficient communication resource utilization for IEEE 802.11ah,” in *Proc. IEEE Latin- Amer. Conf. Commun.*, Nov. 2024, pp. 1–5.
[43] 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, Jun. 2024.
[44] X. Jiang, S. Gong, C. Deng, L. Li, and B. Gu, “Deep reinforcement learning for optimizing restricted access window in IEEE 802.11ah MAC layer,” *Sensors*, vol. 24, no. 10, May 2024, Art. no. 3031.
[45] S. Khairy, M. Han, L. X. Cai, Y. Cheng, and Z. Han, “A renewal theory based analytical model for multi-channel random access in IEEE 802.11ac/ax,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1000–1013, May 2019.
[46] H. Taramit, L. Orozco-Barbosa, and A. Haqiq, “A renewal theory based performance and configuration framework of the IEEE 802.11ah RAW mechanism,” *Digit. Commun. Netw.*, vol. 9, no. 1, pp. 236–251, Feb. 2023.
[47] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, “Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6180–6192, Nov. 2014.
[48] R. Kumar, M. Swarnkar, G. Singal, and N. Kumar, “IoT network traffic classification using machine learning algorithms: An experimental analysis,” *IEEE Internet Things J.*, vol. 9, no. 2, pp. 989–1008, Jan. 2022.
[49] R. K. Jain et al., “A quantitative measure of fairness and discrimination,” *Eastern Res. Lab., Digit. Equip. Corporation, Hudson, MA*, vol. 21, no. 1, pp. 2022–2023, 1984.
[50] L. Tian, S. Deronne, S. Latré, and J. Famaey, “Implementation and validation of an IEEE 802.11ah module for ns-3,” in *Proc. Workshop Ns-3*, New York, NY, USA:ACM, Jun. 2016, pp. 49–56.

Yusen Zhou received the BS degree from Hohai University, China, in 2021, and the MS degree in computer science from Southeast University, China, in 2024. His research focuses on wireless and mobile networks.



Wenjia Wu received the BS and PhD degrees in computer science in 2006 and 2013, respectively, from Southeast University. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University. His research focuses on wireless and mobile networks.

Hui Lv received the BS degree from the Nanjing University of Science and Technology, Nanjing, China, in 2023. He is currently working toward the MS degree with the College of Software Engineering, Southeast University, Nanjing, China. His research focuses on wireless and mobile networks.

Ming Yang received the MS and PhD degrees in computer science and engineering from Southeast University, Nanjing, China, in 2002 and 2007, respectively. He is currently a professor with the School of Computer Science and Engineering, Southeast University. His research interests include network security, privacy, and Internet of Things.

Feng Shan (Member, IEEE) received the PhD degree in computer science from Southeast University, Nanjing, China, in 2015. He visited the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, wireless networks, swarm intelligence, and algorithm design and analysis.

Junzhou Luo (Member, IEEE) received the BSc degree in applied mathematics and the MS and PhD degrees in computer network, from Southeast University, China, in 1982, 1992, and 2000, respectively. He is currently a full professor with the School of Computer Science and Engineering, Southeast University. His research interests include network architecture, network security, cloud computing, wireless network, and Internet of Things. He is a member of IEEE Computer Society and co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design. He is also a member of ACM and Co-Chair of ACM China Awards Committee.