Achieving Robust Resource Orchestration for Highly Dense Heterogeneous IoT Systems

Chun-Chih Lin*, Chenxu Jiang*, Xiaonan Zhang†, and Linke Guo*
*The Holcombe Department of Electrical and Computer Engineering, Clemson University
†Department of Computer Science, Florida State University
Email: {chunchi, chenxuj, linkeg}@clemson.edu, xzhang@cs.fsu.edu

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044551

***Abstract*—The proliferation of the Internet of Things (IoT) has ushered in a wide array of emerging applications. Current mainstream IoT protocols for supporting the above applications, such as Wi-Fi, ZigBee, and Bluetooth, heavily overlap on the 2.4 GHz bands. When deploying those heterogeneous IoT devices with different wireless protocols in a limited geographic area, e.g., manufacturing warehouse and clinic rooms, inevitable packet collisions will occur due to their spectrum overlapping. Those unpredictable collisions will ultimately degrade network performance, mainly due to the lack of coordination across coexisting protocols. This paper revisits the classic resource orchestration problem in a practical wireless coexistence scenario with a dense indoor IoT deployment. We propose to leverage multi-protocol gateways, e.g., Amazon Echo, Google Nest Hub, and Samsung SmartThing Station, to develop a Multi-Agent Reinforcement Learning (MARL) framework, which jointly considers channel status and contextual information for orchestrating limited resources. Based on the protocol heterogeneity and diverse transmission requests, we design a novel resource pool to achieve fine-grained management of available resources, by which gateways can collaboratively decide the system-level optimal strategy. The proposed design will also feature a cascaded RL model to determine a sequential decision for best utilizing available resources. Based on extensive real-world experiments conducted on a Software-Defined Radio (SDR) platform with up to 33 IoT devices, our proposed framework achieves more than 2.19X in throughput. It reduces 69.07% of delay compared with current random-accessed mechanisms.**

***Index Terms*—IoT, Highly Dense, Heterogeneity, Reinforcement Learning.**

I. Introduction

The increasing deployment of the Internet of Things (IoT) has promoted a plethora of emerging applications to benefit people's daily lives. Taking the industrial warehouse as an example, many heterogeneous IoT devices adopt different wireless protocols to perform multi-modal sensing. Regarding the IoT being used in this scenario, mainstream protocols, e.g., Wi-Fi, Bluetooth, and ZigBee, are heavily overlapped on the 2.4 GHz spectrum band, which creates a heterogeneous IoT environment requiring sophisticated interference (i.e., cross-technology interference) management. Even worse, an inevitable fact is that a multitude of heterogeneous IoT devices is often densely packed into a limited geographical area, making many existing resource orchestration schemes less effective. Besides the complex mutual interference among heterogeneous IoT devices, currently adopted random access protocols, e.g., CSMA/CA in Wi-Fi, may hinder maintaining or enhancing system-wise network throughput in a highly dense heterogeneous IoT network. Hence, to fully unleash the power of IoT to enable multi-modal sensing in highly dense scenarios, the following challenges should be addressed.

- **Challenge 1**: How to allocate limited resources to meet the needs of highly dense heterogeneous IoT devices with different protocol settings, spectrum/time requirements, power consumption, quality of service (QoS) needs, etc.?
- **Challenge 2**: Facing the nearly real-time control requirements, how to lower the computation complexity of resource orchestration needed to maintain the system performance?
- **Challenge 3**: Instead of using protocol-dependent gateways, can we leverage multi-protocol gateways with multiple RF ends to serve heterogeneous IoT devices?

Continue to use industrial IoT as an example. Different from the smart city IoT scenario, the industrial IoT is relatively static in terms of the total device number, deployed locations, spectrum usage, and transmission schedule. Those features make the wireless resources highly predictable, the computation complexity of resource orchestration manageable, and the multi-protocol gateway implementable (e.g., Amazon Echo, Google Nest Hub, and Samsung SmartThing Station). With less dynamic in the system, we argue, is there any way that we can use those commercialized multi-protocol gateways with a novel software-based design to achieve optimal resource orchestration for a highly dense scenario? In practice, the potential of multi-protocol gateways has not been fully unleashed mainly because each protocol (with its RF-end) in those gateways still follows its own scheduling mechanisms for only managing its supported IoT devices, lacking coordination across all protocols. Therefore, unpredicted collisions are still likely to occur when packets from heterogeneous IoT devices occupy the overlapping spectrum, resulting in severe packet errors and low spectrum utilization efficiency.

To fundamentally tackle resource orchestration in a highly dense heterogeneous IoT network, we propose a Multi-Agent Reinforcement Learning (MARL) framework to be implemented on each multi-protocol gateway (agent), in which both channel status of all RF ends and contextual information extracted from each packet will be jointly used in allocating resources. The framework allows all gateways to work collaboratively to derive the optimal resource orchestration strategy. Our main contributions are as follows,

- Instead of using traditional random access protocol, we

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

design a novel fine-grained spectrum/time resource pool for each gateway to observe and allocate. With the pre-organized resource allocation, IoT devices can transmit signals without any carrier sensing mechanism before transmission, e.g., CSMA/CA, which will significantly increase the channel access efficiency.

- By formulating the resource orchestration as a Markov Decision Process (MDP), we develop a cascaded RL model for the MARL framework to reduce the extremely large state and action space to enhance efficiency.
- We implement our framework on a real SDR platform and thoroughly evaluate the proposed design with more than 30 IoT devices in a small indoor environment to verify performance improvement.
- The proposed design helps achieve 68.14% of the optimal throughput, which is 2.19X than using the current random access-based approach. The system also reduces the delay compared with the random access by 69.07%.

The rest of this paper is organized as follows. Section II will first elaborate on the motivation of the proposed design by comparing it with classical scheduling mechanisms, in Sec. III, we will formulate the resource orchestration as a Markov Decision Process (MDP), followed by the cascaded RL in Sec. IV. Sec. V demonstrates the performance evaluation via extensive experiments. Then, we summarize existing works in Sec. VI. Finally, Sec. VII concludes the paper.

II. Motivation

We first revisit the classic Cross-technology Interference (CTI) issue in the highly dense IoT system via an empirical study. We will compare the overall performance of the current practice and our intuitive solution with a large number of heterogeneous IoT operating on the same spectrum band.

A. Empirical Study

1) Experimental Settings: For the empirical study, we consider a general wireless coexistence scenario where multiple Wi-Fi, ZigBee, and Bluetooth IoT devices are densely deployed. Assume there is a multi-protocol gateway to simultaneously receive uplink packets from 3 dedicated RF ends. Even though there are multiple channels for Wi-Fi to use, we focus on only one channel for simplicity in the empirical study. We will mostly focus on evaluating the impact of different MAC-layer protocols on the network throughput. The empirical study will be carried out in MATLAB using the following parameters in each protocol as in Table I.

	Wi-Fi	**ZigBee**	**Bluetooth**
Packet Duration	1 ms	4 ms	1 ms
Packet Interval	50 ms	100 ms	10 ms
Trans. Power	20 dBm	4.77 dBm	4.77 dBm
Bandwidth	20 MHz	2 MHz	1 MHz

TABLE I: Emprical Study Settings

2) Experimental Evaluation: Most existing MAC-layer designs are not designed for orchestrating radio resources to a large number of heterogeneous devices. Hence, the protocol-driven MAC layer schemes are less effective in providing reliable network performance. We mainly consider the following three MAC-layer schemes to show the performance degradation in a highly dense heterogeneous network.

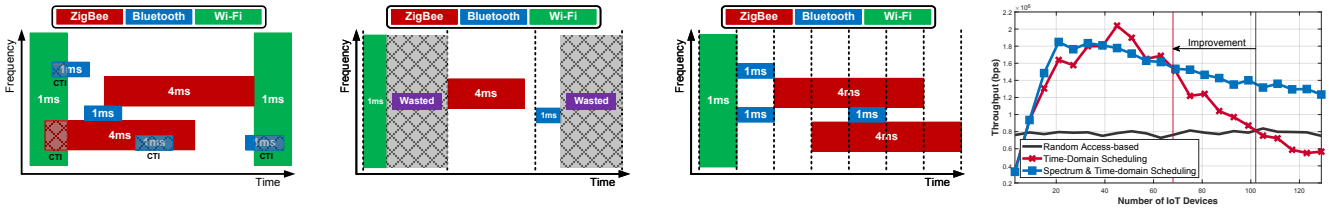
- **Random Access-based (Scheme Design 1)** As illustrated in Fig.1a, IoT devices transmit packets when there is a transmission request. While operating without any coordination, IoT devices adopt random access, i.e., CSMA/CA for Wi-Fi and ZigBee, and FHSS for Bluetooth, to contend a channel for transmission. Short packets, i.e., Wi-Fi packets, may sneak into adjacent long packets with a longer duration, i.e., ZigBee. Lots of unpredictable collisions may occur.
- **Time-Domain Scheduling (Scheme Design 2)** This scheme design dissects the time into slots, where each time slot is assigned to an IoT device. The minimum size of the time slot is set to be the largest transmission duration plus the packet interval, i.e., $4 + 100$ ms from ZigBee protocol. As illustrated in Fig.1b, this design ensures no collision among protocols but not time and spectrum-efficient.
- **Spectrum & Time-domain Scheduling (Scheme Design 3)** Our proposed design schedules both the time and frequency domain to avoid collisions as shown in Fig. 1c. When multiple heterogeneous IoT devices request to transmit simultaneously, the proposed design will choose one for immediate transmission and delay all other devices to a later time slot based on QoS requirements and currently available resources.

3) Experiment Results and Discussions: Our empirical study jointly considers the indoor environment, noise, and device distribution with the increase in the number of heterogeneous IoT devices (device ratio: Wi-Fi:ZigBee:Bluetooth=1:1:1). In Fig. 1d, the **Scheme Design 2** has the lowest overall throughput and remains stable when the number of IoT devices increases, which wastes the majority of spectrum resources. Since the fixed time slot strategy is adopted (regardless of protocols), increasing the device number will not impact its low overall throughput. However, using such a design in a highly dense heterogeneous network will cause significant delays for served IoT devices. Using **Scheme Design 1** will result in a nearly linear overall throughput drop, becoming even worse than **Scheme Design 2** and **Scheme Design 3** at around 65 devices and the worst when serving 102 devices, respectively.

As expected, adopting both spectrum and time-domain scheduling maintains a relatively high throughput when supporting more than 65 devices, becoming an ideal scheduling strategy for highly dense heterogeneous networks. In particular to the traffic distribution, Wi-Fi devices contribute the most to the peak overall throughput, whereas ZigBee and Bluetooth packets contribute around 13.0% and 7.7%, respectively.

- **Discussion.** This empirical study verifies our intuition that using a multi-protocol gateway with multiple RF front ends to coordinate traffic from heterogeneous IoT devices may outperform the random access-based approach, especially for

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.



(a) Random Access (b) Time-Domain Scheduling (c) Spec&Time Scheduling (d) Throughput Comparison

Fig. 1: MAC-layer Scheme Design Comparison

a dense deployment scenario. As shown in Fig. 1d where the proposed design improves the throughput using fewer devices, our objectives are to 1) further reduce the number of heterogeneous IoT devices needed to benefit from the proposed design (i.e., pushing the red line to the far left); and 2) increase the overall throughput when serving a smaller number of heterogeneous IoT devices.

B. Design Intuition

1) Network Environment and Assumptions: Consider a heterogeneous network with N heterogeneous IoT devices using coexisting protocols, including Wi-Fi, ZigBee, and Bluetooth. Those IoT devices will be served by a total of J gateways, each of which has three RF ends using the above protocols. To best simulate the highly dense scenario, we assume all IoT devices and gateways co-locate in a small indoor area, where every IoT device is within the transmission range of all gateways. Each IoT device requests resources for uplink transmission, and then the corresponding gateway will assign the suitable radio resource.

2) Radio Resource Slicing: Based on the above discussion, joint frequency and time-domain scheduling is expected to be beneficial to resource orchestration in highly dense heterogeneous networks. We take a step further to adopt the idea of "network slicing" in 5G [5] to achieve fine-grained control and scheduling over the available radio resources on 2.4 GHz. Specifically, as shown in Fig. 2, the radio resource "pool" is divided into spectrum and time, in which each block represents the usability of the current spectrum at a designated time slot.

Based on the observation of each uplink transmission in the previous time slot, every IoT device will be scheduled by this resource pool with the objective of maximally filling up the available blocks in the next time slot. As shown in Fig. 2, in addition to the resource block allocation obtained from the last time slot, the gateway will obtain the information from served IoT devices regarding different QoS requirements, packet lengths, bandwidth, whether or not to adopt frequency hopping, etc. Then, we expect the proposed design to predict the best resource orchestration strategy to be used in the next time slot. To achieve fine-grained resource orchestration, each block should be designed as small as possible. For our case, we choose the frequency unit as 1 MHz, i.e., the greatest common divisor (GCD) of the three coexisting protocols. Meanwhile, the duration of a packet depends on the length of data that the PLCP Service Data Unit (PSDU), the modulation scheme, and the data field length. Since there is no constant value to get the GCD, we consider each time unit to be 1 ms long.

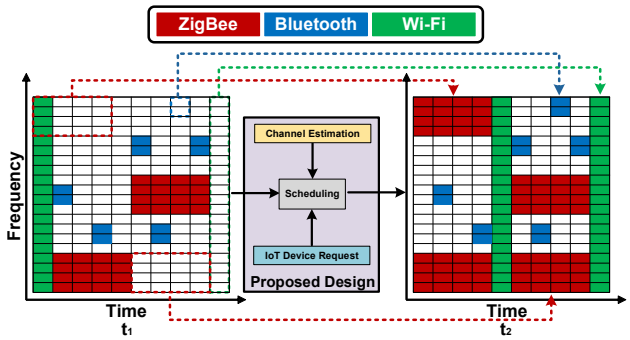


Fig. 2: Heterogeneous System Model

III. MDP Problem Formulation

A. Overview

Effective orchestration of limited resources relies on a full understanding of how the data is collected by heterogeneous IoT devices, criteria including different packet designs (duration), bandwidth, transmission power levels, QoS requirements, etc. Consider a total of N heterogeneous IoT devices scheduled for data collection, along with J multi-protocol gateways, performed as agents to serve the data collection. The resource orchestration for those heterogeneous IoT devices can be modeled as a Markov Decision Process (MDP), where gateways allocate resources and make joint decisions based on the observation of the previous time slot. Different from existing schemes, in our case, the multi-protocol gateway (agent) j is able to capture sufficient contextual information of its managed IoT data transmissions, namely, the current status at time slot t, $\mathbf{s}_j(t)$, as a part of the global state $\mathbf{s}(t)$. Based on this, each agent $j \in J$ will perform resource orchestration by taking the action $\mathcal{A}_j(t)$ and then get a certain reward $r_j(t)$.

B. State Space

The state $\mathbf{s}(t) \in \mathcal{S}$ describes the status information of the current wireless environment. Different from existing schemes, the highly dense wireless environment requires more detailed information to describe its state space, including both the channel condition and contextual information obtained from transmitted packets. For each gateway j, the state at time t is formulated as

$$\mathbf{s}_j(t) = [\mathbf{H}_j(t), \mathbf{C}_j(t), \tilde{n}_r, \tilde{n}_c, T_{n,Qos}(t), \rho_{n,Qos}(t)], \quad (1)$$

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

where $n \in N$, $\mathbf{H}_j(t)$ is the current channel condition, $\mathbf{C}_j(t)$ is the contextual information, $\tilde{n}_r$ is the time-frequency spectrum resource that the gateway can assign, $\tilde{n}_c$ is the set of the serving clients and $T_{n,Qos}(t)$ and $\rho_{n,Qos}(t)$ are latency and packet error rate QoS requirement for the client n.

• **Channel Estimation.** Each gateway obtains Channel State Information (CSI) by simultaneously estimating multiple channels based on long-term received packets, which contribute to

$$\mathbf{H}_j(t)[h_{n\to j}(t), h'_{n\to j}(t), p_{n\to j}(t)], \forall n, n' \in N, n' \neq n. \quad (2)$$

In particular, $h_{n\to j}(t)$ and $h'_{n\to j}(t)$ denote CSI from n and all other coexisting devices n', respectively. $p_{n\to j}(t) \in [p_{\min}, p_{\max}]$ represents the transmission power of n, which can be estimated from RSSI.

• **Contextual Information.** The extracted contextual information via traffic analysis by the gateway j,

$$\begin{aligned}\mathbf{C}_j(t)[&\alpha_{n\to j}(t-1), b_{n\to j,m}(t-1), \rho_{n\to j}(t-1),\\ &I_{n\to j}(t-1), T_{n\to j}(t-1)], \forall n \in N, m \in M.\end{aligned} \quad (3)$$

We define two binary indicators, $\alpha_{n\to j}(t-1)$ to denote whether the IoT device n transmits data to the gateway j, and $b_{n\to j}(t-1)$ describes if the channel m has been used by the client. The packet error rate (PER) $\rho_{n\to j}$, mainly due to interference, can be estimated together with the CSI as in [6]. Similarly, the gateway also estimates the interference (e.g., CTI), which can be given as

$$I_{n\to j}(t-1) = \sum_{n'\neq n, n'\in N} \alpha_{n'\to j}(t-1) p_{n'\to j}(t-1) h_{n'\to j}(t-1) \quad (4)$$

from all n' based on the input form $\mathbf{H}_j(t)$. $T_{n\to j}(t-1)$ denotes the actual transmission duration.

C. Action Space

The action $a \in \mathcal{A}$ assigned by gateways determines the behavior of the IoT device at the next transmission slot, including transmission power level, frequency bands, time slices, and/or the new gateway for uplink transmission.

Based on the discussions in Sec. I, the action space has to be significantly reduced to enhance the decision-making efficiency. To do this, we specifically define the following criteria for the action space, 1) the allocated frequency resources cannot be re-used; 2) Time slot should be allocated to each transmission; and 3) Gateway should reduce taking actions on dormant IoT.

Instead of taking a single action containing all the needed actions to the next state, the resource orchestration strategy in our design can be sequentially divided into the following four steps, gateway selection, frequency band allocation, choosing time slots, and adjusting transmission power levels. Hence, we define sub-action spaces for the MDP in our design, for which the action space can be re-written as,

$$\mathcal{A} = \{\mathcal{A}_j, \mathcal{A}_i, \mathcal{A}_{f,t}, \mathcal{A}_p\}, \quad (5)$$

where $\mathcal{A}_j = [1, \ldots, J]$, $\mathcal{A}_i = [1, \ldots, N_j]$, $\mathcal{A}_{t,f} = \{(\tau_t, \delta_f) | \tau_t = [\tau_1, \ldots, \tau_T], \delta_f = [\delta_1, \ldots, \delta_F]\}$, and $\mathcal{A}_p = [p_{\min}, p_{\max}]$, where J is the total number of gateways, N_j is the total number of served IoT devices for j-th gateway, δ_f is the f-th frequency slice, τ_t is the t-th time slot, and $p_{\min}$ and $p_{\max}$ are the min. and max. transmission power for the IoT device.

D. Reward

The objective of our resource orchestration scheme includes load balancing among gateways, lowering the noise/interference ratio, reducing transmitting power, and fulfilling both the time latency and packet error rate for QoS requirements, all of which should be addressed in the reward function.

The joint action $\mathcal{A}$ is made by multiple multi-protocol gateways. Hence, for each action taken by the gateway j, the global reward r_j can be defined as,

$$r_j = \beta + r_{j,\eta}(a_\eta, s) + r_{j,o}(a_o, s) + r_{j,t,f}(a_{t,f}, s) + r_{j,p}(a_p, s), \quad (6)$$

where β is the reward bias, $r_{j,\eta}(a_\eta, s)$, $r_{j,o}(a_o, s)$, $r_{j,t,f}(a_{t,f}, s)$, and $r_{j,p}(a_p, s)$ represent the rewards of offloading action, order action, resource block selection, and transmission power, respectively.

1) Reward of Offloading Gateways: When IoT devices and gateways are evenly distributed with similar transmission loads, all the gateways should serve approximately the same number of IoT devices. Also, in the case of the gateway not supporting a certain protocol, the gateway should offload those devices to other gateways. Hence, gateways should have the ability to offload some IoT devices' requests to others if it has more than what it is capable of. We propose a metric, the resource occupation ratio, to measure the capability of each gateway to serve or offload IoT devices' requests. Specifically, the occupancy ratio is defined as,

$$\eta_j = \sum_{n\in N_j} \frac{T_n \times F_n}{T \times \Delta}, \quad (7)$$

where T and Δ are the total time and frequency units, and T_n and F_n are the transmission duration and bandwidth for IoT n. The occupancy ratio information is expected to be shared between gateways for making offload decisions. Meanwhile, we also define the average occupancy ratio as,

$$\eta_{\text{avg}} = \frac{1}{J} \sum_{n\in N} \frac{T_n \times F_n}{T \times \Delta}. \quad (8)$$

which will serve as the threshold for determining whether to afford more or offload.

To balance limited resources in the network, it is expected the action should lean toward encouraging a light-loaded gateway to serve more IoT devices while offloading IoT devices from a heavy-loaded to other vacant gateways. Therefore, the reward is designed by comparing the difference between

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

the current resource occupancy and the other gateway or the averaged ratio, which is defined below,

$$r_{j,\eta} = \begin{cases} \beta_\eta(\eta_j - \eta_{j'}), & \text{if offload to gateway } j'; \\ \beta_\eta(\eta_{j'} - \eta_j), & \text{if offload from gateway } j'; \\ \beta_\eta(\eta_j - \eta_{\text{avg}}), & \text{not doing anything.} \end{cases} \quad (9)$$

where β_η is the weighting factor.

2) Reward of Ordering Serving Gateways: Once the offloading decision is made, each gateway needs to determine the service order for all its served IoT devices. Rather than randomly selecting different IoT devices to serve, we can set the order by their protocols, e.g., Wi-Fi > ZigBee > Bluetooth. For simplicity, we assume all the same types of IoT devices have the same priority. Hence, the priority of the n-th decision provides a negative reward can be written as,

$$\zeta_{j,n} = \left(\sum_{n' \in Q} \texttt{diff}(n, n') \right)^2, \quad (10)$$

where Q is the queue containing all the IoT devices that have not been scheduled yet. The $\texttt{diff}(\cdot)$ function compares the IoT priority between device n and n'.

$$\texttt{diff}(n, n') = \begin{cases} l(n') - l(n), & \text{if } l(n') > l(n); \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where $l(n)$ indicates the level of priority. In our case, Wi-Fi, ZigBee, and Bluetooth devices have $l(n)$ as $3, 2$, and 1, respectively. Note that this service order will not conflict with their QoS requirement. For example, some IoT devices must be served with a higher priority to satisfy their latency QoS requirement, which can be denoted as,

$$\mathbf{I}\left[T_n(a_n) < T_{n,QoS}\right], \quad (12)$$

where $\mathbf{I}[c] = 1$ when the condition c is satisfied. The reward of each IoT decision can be written as follows,

$$r_{j,o} = \sum_{n=1}^{N_j} \beta_T \cdot \mathbf{I}\left[T_n(a_n) < T_{n,QoS}\right] + \beta_\zeta \cdot \zeta_{j,n} \quad (13)$$

where β_T and β_ζ are the weighting factor.

3) Reward of Selecting Resource Blocks: Each resource block has to be carefully chosen on both time and frequency domains to avoid collisions as well as enhance utilization efficiency. Specifically, for each block being removed, i.e., becomes unavailable for scheduling in the next time slot, the negative reward increases to penalize the selection in the future. For the n-th round of assignment, the available choice of resources is $|D_{n-1}|$ and will become $|D_n|$ after the scheduling. We defined $\kappa_{j,n}$ as the resource impact at the n-th round as,

$$\kappa_{j,n} = |D_{n-1}| - |D_n|. \quad (14)$$

Besides, the resource blocks assigned to IoT devices might have been shared with other devices served by other gateways. To indicate whether the requirement is fulfilled or not, we denote

$$\mathbf{I}\left[\rho_n(a_n) < \rho_{n,QoS}\right]. \quad (15)$$

Therefore, the overall reward of selecting resource blocks can be expressed as

$$r_{j,t,f} = \sum_{n=1}^{N_j} \beta_\rho \cdot \mathbf{I}\left[\rho_n(a_n) < \rho_{n,QoS}\right] + \beta_\kappa \cdot \kappa_{j,n}, \quad (16)$$

where β_ρ and β_κ are the weighting factor.

4) Reward of Transmission Power: Although a higher transmission power yields a lower packet error and a longer transmission range, it consumes more energy and might interfere with other devices that use the same time-frequency resource. Hence, IoT devices should always use the necessary transmission power to save energy and avoid interfering with other devices. The reward of each transmission power can be written as,

$$r_{j,p} = -\beta_p \sum_{n=1}^{N_j} w_{p,n} \times p_n(t), \quad (17)$$

where $w_{p,n}$ is the weight of transmission power of n-th device, and β_p is the weighting factor for reward.

IV. Cascaded MARL Design

A. Design Overview

With the previous MDP design, the performance of a state-action pair is usually mapped to a Q-value to be stored in the Q-table, indicating the reward of the corresponding action. By searching for the maximum reward, the system moves to a state with a better performance. Unfortunately, our targeted scenario is expected to consist of a large number of heterogeneous IoT devices with a significantly huge state space and action space. The above MDP is extremely hard to solve even with neural network-based solutions, such as RL, not to mention the scalability in a dynamically changing environment. Therefore, we propose to develop a cascaded RL framework to alleviate the keep-growing size of the neural network while providing scalability simultaneously.

B. Cascaded Multi-Agent Reinforcement Learning Network

As discussed in previous sections, the major technical challenge of using RL to allocate limited resources is the exponentially increased states and actions. Instead of using a single RL network to determine the joint action, we propose a cascaded Multi-Agent Reinforcement Learning (MARL) network as shown in Fig. 3, where the action will be determined in different dimensions, i.e., sub-network modules, to correspond to the sequential sub-actions in the action space. Specifically, we design 4 sub-networks for 4 actions defined in Sec. III-C. At the beginning of the transmission, the system observes the current state and then begins a sequential round of predictions for all the IoT devices.

• **Offloading Gateway Sub-Network.** The first sub-network collects both the channel estimation and the current resource occupancy as the observed state $\mathbf{s}_{t,1}$. It also records the same

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

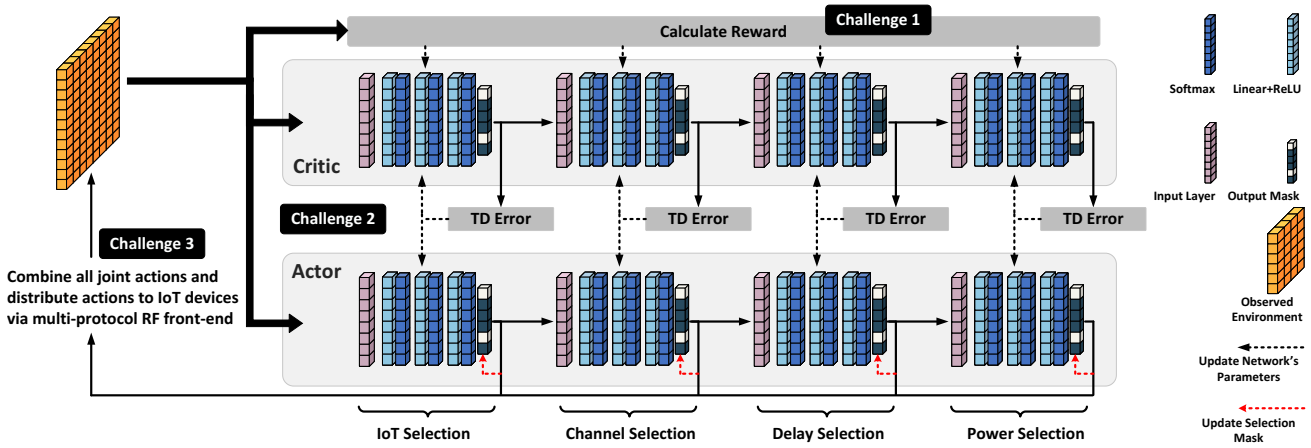


Fig. 3: Proposed Cascaded MARL Framework.

information from other non-serving devices during the time duration t as extra information $i_{j,1}$. By inputting $\mathbf{s}_{t,1}$ and $i_{j,1}$ into this sub-network, the offloading decision on serving IoT devices will be yielded out. Note that only one round of offloading decisions will be made for all IoT devices to keep the system stable.

• **Ordering Serving Gateways Sub-Network.** We use a sub-network to decide the order of the serving IoT device to connect the output of the offloading gateway sub-network. This sub-network takes both the observed state, $s_{j,2}$ and the information of its serving devices, denoted as $i_{j,2}$. The idea is to choose those who use more resources and/or need higher-priority transmission to fulfill their QoS requirement. By using the Softmax layer before the output layer, the service order is determined by the output probability.

• **Resource Selection Sub-Network.** The third sub-network collects and combines the service order from the last sub-network as its observed state, $s_{j,2}$. It also uses the resources and QoS requirement as extra information, $i_{j,2}$. The sub-network then goes through N_j rounds, each of which yields the time-frequency pair for IoT devices to use.

• **Power Selection Sub-Network.** The last sub-network collects the RSSI and packet error rate as its observed state, $s_{j,3}$, and the minimum and maximum transmission power of the assigned IoT as the extra information $i_{j,3}$. The output determines the transmission power for the assigned IoT on the next transmission slot.

C. Using Actor-Critic Algorithm

The traditional policy gradient approach updates models based on full episode samples or the termination criteria. Unfortunately, wireless communication never meets its end stage. Therefore, we must adopt an architecture that can update its network in real time. Meanwhile, multiple sub-networks decide different actions based on the interaction with the environment, for which the state-action pair has to be decoupled as the network updates. The A2C model separates the network into two agents: the Actor, which is a policy-based network with a policy function, and the Critic, which is a value network with a value function. The state-action pair is decoupled since the Critic calculates the reward of making a specific action, and the Actor updates its parameters based on the return from the Critic instead of the environment in which it interacts. Since the state and action space in our design are deemed to be huge, we plan to choose Advantage Actor-critic (A2C) as the architecture to be used in our cascaded RL to bootstrap the training efficiency.

V. Performance Evaluation

This section will evaluate the proposed framework via experiments conducted on a real SDR platform.

A. Experimental Setting

1) Hardware/Software Setup: We use multiple Universal Software Radio Peripherals (USRP), including X310 and B210 models as the RF front-end. The USRPs are driven by USRP Hardware Driver (UHD) v4.2.0.0 and connected to an Ubuntu 20.04 desktop via a 10 Gbps cable to provide the high-speed link while controlling the USRP X310 and through a USB 3.0 cable for USRP B210. We develop the entire backend while supporting instantaneous control over radio devices. The application front-end uses GNU Radio v3.10.5 to build the PHY and MAC layer control functionality for all the protocols. The backend is implemented on Python 3.8 with the Redis database for collecting and storing data and triggering actions. The ZigBee devices are developed on the SimpleLink CC2652R device produced by Texas Instruments and programmed by Code Composer Studio. The Wi-Fi and Bluetooth devices are implemented in GNU Radio with USRP as the RF front-end.

2) RL Model Training.: We carry out our framework design on PyTorch 1.13.1 with NVIDIA CUDA toolkit 11.6. The training process is separated into two parts, assignment and learning. In the assignment part, the gateway follows the ϵ -greedy rule to achieve the balance between exploration and exploitation. We set the start of ϵ-greedy parameter, ϵ_{max}, to be 0.9 and the decay rate, ϵ_{decay} as 500, and the end of the parameter is $\epsilon_{\text{min}} = 0.05$. The ϵ-greedy threshold is defined as follows,

$$\epsilon_{\text{threshold}} = \epsilon_{\text{min}} + (\epsilon_{\text{max}} - \epsilon_{\text{min}}) \times \exp\left(\frac{-n}{\epsilon_{\text{decay}}}\right), \qquad (18)$$

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

where n is the number of epochs. The gateway also resets its observation and labels all the tags/channels available for all the IoT devices. After the initialization, gateways' sub-networks start deciding their sub-actions. After making each decision, gateways record the observed state, decisions, rewards, and next state as the memory. After collecting enough memory, predefined as batch size, the sub-networks start the learning process based on randomly sampled memory from their memory buffers.

3) Wireless Environment Setups: We conduct the experiment in a $10m \times 10m$ indoor environment as shown in Fig. 4, in which 2 Wi-Fi, 27 ZigBee, and 4 Bluetooth devices are deployed. Three gateways serve those devices by assigning actions, which are located on each side of the room. The RL station exchanges information (e.g., environment dynamics and actions) with gateways via wired UDP connections.

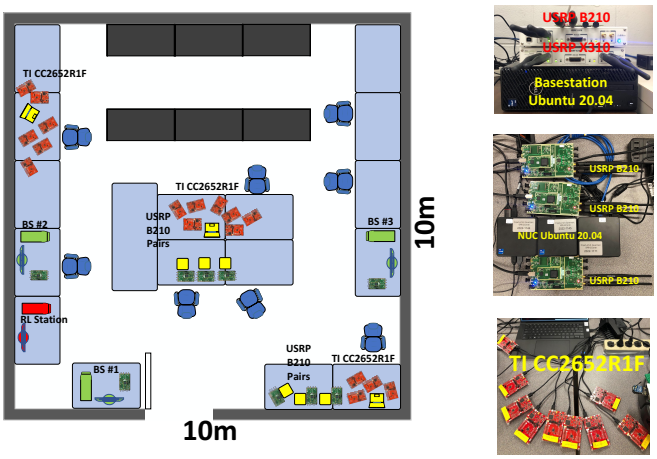


Fig. 4: Experiment Topology

In our experiment, each gateway maintains a resource pool with 20 MHz frequency band and 10ms time duration, where a single block is sliced into a 2 MHz×1ms resource unit. The detailed frequency usage is as follows.

	Channel	Center Frequency (GHz)
Wi-Fi	11	2.412
ZigBee	11-14	2.405, 2.410, 2.415, 2.420
Bluetooth	0-6	2.403, 2.406, 2.409, 2.412, 2.415, 2.418, 2.421

TABLE II: Frequency and Channel Usage

B. Experiment Designs

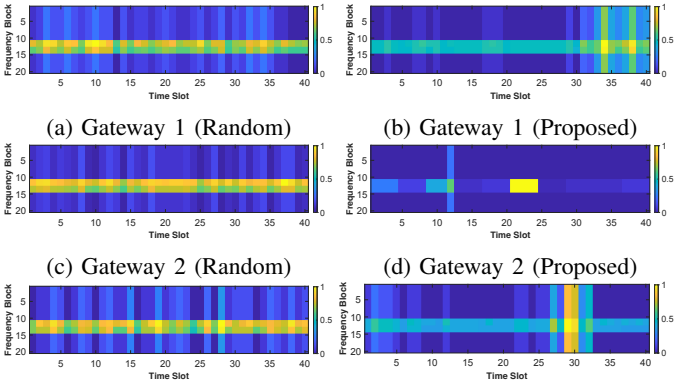
Our design is robust to system failures, for which the cascaded MARL framework can adjust to unexpected errors, e.g., gateway malfunctions can be dealt with by the offloading mechanism as mentioned in Sec. III-D1, and still maintain the best possible resource orchestration strategy. Hence, we consider the following two scenarios and compare them with the random access-based approach.

- Scenario 1. Continue using the previous setting, where all three gateways normally receive packets from every protocol.

- Scenario 2. To best describe system failure, Gateway 1 can only operate with ZigBee protocol, while Gateway 2 and 3 act normally.

C. System Performance Analysis

1) Resource Allocation Comparison.: For *Scenario 1*, the proposed design finds a suitable assignment for all the serving IoT devices without too much CTI from each other as shown in Fig. 5b, 5d, and 5f. On the other hand, the random access-based one fails and assigns the resource blocks randomly, shown in Fig. 5a, 5c, and 5e. When experiencing system failure, the proposed design can adjust the strategy by learning from the uplink contextual information. Fig. 6b, 6d, and 6f show the resource assignment in *Scenario 2* at the end of the training process. Fig. 6b shows that all the frequency blocks have been used for ZigBee due to Gateway 1's malfunction. As a result, the framework offloads all other devices to Gateway 2 and 3 by taking up many available frequency blocks. The random access-based approach, however, cannot change the resource orchestration strategy for supporting Wi-Fi and Bluetooth devices.

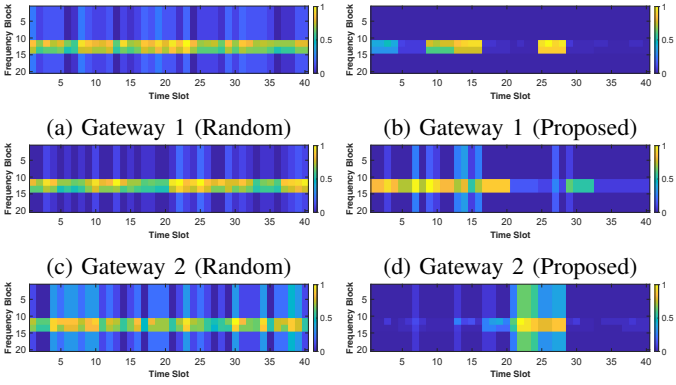


(a) Gateway 1 (Random) (b) Gateway 1 (Proposed)

(c) Gateway 2 (Random) (d) Gateway 2 (Proposed)

(e) Gateway 3 (Random) (f) Gateway 3 (Proposed)

Fig. 5: Adaptive Resource Assignment (Scenario 1)



(a) Gateway 1 (Random) (b) Gateway 1 (Proposed)

(c) Gateway 2 (Random) (d) Gateway 2 (Proposed)

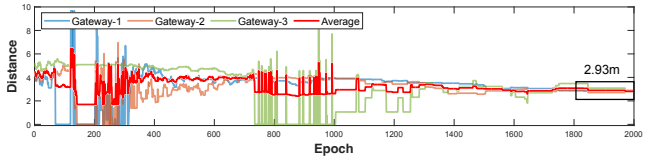
(e) Gateway 3 (Random) (f) Gateway 3 (Proposed)

Fig. 6: Adaptive Resource Assignment (Scenario 2)

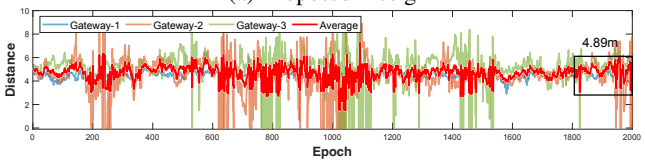
2) Offloading Performance: Benefited by the proposed adaptive resource orchestration, the uneven number of heterogeneous IoT devices will find their closest serving gateways, and thus, each gateway will serve fewer devices to achieve the best performance.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

• **Serving Distances.** Fig. 7 shows the distances between all IoT devices and their serving gateway throughout time. Gateways offload IoT devices to another gateway if they either sense the serving IoT devices have lower RSSI or do not have enough resources to support all the devices. For *Scenario 1*, the average distance between IoT devices and their serving gateways is 2.93m (proposed) and 4.89m (random).



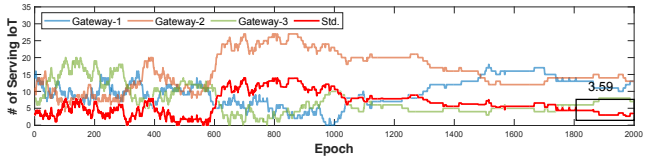
(a) Proposed Design



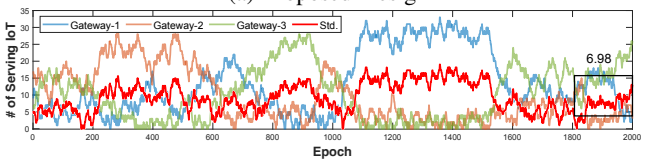
(b) Random Access-based Design

Fig. 7: Serving Distance Comparison

• **Devices Balancing.** Fig. 8 shows the number of serving IoT on each gateway. The proposed approach finally learns a stable state for serving IoT devices in the environment while providing sufficient service. However, the random access-based approach fails to find a stable state and keeps switching IoT devices among all gateways. At the end of the experiment, the standard deviation of the number of serving IoT devices among all gateways is 3.69 and 6.98 in the proposed and random access-based approach, respectively. Note that even though it seems the random access-based approach can serve more devices, the transmission delay and successfully delivered packages are not guaranteed.



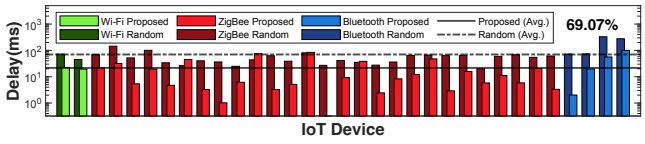
(a) Proposed Design



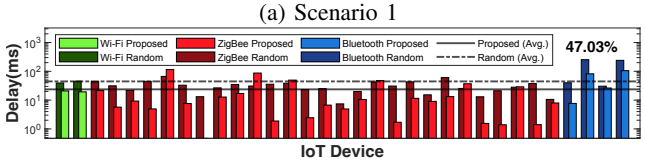
(b) Random Access-based Design

Fig. 8: Number of Serving Devices Comparison

3) Transmission Delay: The transmission delay comes from two reasons: 1) failure to get a resource assignment from the system and 2) failure to be received by gateways due to interference from others. Since the proposed design offloads IoT devices and assigns proper resources for communication, it experiences less failure compared with random access. Fig. 9a and 9b show the delay of all the IoT devices experienced in both scenarios. The proposed system has, on average, 69.07% and 47.03% less delay than the random access-based approach, where Wi-Fi experienced 35.70%, 47.498%, ZigBee experienced 34.04%, 63.37%, and Bluetooth experienced 23.68%, 39.26% less delay in those two scenarios.



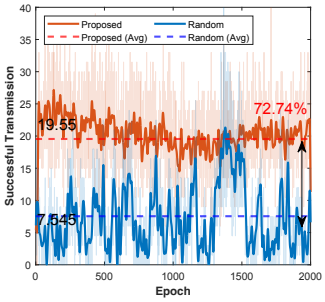
(a) Scenario 1



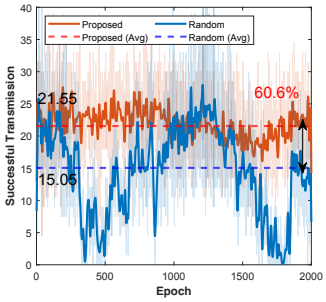
(b) Scenario 2

Fig. 9: Transmission Delay Comparison

4) Successful Transmissions: We deploy a total of 25 heterogeneous IoT devices in both scenarios. In *Scenario 1*, the proposed design has an average 72.74% more successful transmission throughout the experiment than the random access, shown in Fig. 10a. For *Scenario 2*, even if one of the gateways cannot send Wi-Fi and Bluetooth packets, the proposed design learns how to offload those packets to another available gateway and maintains the overall performance. The proposed method achieves 60.6% more successful transmissions than the traditional random access approach when the learning process is converged.



(a) Scenario 1



(b) Scenario 2

Fig. 10: Transmission Successful Comparison

Further, Table III shows the variance of the number of successful transmissions in different scenarios. The variance of using the random access approach is significant in two scenarios. Especially in *Scenario 2*, with a variance of 30.1335, all the IoT devices failed to find a gateway for a stable connection. While using the proposed method, the system maintains a variance of 2.25 and 4.22 in two scenarios, showing the stability of using the proposed approach, given one gateway cannot function normally.

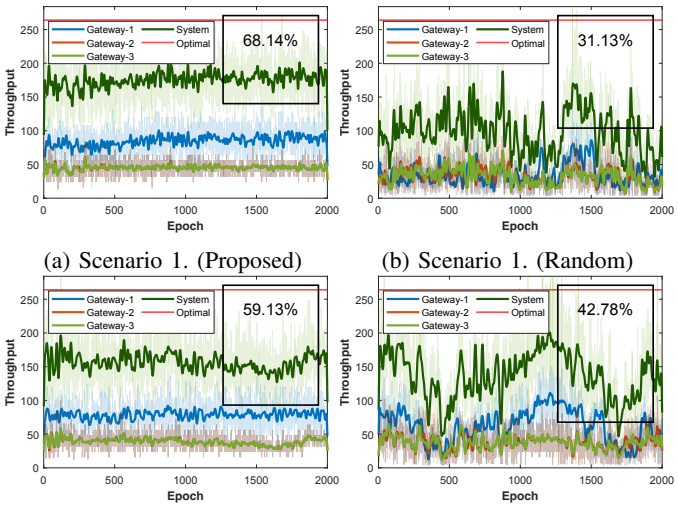
Scenario	1	2
# of Success TX Var. (Proposed)	2.2472	4.2165
# of Success TX Var. (Random)	13.6760	30.1355

TABLE III: Transmission Stability of Serving IoT devices

5) Network Throughput: As shown in Fig. 11a, and 11b, the network throughput remains similar at the first several epochs while the proposed scheme starts to learn better action over time. Eventually, the overall system throughput can achieve

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

up to 68.14% of the optimal throughput using the proposed design, which is 2.19X more than the random access at 31.13% of the optimal throughput. In *Scenario 2*, the throughput decreases in both the proposed scheme and random access. However, the proposed scheme can still achieve up to 59.13%, a 1.38X more than the random access at 42.78%.



(a) Scenario 1. (Proposed) (b) Scenario 1. (Random)

(c) Scenario 2. (Proposed) (d) Scenario 2. (Random)

Fig. 11: System Throughput

D. Training Time Cost Analysis

The system-level time cost mainly consists of two parts: training (including state observation, reward calculation, and model updates) and decision-making by all agents. In Fig. 12, the system-level aggregated time consumption increases with the number of serving IoT devices, in which the decision-making takes less training time. Upon receiving state information, each training epoch takes 0.27, 0.28, 0.30, and 0.36 seconds for 15, 20, 25, and 30 IoT devices, respectively.

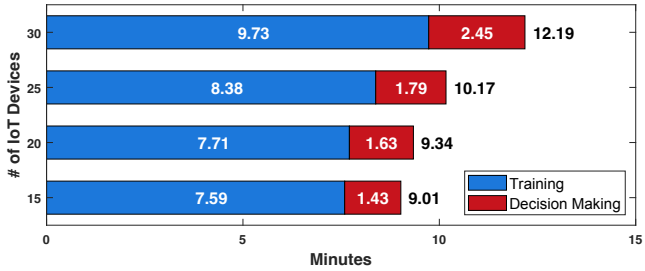


Fig. 12: System-level Aggregate Time Cost (Min.)

VI. Related Works

• **CTI in Heterogeneous Network.** Heterogeneous devices with co-existing protocols suffer from interference from other protocols. The impact of such CTI is inevitable, especially in a dense heterogeneous wireless network [7], [11], [12]. Existing MAC-layer protocols, such as CSMA/CA, cannot help adapt [23]. In the meanwhile, very few research works have considered the fact that the increasing CTI in the highly dense heterogeneous environment cannot be handled by existing protocol-dependent approaches [4], [7]–[10], [17]. Recent works find different ways to mitigate the interference. [19] proposed an interference alignment algorithm to fight against interference in a 5G network. However, not all IoT devices are capable of affording the extra computation overhaul to implement the new algorithm. The transmission power at various small BSs was changed in [2] to manage interference in the self-organizing network. On top of the power control, our proposed method also provides offloading, channel choosing, and delay timing selection. Effective resource orchestration in a dense heterogeneous network has drawn the researchers' attention. In [13], Lin *et. al* focused on large-scale online data analysis and sharing using non-orthogonal multiple access (NOMA) assisted architecture. However, none of the above-mentioned works conducted a real-world experiment to evaluate the performance.

• **Resource Allocation in Dense Heterogeneous IoT Network.** Heterogeneous IoT devices demand different subsets of available resources such as spectrum and power to satisfy diverse and stringent QoS requirements [1], [3], [20]. Rezaei *et.al* in [16] increased the spectral efficiency by using a power domain non-orthogonal multiple access scheme in which the same spectrum is shared among several users. In [24]–[26], spectrum allocation was discussed by formulating the topic into an optimization problem. ML algorithms, including reinforcement learning and federated learning algorithms, are used in several works to actively control the behavior of IoT devices. [14], [21], [22] implemented a Q-learning algorithm to interact with the environment, while [21], [22] only considered the transmission power control and [14] implemented the offloading ability. Our proposed method provides extra channel and delay timing control as well as uses the spectrum resource in a more efficient manner. [15] deployed a federated learning approach to allow multiple devices to train a global model jointly. However, it introduces more computation overhaul to all IoT devices and thus increases the complexity. Deep reinforcement learning was adopted in [18] to achieve the TDD uplink and downlink resource allocation in a 5G HetNet.

VII. Conclusion

This paper proposes a cascaded MARL framework for resource orchestration in highly dense heterogeneous IoT systems. The proposed framework leverages contextual information from the physical layer to make decisions for resource orchestration including IoT device offloading, channel selection, transmission timing, and transmit power. The proposed design renovates the current random access approach to pre-determine adaptive resource orchestration strategy for static highly dense heterogeneous IoT systems. Extensive real-world experiment demonstrates significant network performance enhancement.

Acknowledgment

The work of L. Guo is partially supported NSF under grant CNS-2008049, CCF-2312616, CCF-2427875, and CNS-2431440, and Army Research Office (ARO) under Grant Number W911NF-24-1-0044. The work of X. Zhang is partially supported by NSF under grant CCF-2312617 and CNS-2431439.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Basim KJ Al-Shammari, Nadia Al-Aboody, and Hamed S Al-Raweshidy. Iot traffic management and integration in the qos supported network. *IEEE Internet of Things Journal*, 5(1):352–370, 2017.

[2] Roohollah Amiri, Mojtaba Ahmadi Almasi, Jeffrey G. Andrews, and Hani Mehrpouyan. Reinforcement learning for self organization and power control of two-tier heterogeneous networks. *IEEE Transactions on Wireless Communications*, 18(8):3933–3947, 2019.

[3] Kun Cao, Guo Xu, Junlong Zhou, Tongquan Wei, Mingsong Chen, and Shiyan Hu. Qos-adaptive approximate real-time computation for mobility-aware iot lifetime optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(10):1799–1810, 2018.

[4] Shih Heng Cheng and Ching Yao Huang. Coloring-based inter-wban scheduling for mobile wireless body area networks. *IEEE Transactions on parallel and distributed systems*, 24(2):250–259, 2012.

[5] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K. Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.

[6] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *ACM SIGCOMM Computer Communication Review*, 40(4):159–170, 2010.

[7] Frederik Hermans, Olof Rensfelt, Thiemo Voigt, Edith Ngai, Lars-Åke Nordén, and Per Gunningberg. Sonic: Classifying interference in 802.15. 4 sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 55–66, 2013.

[8] Anwar Hithnawi, Su Li, Hossein Shafagh, James Gross, and Simon Duquennoy. Crosszig: combating cross-technology interference in low-power wireless networks. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. Ieee, 2016.

[9] James Hou, Benjamin Chang, Dae-Ki Cho, and Mario Gerla. Minimizing 802.11 interference on zigbee medical sensors. In *4th International ICST Conference on Body Area Networks*, 2010.

[10] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *The 18th IEEE International Conference on Network Protocols*, pages 305–314. IEEE, 2010.

[11] Kaushik Lakshminarayanan, Srinivasan Seshan, and Peter Steenkiste. Understanding 802.11 performance in heterogeneous environments. In *Proceedings of the 2nd ACM SIGCOMM workshop on Home networks*, pages 43–48, 2011.

[12] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 309–322, 2010.

[13] Kai Lin, Chensi Li, Joel J. P. C. Rodrigues, Pasquale Pace, and Giancarlo Fortino. Data-driven joint resource allocation in large-scale heterogeneous wireless networks. *IEEE Network*, 34(3):163–169, 2020.

[14] Yi Liu, Huimin Yu, Shengli Xie, and Yan Zhang. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Transactions on Vehicular Technology*, 68(11):11158–11168, 2019.

[15] Van-Dinh Nguyen, Shree Krishna Sharma, Thang X. Vu, Symeon Chatzinotas, and Björn Ottersten. Efficient federated learning algorithm for resource allocation in wireless iot networks. *IEEE Internet of Things Journal*, 8(5):3394–3409, 2021.

[16] Atefeh Rezaei, Paeiz Azmi, Nader Mokari Yamchi, Mohammad Reza Javan, and Halim Yanikomeroglu. Robust resource allocation for cooperative miso-noma-based heterogeneous networks. *IEEE Transactions on Communications*, 69(6):3864–3878, 2021.

[17] Anand Prabhu Subramanian, Himanshu Gupta, Samir R Das, and Jing Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE transactions on mobile computing*, 7(12):1459–1473, 2008.

[18] Fengxiao Tang, Yibo Zhou, and Nei Kato. Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5g hetnet. *IEEE Journal on Selected Areas in Communications*, 38(12):2773–2782, 2020.

[19] David Alejandro Urquiza Villalonga, Alejandro López Barrios, and M. Julia Fernández–Getino García. Hardware evaluation of interference alignment algorithms using usrps for beyond 5g networks. In *IEEE EUROCON 2023 - 20th International Conference on Smart Technologies*, pages 394–399, 2023.

[20] Jiafu Wan, Jun Yang, Shiyong Wang, Di Li, Peng Li, and Min Xia. Cross-network fusion and scheduling for heterogeneous networks in smart factory. *IEEE Transactions on Industrial Informatics*, 16(9):6059–6068, 2019.

[21] Jingjing Wang, Chunxiao Jiang, Kai Zhang, Xiangwang Hou, Yong Ren, and Yi Qian. Distributed q-learning aided heterogeneous network association for energy-efficient iiot. *IEEE Transactions on Industrial Informatics*, 16(4):2756–2764, 2020.

[22] Liang Xiao, Hailu Zhang, Yilin Xiao, Xiaoyue Wan, Sicong Liu, Li-Chun Wang, and H. Vincent Poor. Reinforcement learning-based downlink interference control for ultra-dense small cells. *IEEE Transactions on Wireless Communications*, 19(1):423–434, 2020.

[23] Bo Yang, Xuelin Cao, Zhu Han, and Lijun Qian. A machine learning enabled mac framework for heterogeneous internet-of-things networks. *IEEE Transactions on Wireless Communications*, 18(7):3697–3712, 2019.

[24] Qing Yang, Ting Jiang, Norman C. Beaulieu, Jingjing Wang, Chunxiao Jiang, Shahid Mumtaz, and Zheng Zhou. Heterogeneous semi-blind interference alignment in finite-snr networks with fairness consideration. *IEEE Transactions on Wireless Communications*, 19(4):2472–2488, 2020.

[25] Zhenyu Zhou, Xinyi Chen, Yan Zhang, and Shahid Mumtaz. Blockchain-empowered secure spectrum sharing for 5g heterogeneous networks. *IEEE Network*, 34(1):24–31, 2020.

[26] Binnan Zhuang, Dongning Guo, Ermin Wei, and Michael L. Honig. Large-scale spectrum allocation for cellular networks via sparse optimization. *CoRR*, abs/1809.03052, 2018.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on July 13,2025 at 15:47:08 UTC from IEEE Xplore. Restrictions apply.