Random Access Control In Massive Cellular Internet of Things: A Multi-Agent Reinforcement Learning Approach

Jianan Bai

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Prof. Lingjia Liu, Chair

Prof. Yang (Cindy) Yi

Prof. Haibo Zeng

December 10, 2020

Blacksburg, Virginia

Keywords: Random access; internet-of-things, multi-agent Reinforcement Learning

Random Access Control In Massive Cellular Internet of Things: A Multi-Agent Reinforcement Learning Approach

Jianan Bai

(ABSTRACT)

Internet of things (IoT) is envisioned as a promising paradigm to interconnect enormous wireless devices. However, the success of IoT is challenged by the difficulty of access management of the massive amount of sporadic and unpredictable user traffics. This thesis focuses on the contention-based random access in massive cellular IoT systems and introduces two novel frameworks to provide enhanced scalability, real-time quality of service management, and resource efficiency. First, a local communication based congestion control framework is introduced to distribute the random access attempts evenly over time under bursty traffic. Second, a multi-agent reinforcement learning based preamble selection framework is designed to increase the access capacity under a fixed number of preambles. Combining the two mechanisms provides superior performance under various 3GPP-specified machine type communication evaluation scenarios in terms of achieving much lower access latency and fewer access failures.

Random Access Control In Massive Cellular Internet of Things: A Multi-Agent Reinforcement Learning Approach

Jianan Bai

(GENERAL AUDIENCE ABSTRACT)

In the age of internet of things (IoT), massive amount of devices are expected to be connected to the wireless networks in a sporadic and unpredictable manner. The wireless connection is usually established by contention-based random access, a four-step handshaking process initiated by a device through sending a randomly selected preamble sequence to the base station. While different preambles are orthogonal, preamble collision happens when two or more devices send the same preamble to a base station simultaneously, and a device experiences access failure if the transmitted preamble cannot be successfully received and decoded. A failed device needs to wait for another random access opportunity to restart the aforementioned process and hence the access delay and resource consumption are increased. The random access control in massive IoT systems is challenged by the increased access intensity, which results in higher collision probability. In this work, we aim to provide better scalability, real-time quality of service management, and resource efficiency in random access control for such systems. Towards this end, we introduce 1) a local communication based congestion control framework by enabling a device to cooperate with neighboring devices and 2) a multi-agent reinforcement learning (MARL) based preamble selection framework by leveraging the ability of MARL in forming the decision-making policy through the collected experience. The introduced frameworks are evaluated under the 3GPP-specified scenarios and shown to outperform the existing standard solutions in terms of achieving lower access delays with fewer access failures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Internet of things (IoT), as an emerging and promising paradigm, is expected to interconnect a vast number of machine type devices in the next decade [1, 2, 3]. The prevalence of IoT roots in both advanced large-scale parallel data processing techniques [4] and massive connectivity provided by wireless communication systems [5]. Indeed, IoT or massive machine type communications (mMTC), has became an important application regime for the fifth generation (5G) and beyond 5G wireless communication techniques [6, 7]. Meanwhile, providing reliable and low-latency coverage to massive number of devices is challenged by the sporadic and unpredictable nature of IoT/mMTC traffics and the massive amount of devices.

IoT traffic pattern differs from that of conventional human-type communications mainly in three aspects. First, usually only a small subset of devices will be active with a small transmission payload. Second, IoT devices, such as sensors, may have strict requirements on energy consumption and battery life and they cannot keep connected for a long time. Third, data freshness is crucial for some delay-sensitive applications, such as self-driving and remote surgery. These three features, coupled with the massive amount of IoT devices, renders the deployment of IoT networks very difficult in practice. There have been a lot of efforts made towards addressing this problem. From data transmission perspective, massive antenna based techniques provide high spatial multiplexing capability to serve multiple end devices simultaneously using the same time-frequency resource block with improved spectral

efficiency. These data transmission techniques, e.g., massive multiple-input multiple-output (MIMO), are being applied in real wireless systems. On the other hand, the design of access control, which could suffice to manage the increased access intensity in an efficient way, is still a critical research topic.

In existing wireless communication systems, the wireless connection between a device and a base station is usually established by contention-based random access, which is a four-step handshaking process initiated by the device through sending a randomly selected preamble sequence to the base station. After successfully receiving and decoding the preamble, the base station will send the transmission grant and scheduling information in the remaining steps. LTE systems are configured with 64 orthogonal preamble sequences, while 10 of them are reserved for contention-free random access (not in the scope of this work). When two or more devices send the same preamble at the same time, preamble collision happens and the access attempts could be failed with a very high probability. Since the devices perform random access in a completely uncoordinated manner, i.e., random preamble selection, the access capacity of the systems is severely limited. For example, when 54 preambles are available, the maximum expected number of collision-free preamble transmissions is only around 20 [8]. However, the standard solutions, including access barring and back-off schemes, cannot even achieve this number. Clearly, as the number of simultaneous random access attempts increase in IoT systems, the current random access framework cannot provide satisfactory performance due to the high collision probability.

To tackle the limitations of contention-based random access in massive IoT networks, there have been many efforts from both industrial standards and academia. The 5G New Radio (NR) includes grant-free transmission solutions, in which the devices transmit a preassigned preamble sequence together with the metadata (e.g., pilot sequence) and the payload data. The base stations can detect user activities using compressed sensing techniques and the

handshaking process can be avoided. However, grant-free random access requires redesign of the random access protocol and the user activity detection can be very computational demanding. Although there have been many research works showing superior performance of grant-free random access over contention-based random access, the effectiveness of it in real wireless systems is still not clear. Low-complexity and reliable solutions to grant-free random access is still under development. Thus, in this thesis, we keep our focus on the conventional contention-based random access, while we introduce two novel schemes that greatly reduce the access delays with much fewer access failures:

First, we introduce a congestion control framework, which enables the neighboring devices to cooperate without exchanging sensitive information. To be specific, a cluster head (CH) will be selected in each IoT cluster to distribute the random access permissions to the cluster members based on their delay tolerances. In this way, the contention between devices in the same cluster can be avoided and the access permissions can be allocated based on the quality of service requirements. Second, a multi-agent reinforcement learning (MARL) based preamble selection framework is developed to address the low resource efficiency in contention-based random access. To be specific, the MARL agents, possessed by the CHs, are jointly trained to make preamble selection decisions for the participating devices. The policies are updated towards achieving the lowest access failure probability by using the collected experience. Meanwhile, we design an action-branching based reinforcement learning structure to tackle the exponentially growing action space by processing each action dimension separately.

Notably, in the thesis, the research work mainly reproduces my submitted journal paper [9].

Chapter 2

Multi-agent Reinforcement Learning Meets Random Access in Massive Cellular Internet of Things (IoT)

2.1 Abstract

Internet of things (IoT) has attracted considerable attention in recent years due to its potential of interconnecting a large number of heterogeneous wireless devices. However, it is usually challenging to provide reliable and efficient random access control when massive IoT devices are trying to access the network simultaneously. In this paper, we investigate methods to introduce intelligent random access management for a massive cellular IoT network to reduce access latency and access failures. Towards this end, we introduce two novel frameworks, namely local device selection (LDS) and intelligent preamble selection (IPS). LDS enables local communication between neighboring devices to provide cluster-wide cooperative congestion control, which leads to a better distribution of the access intensity under bursty traffics. Taking advantage of the capability of reinforcement learning in developing cooperative multi-agent policies, IPS is introduced to enable the optimization of the preamble selection policy in each IoT clusters. To handle the exponentially growing action space in IPS, we design a novel reinforcement learning structure, named branching actor-critic, to

ensure that the output size of the underlying neural networks only grows linearly with the number of action dimensions. Simulation results indicate that the introduced mechanism achieves much lower access delays with fewer access failures in various realistic scenarios of interests.

2.2 Introduction

Internet-of-things (IoT) envisions a hyper-connected world with massive interrelated devices, machines, and objects. It has the potential to reform both industrial manufacture and life services with enhanced efficiency, autonomy, and robustness [1, 2, 3]. Meanwhile, IoT is also one of the main drives of the development of the fifth-generation (5G) cellular technology. To support IoT in existing cellular networks, the 3rd Generation Partnership Project (3GPP) has released two low power wide area network (LPWAN) radio technology standards, including the narrowband IoT and the enhanced machine-type-communications (eMTC) [10]. According to Statista, 50 billion IoT devices are expected to be in use around the world by the year 2030 [11] envisioning a truly massive IoT/MTC scenario for future cellular networks.

IoT networks differ from traditional cellular networks in supporting heterogeneous traffic patterns, imposing high requirements on energy efficiency, and serving various delay-sensitive applications (e.g., remote surgery and self-driving), just to name a few. As massive devices are deployed to achieve massive IoT/MTC, the access intensity increases accordingly and it becomes challenging to provide efficient and reliable access control especially when the system resources are limited. Consequently, the quality-of-service (QoS) could degrade significantly under massive IoT/MTC scenarios. On the other hand, due to the heterogeneity of the IoT services, the QoS handling in massive IoT/MTC systems becomes exceedingly complicated.

These features render the access control in massive IoT/MTC networks a critical topic.

2.2.1 Motivations

In this paper, we design a new access control framework for massive cellular IoT/MTC to provide performance improvement in the following three aspects. *Scalability:* Under the massive connectivity of IoT networks, which is completely different from human-to-human (H2H) communications, it is important to ensure the access control framework could scale in large-scale wireless systems with relatively small control overhead. *Real-time QoS Management:* Since the traffics in IoT networks are usually heterogeneous, the delay-constrained/sensitive applications could be subject to high latency or access failures under bursty traffics. Therefore, it is important to provide QoS management in real-time. *Resource Efficiency:* While the access intensity is growing rapidly in IoT networks, the available radio resources will remain limited with a much slower growing rate. Accommodating the demand of increasing access requests with the fixed system resources becomes important for massive IoT/MTC networks.

In general, there are two kinds of access strategies for IoT networks: centralized and distributed [12]. Due to the unique features of massive IoT/MTC networks, in this paper, we will focus on exploring distributed access control mechanisms to reduce the associated control overhead while providing enhanced scalability and flexibility. In distributed access strategies, since the controllers are distributed in the IoT network leading to the geological proximity, they can respond more promptly to the devices while consuming less communication resource. Furthermore, a distributed controller will serve much fewer devices than a central controller making it possible to deploy low complexity control policies. On the other hand, since distributed controllers can only have access to the partial information of the

whole network, it becomes challenging to design algorithms to maximize the overall network performance. Towards this end, we consider the multi-agent reinforcement learning (MARL) framework to develop model-free distributed control policies and update the control decisions (actions) in real-time in response to the changing environment and the states of the devices.

Due to the fact that the random access process takes place before the IoT devices are connected to the eNBs, cooperation among different IoT devices is usually difficult to be achieved during this process. Accordingly, modern cellular networks utilize random preamble selection (see details in Section 2.3.1) which suffer from the low resource efficiency. For example, assuming 54 preambles are available in a random access opportunity (RAO), only around 20 preambles are expected to be transmitted without collisions when applying optimal congestion control parameters with random preamble selection according to the study in [8]. Severe performance degradation during random access could be encountered in massive IoT/MTC systems when the access intensity exceeds the random access channel (RACH) capacity leading to lower resource efficiency. In this paper, we introduce cooperation among IoT devices to assist the random access process to improve efficiency. The benefits of introducing cooperation are mainly in the following two aspects: First, through cooperation the involved IoT devices can avoid colliding with each other. Second, more system resources can be reserved to IoT devices with low delay tolerance traffic, i.e., delay-sensitive traffic. It is important to note that the delay tolerance of a device may change in different RAOs according to the latency requirements. Since it is expected that the IoT devices will be densely distributed and thus can enjoy the short-distance communication links between adjacent devices in massive IoT/MTC networks, it is natural to explore how the cooperation between neighboring devices (more formally, devices in the same cluster) could facilitate random access.

2.2.2 Related Works

Due to its significance, random access strategies have been extensively studied to address the access congestion problem. Access class barring (ACB) is an efficient scheme to alleviate congestion by limiting the number of simultaneous access attempts from wireless devices [8]. The modified version of ACB, extended access barring (EAB), is introduced as a baseline solution in the 3GPP standards for IoT/MTC traffics [13]. Furthermore, dynamic access barring (DAB) enables the eNBs to monitor the network loading and dynamically update the access barring parameters to further improve the efficiency of access barring [14]. The performance analysis and comparison of different access barring schemes can be found in [15]. Furthermore, back-off mechanisms are implemented in real-world communication systems [16]. New preamble design and detection methods are also explored to handle the increasing access requests [17].

To tackle the massive connectivity in the upcoming massive IoT/MTC networks, various schemes have been developed. Taking advantage of the device-to-device (D2D) communications, cluster-based random access was introduced to improve energy efficiency and reduce preamble collision [18, 19, 20, 21]. Meanwhile, various techniques are developed to provide efficient clustering algorithms in IoT networks [22]. To improve the success rate of grant-free random access, a super preamble structure containing multiple preambles is introduced and theoretically analyzed in [23]. Learning algorithms are also explored to show performance improvement in various scenarios of interests [24].

Reinforcement learning is a learning category between supervised learning and unsupervised learning [25, 26, 27, 28]. Instead of requiring explicit training labels, which are usually expensive in real-world applications, reinforcement learning enables an agent to directly interact with real environments and learn from the experiences. Through exploration and

exploitation, the agent gradually updates its policy to achieve higher rewards. Using neural networks as function approximators, deep reinforcement learning algorithms have been shown to achieve human-level control in numerous applications [29, 30, 31]. Reinforcement learning algorithms have also been successfully applied to wireless communication systems recently [32, 33, 34, 35, 36, 37, 38]. For random access, the authors in [39] designed a reinforcement learning-based eNB selection algorithm to enable MTC devices to choose eNB in a self-organizing fashion. In [40], reinforcement learning algorithms are used to control the access barring parameters. A MARL-based framework was introduced in [12] to enable a participating device to choose the number of preambles to transmit in each RAO.

2.2.3 Contributions

In this paper, a cluster-based random access framework is developed, where IoT devices in the same cluster can perform D2D communications at a low cost. A cluster head (CH) is selected in each IoT cluster to collect information from devices and make control decisions for random access. To protect data privacy, only non-sensitive information (e.g., task delay) are allowed to be shared to the CHs. The main contributions of this paper can be summarized as follows:

- We introduce a novel congestion control framework, named local device selection (LDS). In LDS, a CH selects participating devices from an IoT cluster by considering the delay tolerance of all active devices in the current RAO. Taking advantage of the cooperations with neighboring devices, a device with lower delay tolerance is granted with a higher chance to participate and hence receive higher QoS protection.

- A MARL-based preamble selection mechanism, intelligent preamble selection (IPS), is introduced. During each RAO, a participating IoT device will transmit preambles

selected by a MARL agent, which is possessed by the CH. The agents are trained to minimize access failures with reduced access delay.

- To address the exponentially growing action space in the IPS scheme, a novel reinforcement learning structure, named branching actor-critic (BAC), is developed by distributing high-dimensional decisions into multiple action branches and hence successfully compresses the output size of the underlying neural networks.

The remainder of the paper is organized as follows. In Section 2.3, we introduce the random access process and the system model. In Section 2.4, we introduce the proposed access control framework by incorporating LDS and IPS for congestion control and preamble selection, respectively. In Section 2.5, we introduce MARL and actor-critic algorithms. Furthermore, BAC is formally introduced and applied to the underlying MARL framework of IPS. In Section 2.6, we show the experiment results. We conclude the paper in Section 2.7.

2.3 Random Access and System Model

We consider a massive cellular IoT/MTC network as shown in Fig. 2.1, where densely located IoT devices are distributed in each cell. When receiving a task, which is generated at random by some traffic models, an IoT device becomes active and seeks for the access grant in subsequent RAOs until success or failure. The detailed random access procedures and the associated challenges are discussed in Section 2.3.1. The MTC traffic models, the state transitions of IoT devices, and the performance evaluation metrics are introduced in Section 2.3.2.

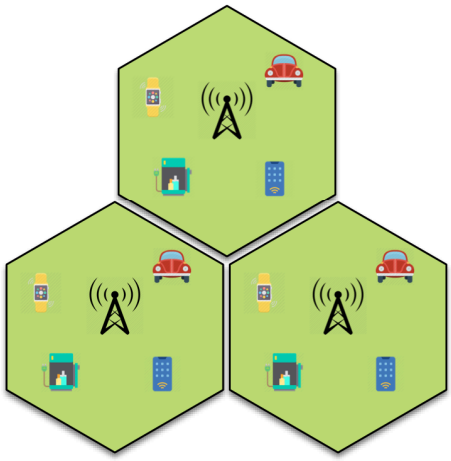


Figure 2.1: Cellular IoT network.

2.3.1 Random Access

Random access is a handshaking process initiated by a device to establish the wireless connection with an eNB. 3GPP supports two types of random access, including contention-free random access and contention-based random access [41]. Contention-free random access requires the existence of a powerful infrastructure to perform centralized random access control, leading to high control overhead and poor resource utilization (orthogonal random access resources are reserved for selected devices). Therefore, contention-free random access is undesirable in the scenario of massive connectivity in IoT networks. Instead, contention-based random access is considered in this paper.

As depicted in Fig. 2.2, the random access process includes four steps: In step 1, a participating device sends a randomly selected preamble on the physical random access channel (PRACH) to the eNB. The preambles are constructed using Zadoff-Chu sequences to guarantee the orthogonality of preambles. Existing LTE systems are configured with 64 preambles with 10 of them reserved for contention-free random access. Upon reception of the preamble, the eNB sends a RAR message to the device in step 2. In step 3,the RAR message will be replied with a message containing its identity information to request radio resources and

scheduling information for data transmission. Finally, in the last step, the eNB broadcasts a message to specify the terminal identity of successfully accessed device.

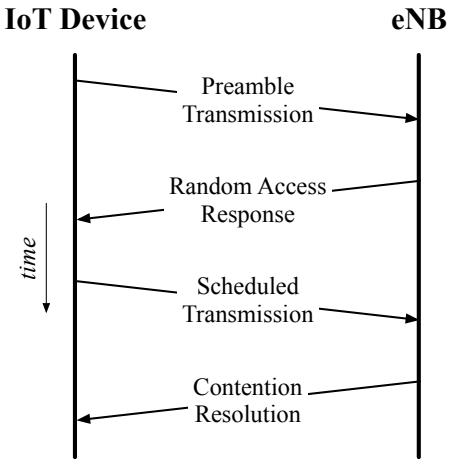


Figure 2.2: The procedure of contention-based random access.

The performance of random access would be detrimentally harmed by preamble collisions, which occur when two or more devices sends the same preamble in step 1 simultaneously. Upon collision, it is assumed that the eNB cannot decode the received preamble, so that all collided devices cannot receive the RAR message within the RAR window. As a result, collided users have to wait a back-off time for preamble re-transmission in another RAO [41]. Clearly, preamble collisions could result in a larger access delay and extra energy consumption due to the preamble re-transmissions and hence affect the QoS of devices [42, 43]. More severely, access failure occurs when a device cannot successfully access after finishing a pre-determined number of preamble transmissions, namely *preambleTransMax*, or the delay exceeds a maximum delay constraint. In LTE, the RAO could occur once every subframe up to once every other radio frame. When the access intensity exceeds the random access capacity (the number of successful access attempts expected in each RAO) of a wireless system, the collision probability increases and the performance degrades (e.g., lower access success probability and larger access delay). In this regard, evenly distributing the access

attempts among different RAOs and increasing the access capacity are two critical directions in random access management.

2.3.2 System Model

3GPP specifies two different traffic models (as shown in Table 2.1) for the evaluation of MTC systems under different access intensities [41]. Traffic model 1 can be considered as a realistic scenario in which MTC devices access the network uniformly over a period of time in a non-synchronized manner. Traffic model 2 characterizes an extreme scenario in which a large amount of MTC devices access the network in a highly synchronized manner with a burst of traffics.

Table 2.1: MTC traffic models.

Characteristics	**Traffic model 1**	**Traffic model 2**
Number of devices	1000,3000,5000, 10000,30000	1000,3000,5000, 10000,30000
Arrival distribution	Uniform distribution	Beta$(3, 4)$ distribution
Distribution period	60 seconds	10 seconds

Once receiving a communication task, an IoT device becomes active and initiates the random access process in the incoming RAOs to request a wireless connection with the eNB. Under congestion control (e.g., ACB and back-off scheme), the device can participate in a RAO only if some required conditions are satisfied. If the device succeeds in a random access attempt, the wireless connection with the eNB will be established for data transmissions. However, if the device cannot successfully access within *preambleTransMax* or the maximum delay constraint, it has to discard the outdated data or report the access problem to upper layers. This process can be summarized as the state transition diagram in Fig. 2.3.

In 3GPP, five performance indicators of RACH capacity are considered [41], including colli-

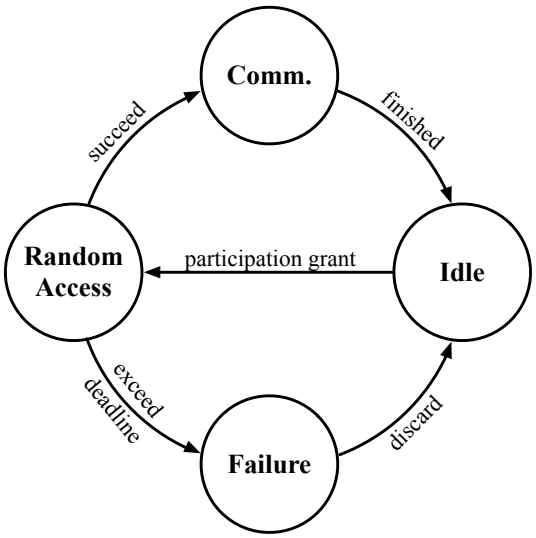


Figure 2.3: State transition diagram.

sion probability, access success probability, statistics of number of preamble transmissions, statistics of access delay, and statistics of simultaneous preamble transmission. In this paper, we mainly focus on the **access success probability** and the **access delay** as the metrics for performance evaluation.

2.4 Enhanced Access Control by LDS and IPS

As analyzed in Section 2.3.1, to improve the network performance in random access, two directions are of interest: **congestion control** and **preamble selection**. In congestion control, the access attempts from devices are supposed to be evenly and properly distributed among different RAOs to eliminate the effects of bursty traffics. In preamble selection, the access capacity can be improved if the participating devices can select preambles in a more cooperative way. Furthermore, it is preferable to provide QoS protection to the devices by both congestion control and preamble selection. Towards this end, we introduce a novel access control framework, including both LDS and IPS, which provides the aforementioned

features.

2.4.1 Local Device Selection

Existing cellular networks implement congestion control mainly through access barring schemes, e.g., ACB and EAB, to limit the number of simultaneous access attempts from wireless devices. In ACB, the eNB broadcasts an ACB factor, $P_{\text{ACB}} \in [0, 1]$, to the devices. In each RAO, a backlogged device generates a random number sampled from a uniform distribution, $\mathcal{U}(0, 1)$, and it can participate only if the generated number is smaller than P_{ACB}. Otherwise, the device needs to wait for a barring time T_{ACB} to resume the ACB process. In EAB, the devices are distributed into 16 access classes numbered from 0 to 15. The first ten access classes are for normal traffics. Access class 10 is for emergence calls and the remaining access classes are reserved for high priority tasks. Access barring schemes rely on the central control by eNB, which broadcasts barring parameters (e.g., ACB factor or access class), while devices work in a non-cooperative way.

Due to the fact that massive IoT/MTC devices are usually densely located, many of them lie in a close proximity and they are naturally clustered. Taking advantage of the short-distance communication links, it is naturally and widely assumed that devices in the same cluster can perform D2D communications at a low cost. Here, we introduce a novel congestion control scheme, named LDS, by utilizing the geological proximity between IoT devices. Unlike in many researches, where a CH can collect the data from intra-cluster devices and send the aggregated message to the eNB to improve energy efficiency, we do not allow sharing data between devices due to data privacy issues. Instead, we enable the CHs to collect some non-sensitive state information, called delay tolerance level (DTL), from the local devices. The details of LDS is presented as follows:

Assuming device i, where $i \in \{1, 2, \cdots, N\}$ with N being the total number of devices in this cluster, is active at the t-th RAO, it will first send its current DTL $e_i^{(t)}$ to the CH. The DTL is defined as the remaining time before the deadline of the current task. Noting as $\mathcal{S}^{(t)}$ the set of all active devices, the CH calculates a participation control factor as

$$c_i^{(t)} = \begin{cases} \frac{\exp\left(-\omega_i^{(t)} e_i^{(t)}\right)}{\sum_{j \in \mathcal{S}^{(t)}} \exp\left(-\omega_j^{(t)} e_j^{(t)}\right)}, & i \in \mathcal{S}^{(t)} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where $\omega_i \in \mathbb{Z}^+$ is the priority level of device i. Note as $\mathbf{c}^{(t)}$ a N-dimensional vector with entries being $c_i^{(t)}$'s. Notice that $\mathbf{c}^{(t)}$ is a probability distribution with $c_i^{(t)}$ being the probability of choosing device i as a participating device at the t-th RAO. The CH will select K devices as participating devices by sampling without replacement according to $\mathbf{c}^{(t)}$. Here, K is a system parameter broadcast by the eNB to restrict the maximum number of participating devices in an IoT cluster. By using LDS in a RAO, the CH selects the participating devices in a cooperative way: devices with lower delay tolerance are selected with a higher chance. Consequently, dynamic QoS handling is provided to all devices to reduce access failures.

Throughout the paper, we assume that a CH is one-hop away from the cluster members. The local outband D2D communication is performed in a time-division or frequency-division manner as in [21]. That is, there is no intra-cluster channel contention involved in the LDS phase. Meanwhile, we assume that all nodes are honest and they send out correct information.

2.4.2 Preamble Selection

In existing cellular systems, each participating device sends a randomly selected preamble to the eNB. The wireless connection is established if the eNB can successfully receive and

decode the preamble. However, when preamble collision occurs, the collided devices may fail to receive the access grant and have to wait for another RAO after a back-off period. Although this scheme is easy to implement and requires little control overhead, it has two drawbacks. First, since all preambles are randomly selected, it suffers from poor resource efficiency. As shown in [8], with a given number of available preambles, M, the expected number of collision-free preamble transmissions is maximized when there are M participating devices and the maximum value is

$$c(M) = \frac{1}{\log\left(\frac{M}{M-1}\right)} \left(1 - \frac{1}{M}\right)^{\frac{1}{\log\left(\frac{M}{M-1}\right)}-1}. \tag{2.2}$$

That is, for example, assuming 54 preambles are available, only around 20 collision-free preamble transmissions are expected during a RAO. Compared with H2H communications, IoT/MTC is supposed to serve much larger number of devices using limited random access resources. The random preamble selection framework may become unaffordably resource-inefficient for IoT/MTC communications. Second, there is no QoS handling and all participating devices have the same priority in this process. Indeed, to avoid access failure, devices with emergent tasks may access with higher priority. However, the QoS handling cannot be implemented through central control, since the random access devices have not been connected to the eNBs.

To incorporate QoS handling and improve resource efficiency, we introduce two new features to the preamble selection process. First, the participating devices are allowed to use multiple preambles in a RAO and the number of preambles to transmit (referred as **greedy level** in this paper) is decided based on the task emergence. Using multiple preambles has been explored in [23], while [12] uses a MARL-based algorithm to dynamically adjust the greedy levels for each device. Second, instead of random preamble selection, which results in low resource efficiency, a participating device can decide which preambles to use before

transmission. The two features are realized by a MARL-based preamble selection framework called IPS. To be specific, the devices in an IoT cluster share a MARL agent (possessed by the CH). After collecting the required information from devices, the CH will make preamble selection decisions and inform the selected devices which preambles they should transmit in the current RAO. The MARL agents are trained to minimize the access failure probability and access delay. The details of IPS will be discussed in the next section.

2.4.3 Discussions

In LDS and IPS, we exploit the geological proximity between adjacent IoT devices and enable D2D communications in IoT clusters. The IoT clusters can be generated by various clustering algorithms, e.g., Low Energy Adaptive Clustering Hierarchy (LEACH) [44]. By collecting related information (DTL in this paper), the CHs make access control decisions that provides real-time QoS handling to the devices. In LDS, devices with larger DTL have higher chance (priority) to participate in a RAO. In IPS, the priority handling is realized by enabling devices to use different greedy levels. Since D2D communications with short link distance usually operates at a low cost, the introduced framework maintain the feasibility in large-scale IoT systems. For certain applications, where the devices cannot afford the cost of training and running a MARL agent, LDS can be a standalone solution for congestion control. The proposed control framework is summarized as in Algorithm 1.

Algorithm 1 Access control in an IoT cluster

```
for the t-th RAO do
    on the CH:
        collect the current DTLs, e_i^(t), from all active devices i ∈ S^(t)
        calculate the participation control factors, c_i^(t), according to Eq. (2.1)
        select K participating devices by sampling without replacement based on c^(t)
        run IPS algorithm and send the preamble selection results to the devices with participation grants
    on each device i:
    if receive participation grant then
        transmit preambles based on the IPS result
    end if
end for
```

2.5 MARL In Random Access

2.5.1 Multi-Agent Reinforcement Learning

MARL is a machine learning paradigm designed to solve an optimal policy in a Markov game, which can be represented as a four-dimensional tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$. Here, $\mathcal{S}$ is the state space, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$ is the joint action space with $\mathcal{A}_i$ being the action space of the i-th agent, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition function mapping from a state-action pair to the next state, and $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2 \times \cdots \times \mathcal{R}_N$ is the joint reward function with $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ being the reward function for the i-th agent. An policy, $\pi : \mathcal{S} \to \mathcal{A}$, is optimal if it maximizes the discounted future reward at every time step t. The discounted future reward is given by $R_t = \sum_{\tau=t}^{T} \gamma^\tau r_\tau$, where T is the time horizon, r_τ is the immediate reward at time step τ, and $\gamma \in (0, 1]$ is the discount factor. In many real-world applications, however, an agent can only receive a partial observation $o_i \in \mathcal{O}_i$ of the environment state $s \in \mathcal{S}$, such that the agents may develop different policies, i.e., $\pi_i : \mathcal{O}_i \to \mathcal{A}_i$.

Fig. 2.4 illustrates MARL in IPS (a partially-observable environment). At one time step, each agent receives an observation from the environment and selects an action to respond

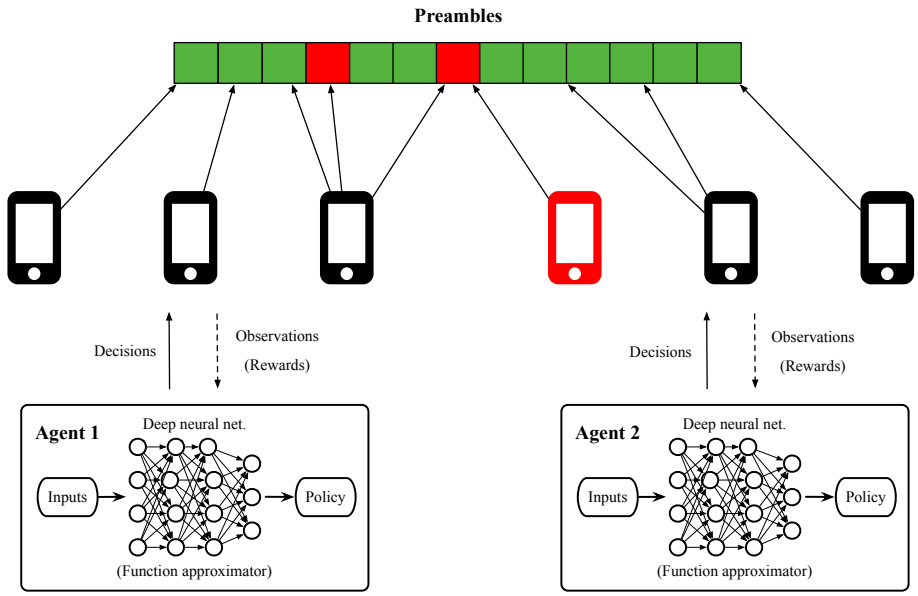


Figure 2.4: Multi-agent reinforcement learning in IPS.

according to the current policy. The policy information is recorded in one (or multiple) function approximator(s), which is usually a deep neural network. The actions of all agents will jointly change the environment to the next state. Meanwhile, the agents receive rewards (possibly different between agents) from the environment. After recording enough experiences, an agent can train its neural network(s) to maximize the discounted future reward by applying selected reinforcement learning algorithms. The agents in MARL can be either competitive or cooperative depending on the design of reward functions.

2.5.2 Actor-Critic and Action Branching

Actor-Critic

Actor-critic is one of the reinforcement learning frameworks[45], which employs an actor (policy generator) and a critic (policy evaluator) to jointly train the policy. The actor network, parameterized by θ, generates the policy $\pi_\theta(a|o)$, which is a probability distribution

of action $a \in \mathcal{A}$ (here, we only consider discrete action spaces) given the observed state $o \in \mathcal{O}$. The critic network, parameterized by w, generates a value function $q_w(o, a)$, which measures the performance of using an action, a, under a given state, o, to evaluate the policy generated by the actor. In practice, the value function is usually an action-value function or an advantage function. The actor network is trained to generate an optimal policy, in terms of maximizing the objective

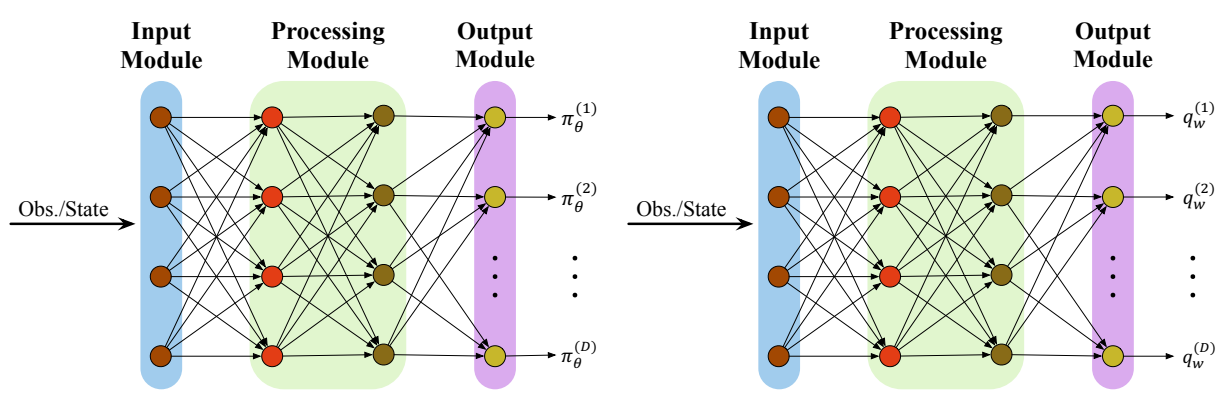
$$J(\theta) = \mathbb{E}_{(o,a)\sim\mathcal{D}}[\pi_\theta(a|o) q_w(o, a)], . \tag{2.3}$$

where $\mathcal{D}$ is the experience replay buffer.

Meanwhile, the critic network is trained to get a better estimation of the value function. Thus, the parameters w is updated to minimize the temporal-difference (TD) error

$$L(w) = \mathbb{E}_{(o,a,r,o')\sim\mathcal{D}} \left[\left(y_w(r, o') - q_w(o, a) \right)^2 \right], \tag{2.4}$$

where $y(r, o') = r + \gamma \max_{a'} q_w(o', a')$, r is the immediate reward, and s' is the next state after performing the action, a.



(a) Actor network (b) Critic network

Figure 2.5: Underlying neural network structures for IPS.

Successful implementation of reinforcement learning requires a moderate action space size. When the action space becomes exceedingly large, training is usually intractable, since it becomes difficult to calculate the expectation in (2.5) accurately with limited number of trials. However, an action in IPS corresponds to a combination of arbitrary preamble indices, so that the action space grows exponentially with the number of preambles available in the resource pool. To deal with the explosive growth of action space, we employ a novel reinforcement learning structure called BAC, which decomposes a multi-dimensional action into multiple sub-actions. The idea of action branching was first discussed in [46] for Q-learning and showed considerable performance improvement. As shown in Fig. 2.5, the output module of the actor network consists of multiple branches, each of which generates the sub-policy for a sub-action. Similarly, in the critic network, each branch generates the value function for a sub-action. Since the input module and the processing module are shared among all branches, a common latent representation of the input is embedded to facilitate the coordination between different branches.

To be specific, consider $a = (a_1, a_2, \cdots, a_D)$ a D-dimensional action in the action space $\mathcal{A}$. For each sub-action dimension $d \in \{1, 2, \cdots, D\}$, the actor network and the critic network generate the sub-policy $\pi_\theta^{(d)}(a|s)$ and the corresponding value function $q_w^{(d)}(a, s)$ from d-th branch, thus each branch only process a reduced action space, which is easier to deal with. Correspondingly, the training objectives of the actor and the critic are supposed to leverage all action dimensions, i.e.,

$$J(\theta) = \mathbb{E}_{(o,a)\sim\mathcal{D}} \left\{ \sum_{d=1}^{D} \pi_\theta^{(d)}(a_d|o) q_w^{(d)}(o, a) \right\}. \tag{2.5}$$

and

$$L(w) = \mathbb{E}_{(o,a,r,o')\sim\mathcal{D}} \left[\sum_{d=1}^{D} \left(y_w^{(d)}(r, o') - q_w^{(d)}(o, a) \right)^2 \right], \tag{2.6}$$

where $y_w^{(d)}(r, o') = r + \gamma \max_{a'} q_w^{(d)}(o', a')$.

In BAC, the output size of the underlying neural networks increases linearly (instead of exponentially) with the action dimension. Thus, the BAC structure keeps the tractability of the IPS mechanism, even with a relatively large number of preambles. It is worth mentioning that the proposed structure is widely applicable in wireless communications systems, when the control decisions can be structured as multi-dimensional actions.

2.5.3 Multi-Agent BAC in IPS

To implement IPS in an IoT network, each CH possesses a BAC agent. After collecting the required information and selecting the participating devices through LDS, the BAC agent performs preamble selection for each participating device sequentially and send the selection results to them. The action, observation, and reward function are defined as follows:

Action

Noting as M the number of preambles available in each RAO, an action in IPS is represented as a M-dimensional tuple $a = (a_1, a_2, \cdots, a_M)$, where a_m, for $m = 1, 2, \cdots, M$, is a binary decision on whether to use the m-th preamble (e.g., $a_m = 1$ indicates using the preamble). Notice that the action space $\mathcal{A}$ contains 2^M actions, but a sub-action space, $\mathcal{A}^{(m)}$, only has two entries. Notice that one execution of the BAC agent only generates the preamble selection result for a single participating device. To select the preambles for all participating devices in the cluster, the BAC agent needs to execute K times. Meanwhile, to distinguish different participating devices in the cluster, the index of the participating device k is incorporated in the design of observation.

Observation

The observation can be represented as a tuple $o = \left(\{e_i\}_{i=1}^{K}, k\right)$. Here e_i is the DTL of the i-th participating device and K is the maximum number of participating devices in an IoT cluster. Meanwhile, as we discussed in the design of actions, $k \in \{1, 2, \cdots, K\}$ indicates that the selected preambles are used by the k-th participating device to distinguish different participating devices. By incorporating $\{e_i\}_{i=1}^{K}$ and k in the observation, the participating devices in the same cluster can cooperate and avoid colliding with each other.

Reward

The reward is designed to balance the trade-off of competition and cooperation between different agents. On the one hand, an agent gets a positive reward if it successfully access in a RAO. On the other hand, the agent gets punished if it uses additional random access resources (preambles). Therefore, the reward function is given by

$$r = \mathbb{I}\{\text{successful access}\} - \alpha \cdot \sum_{m=1}^{M} a_m, \tag{2.7}$$

where α is a pre-determined parameter. We note that the first term in (2.7) is set to one if there is at least one preamble being successfully received and decoded.

To see why the above designs formulate a Markov game, we start by defining the state, s, which is represented as $\{e_i\}_{i \in \mathcal{D}}$, where e_i represents the DTL of device i, and $\mathcal{D} = \{1, 2, \cdots, N\}$ is the set of all devices with N being the number of devices. Notice that the DTL of an idle device can be set to a large value of infinity and we note it as δ. By dividing the devices into G clusters, we have $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_G$, where $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_G$ are disjoint subsets representing the devices in different clusters. Meanwhile, the observation

of the g-th agent (a CH), is $o_g = \{e_i\}_{i \in \mathcal{Z}_g}$, where $\mathcal{Z}_g$ represents the devices selected during LDS and it is a subset of $\mathcal{D}_g$ with cardinality $|\mathcal{Z}_g| = K$. The joint action selected by the agents can be denoted as $a = (a_1, a_2, \cdots, a_G)$, where $a_g = \{p_i\}_{i \in \mathcal{Z}_g}$ with p_i representing the preamble selection result of device i. To make the formulation more general to incorporate all devices, we can look at an extended action space, where an action $\overline{a} = (\overline{a}_1, \overline{a}_2, \cdots, \overline{a}_G)$ with $\overline{a}_g = \{p_i\}_{i \in \mathcal{D}_g}$. For device $i \notin \mathcal{Z}_g$, for all g's, p_i means no preamble is selected by default. With the formal definition of state, observation, and extended action, the state transition function can be given straightforwardly. Here, we use an additional subscript t to denote the RAO index. Assuming an extended action $\overline{a}_t$ is made under the state $s_t = \{e_{t,i}\}_{i \in \mathcal{D}}$, and a subset of devices, $\mathcal{X}_t$, successfully accessed and their DTLs are set to δ. For active devices that did not participate or failed in this RAO, their DTLs are updated according to the definition. The last step in state transition is to set the DTLs of the arriving devices at the $(t+1)$-th RAO to the value specified by the task deadlines.

Here, we introduce the underlying neural network structures for both the actor and the critic. After receiving the observation o, the actor (critic) network first processes it by an input module, which is an embedding layer to map the observation to a high dimensional vector that matches the size of the hidden layers. The processing module consists of feedforward layers. The output layer consists of M branches with a single output neuron in each branch. In the actor network, we get a probability $\pi_\theta^{(m)}(a_m|o)$ from each branch by applying the Sigmoid activation function. In the critic network, we use linear activation function and get a state-value, $V_w^{(m)}(o)$, from each branch. After performing the joint action a, the agent receives the reward r, by using which we can calculate the advantage $A_w^{(m)}(o, a) = r - \sum_{m=1}^{M} V_w^{(m)}(o)$ as $q_w^{(m)}(o, a)$. The actor network and the critic network are trained to maximize $J(\theta)$ in (2.5) and to minimize $L(w)$ in (2.6), respectively.

Table 2.2: System Parameters

Parameter	Setting
Periodicity of RAOs	5 ms
Subframe length	1 ms
Available preambles for contention-based random access	$M = 54$
Maximum number of preamble transmissions	$preambleTransMax = 10$
Maximum delay constraint	10 s
Preamble detection probability for the k-th preamble transmission	$1 - \frac{1}{e^k}$
Back-off indicator	$BI = 20$ ms
Barring rate	$P_{\text{ACB}} = 0.3$
Mean barring time in ACB	$T_{\text{ACB}} = 1$ s
Priority levels	$w_i = 1$ for all devices

2.6 Performance Evaluation

We consider a single cell environment, where $N = 30000$ devices (unless otherwise stated) seek to access the eNB based on traffic model 2 in 2.1. To enable LDS and IPS in a clusted IoT scenario, we assume the devices are evenly distributed in 20 clusters. In real systems, the clusters can be created by using dedicated algorithms, e.g., LEACH. Similar to [8], we assume a typical PRACH configuration, $prach\_ConfigIndex = 6$, with some simplifications. The system parameters are summarized in Table 2.2. Since we only consider collisions in random access step 1, the delays in the later steps are not considered.

2.6.1 Access Control Methods

We consider the following access control methods for comparison:

- **Baseline.** No congestion control is implemented. When a device becomes active, it

always attempts to access in all the incoming RAOs until success, finishing *preamble-TransMax* preamble transmissions, or exceeding the maximum delay constraint.

- **ACB&BO**. The default congestion control framework implemented in existing MTC systems. If an active IoT device is barred in a RAO (with probability P_{ACB}), it needs to wait for a random barring time before next attempt, i.e.,

$$T_{\text{barring}} = [0.7 + 0.6 \times \mathcal{U}[0,1)] \times T_{\text{ACB}}. \tag{2.8}$$

  When preamble collision occur, the IoT device is required to wait a back-off time before resuming the random access process. Here we use an uniform back-off scheme, i.e., $T_{\text{BO}} = \mathcal{U}(0, BI)$, due to the existence of the maximum delay constraint.

- **LDS.** As discussed in 2.4.3, LDS can be used as a standalone congestion control solution. In LDS, a CH selects K participating devices to transmit preambles in the current RAO based on their DTLs. Notice that with M available preambles, the expected number of successful preamble transmission is maximized when exactly M preambles are transmitted in each RAO (under random preamble selection). The eNB can broadcast optimal K's to the CHs, such that M IoT device are selected to participate in each RAO. However, in this paper, we assume imperfect information at eNB, and $K = 3$ is used by all CHs. The weights ω_i's in (2.1) are set to 1 for all devices.

- **LDS&IPS.** This is the complete access control framework we designed in this paper. After selecting the participating devices by LDS, the CHs will run the IPS algorithm to perform preamble selection for each participating device in the cluster. The BAC agents are trained under the following settings: The hidden size of the actor/critic networks is 256 with two-layer processing modules. The training batch size is 512.

The learning rate is 10^{-4}. The discount factor γ is 0.9. The parameter α in the reward function (2.7) is set to 0.1. The experience buffer consists of all transitions in a current episode.

2.6.2 Training in IPS

To implement IPS, we need to train the MARL agents, which make preamble selection decisions based on the received observations in each RAO. As mentioned in 2.5.2, the action space in IPS grows exponentially with the number of preambles available in a RAO. For example, when setting the number of preambles $M = 54$, the action space consists of 2^{54} actions. The overwhelmingly large action space makes the training intractable. In this regard, BAC is proposed to distribute a multi-dimensional action policy into multiple sub-action branches. Each branch only generates a sub-policy for one action dimension. In the scenario of preamble selection, each action dimension is a binary decision. In this section, we evaluate the training efficiency of IPS when using multi-agent BAC. A baseline method, implementing IPS using a naive multi-agent actor-critic (AC) algorithm (that is, the AC agents are trained using equations (2.3) and (2.4) without action branching), is also tested and compared the result of BAC. Both of them use feedforward processing modules. To make the baseline method feasible, we assume only $M = 10$ preambles are available with 5000 access devices arriving according to Traffic Model 2.

In Fig. 2.6, we plot the training curve of BAC and AC averaged over ten trials. The training lasts for 500 episodes, each of which includes the experience in the 4000 RAOs. Clearly, BAC achieves much higher final reward (≈ 0.75) compared with AC (≈ -0.1). The convergence of BAC appears at around the 200-th episode with a smaller training variance. We can observe the by using the traditional AC algorithm, the training of IPS becomes extremely inefficient

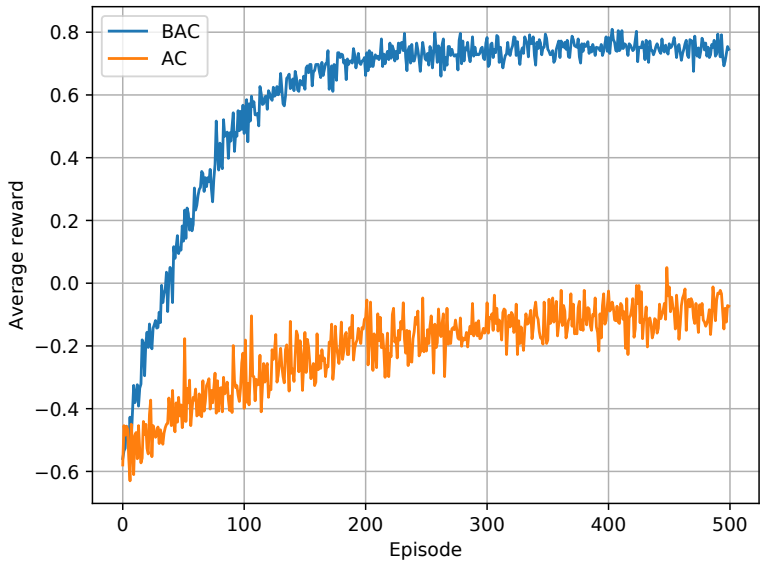


Figure 2.6: The training curves.

even with $M = 10$ preambles (the action space consists of 1024 actions). The use of BAC framework can effectively resolve this problem with a significantly reduced training time.

2.6.3 Performance Under 3GPP-specified Scenario

We start by evaluating the performance of the four access control methods under the most congested scenario specified by 3GPP. In this scenario, a total of 30000 devices access during a period of 10 seconds (2000 RAOs are available in this period). The arrival distributed is Beta$(3, 4)$ as specified in Traffic Model 2 in Table 2.1. In addition to setting $preambleTransMax = 10$, we introduce a maximum delay constraint as 10 seconds. If a device cannot successfully access within 10 seconds with no more than 10 preamble transmissions, it will declare an access failure. Correspondingly, the total simulation time is 20 seconds with a total of 4000 RAOs available. By averaging over ten trials of experiment, the access success probability of the four access control methods are 31.54%, 93.98%, 97.51%,

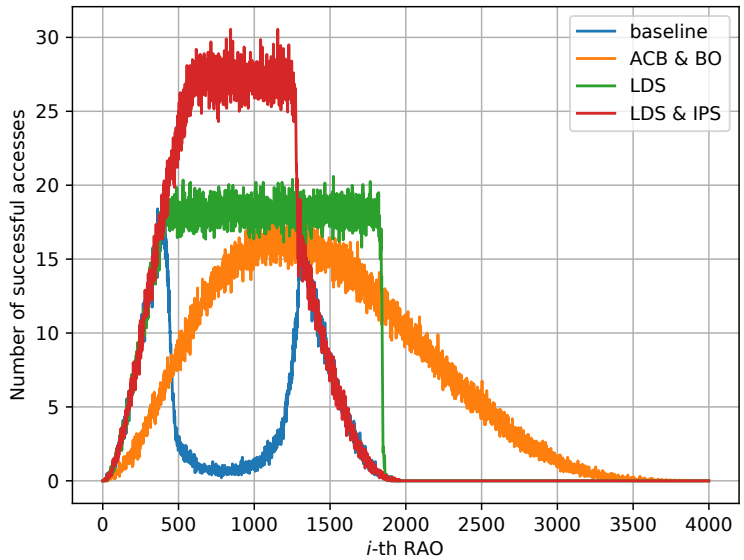


Figure 2.7: Number of successful accesses in each RAO.

and 99.71%, respectively.

In Fig. 2.7, we show the number of successful accesses in each RAO by using the four access control methods. Using the baseline method results in severely congested period between the 500-th RAO and the 1200-th RAO, during which few devices can successfully access. This is because the Traffic Model 2 induces a burst of traffics in this period, preambles sent by participating devices collide with a very high probability when there is no access control. By using ACB and uniform back-off, the access congestion gets alleviated, since the congestion control framework distributes the bursty traffic more evenly across different RAOs, and the access success probability increases significantly compared with the baseline method. However, by using ACB&BO, some devices successfully access after the 3500-th RAO, which results in larger access delay. The LDS method further increases the access success probability to 97.51% while most devices successfully access within the first 2000 RAOs. Additionally, in LDS, the number of successful accesses remain very close to the

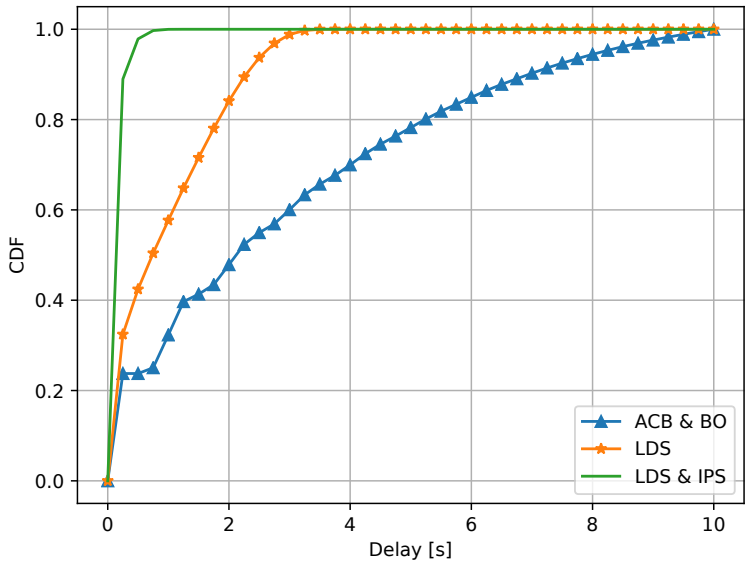


Figure 2.8: Statistics of access delay.

optimal value given in (2.2) from the 500-th RAO to the 1800-th RAO. By incorporating IPS in the access control, the access get granted more promptly. Furthermore, since IPS introduces semi-cooperative and QoS-protecting preamble selection, the maximum number of successfully accesses can exceed the optimal value in random preamble selection.

In Fig. 2.8, we present the cumulative density function (CDF) of the access delay of the devices that successfully accessed by using ACB&BO, LDS, and LDS&IPS. The result of the baseline method is omitted since it has much lower access success probability compared with the other three methods. As we can see in the figure, ACB&BO induces larger access delay for many devices, and this finding is consistent with what we observed in Fig. 2.7. LDS and IPS can significantly reduce the access delay. In LDS, most of the devices can complete random access within 3 seconds. Meanwhile, by using LDS&IPS as the access control solution, random accesses are usually finished in 1 second. As we can see in Fig. 2.7 and Fig. 2.8, LDS and IPS can effectively increase the probability of successful access

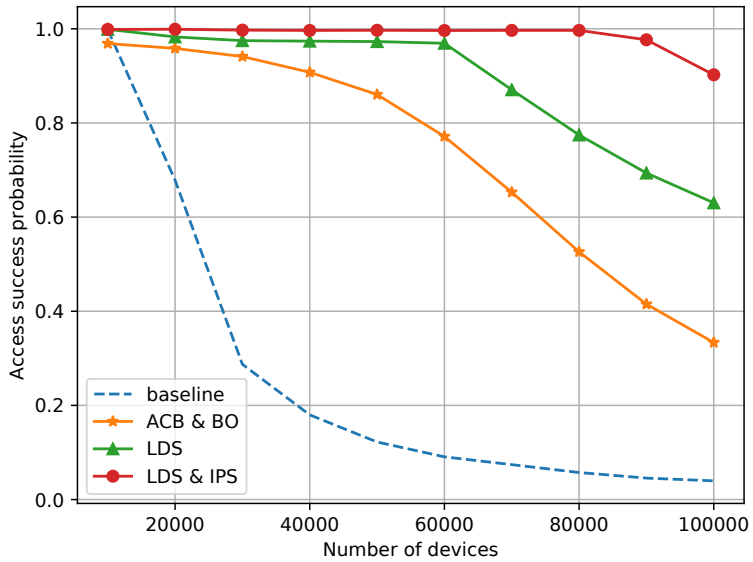


Figure 2.9: Access success probability under different access intensities.

and decrease the access delay under the most heavily congested MTC scenario suggested in 3GPP standards.

2.6.4 Performance Under Different Access Intensities

In Section 2.6.3, we evaluated the performance of the four access control methods under MTC Traffic Model 2 with 30000 devices. However, during the simulation time (20 seconds with 4000 RAOs), a total of 216000 preambles are available which is much more than the number of IoT/MTC traffics we tested. When more devices are deployed, the wireless systems are supposed to experience more severely congested scenarios. Consequently, in this section, we evaluate the performance of the access control methods under different access intensities, i.e., the number of devices $N \in \{10000, 20000, \cdots, 100000\}$. The access success probability and the access delay are presented and compared between different methods.

The access success probability for different access control methods are shown in Fig. 2.9. As we can see, the performance of the baseline method degrades rapidly when the number of devices is larger than 10000, which manifests the importance of congestion control under massive connectivity. The access success probability of ACB&BO decreases as more devices are involved, while it shows better resilience to intense accesses compared with the baseline method. By using LDS and IPS, more accesses can be granted by the eNB under massive connectivity. To be specific, in LDS, most devices can successfully access when the number of devices is 50000, while less then 80% of the devices get the access grants in ACB&BO. In LDS&IPS, the degrade of performance becomes obvious when the number of access devices gets larger than 80000, and it shows the best performance in the four chosen access control methods.

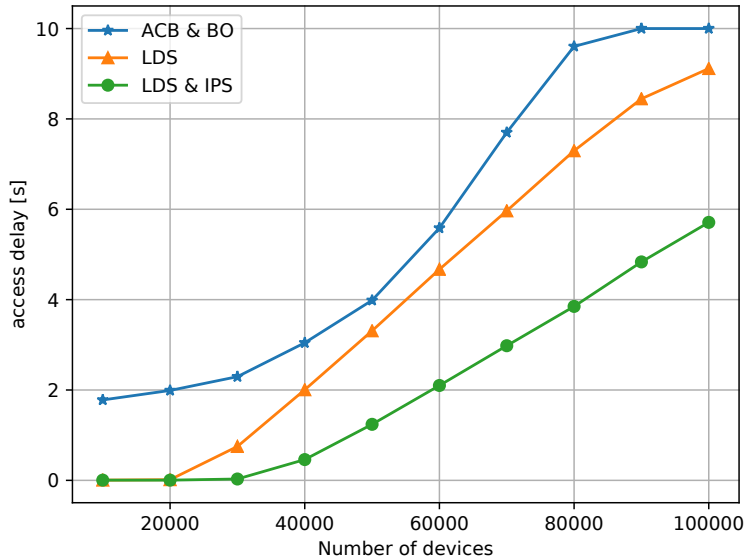


Figure 2.10: 50-th percentile of access delay under different access intensities.

Similar to [8], we evaluate the access delay in terms of percentiles. Here, the P-th percentile access delay is defined as the maximum access delay experienced by the first P% devices with the lowest delay. In Fig. 2.10, we show the 50-th percentile of access delay by using

ACB&BO, LDS, and LDS&IPS under different access intensities ranging from 10000 devices to 100000 devices. Devices controlled by ACB&BO suffer from the highest access delay among the three methods. When the number of devices reach 90000, the 50-th percentile of access delay becomes 10 seconds, which equals to the maximum delay constraint, and it corroborates the observation we obtained in Fig. 2.9: the access success probability is less than 50%. By comparing the curves of LDS and LDS&IPS, we can observe that the advantage of incorporating IPS gets more significant in severely congested networks, since IPS greatly improves the resource efficiency.

2.6.5 Performance Degradation With Clustering Overhead

In reality, forming clusters and reporting to the CHs during a RAO can introduce additional costs. The advantages of clusters comes at the cost of extra delays introduced by clustering. To address this problem, we conduct experiments to explore the impacts of the extra latency. To be specific, instead of considering only the RAO duration, T_{RAO}, which is set as 5 ms, we consider an additional delay, T_{LDS}, to account for the overhead of performing LDS. That is, the effective RAO duration is

$$T_{\text{RAO}}^{\text{eff}} = \begin{cases} T_{\text{RAO}} + T_{\text{LDS}}, & \text{if LDS is enabled,} \\ T_{\text{RAO}}, & \text{otherwise.} \end{cases} \tag{2.9}$$

For example, if the LDS delay is set to 5 ms, then there are only 2000 RAOs during the period of 20 seconds, rather than 4000 in the previous experiments. Under this setup, the overhead of LDS comes at the price of having fewer RAOs. The corresponding results are shown in Fig. 2.11. It is clear to see that the performance improvements of using LDS and IPS reduce and diminish with larger LDS delay. However, when the extra delay is in a

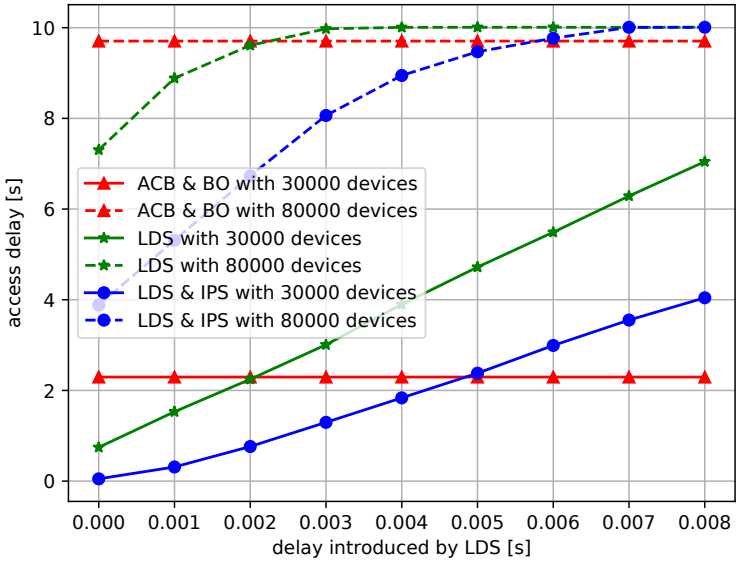


Figure 2.11: 50-th percentile of access delay under different LDS latency.

reasonable range, the introduced frameworks outperforms the existing options.

2.7 Conclusion

In this paper, we developed a novel access control framework, which is able to provide enhanced scalability, real-time QoS management, and resource efficiency during grant-free random access, especially under massive connectivity. The framework relies on the clustering of densely located IoT devices and consists of two control modules, namely LDS and IPS. In LDS, the CHs collect some required information (DTL) from devices in the same cluster and select the devices to participate in the current RAO based on the received DTLs. In IPS, instead of using random preambles, the preamble selection decisions are made by MARL agents to reduce preamble collisions. To tackle the exponentially growing action space in IPS, a reinforcement learning structure, named BAC, is introduced by allocating different

action dimensions to individual action branches. The experiment results show that LDS and IPS can significantly alleviate access failures and reduce access delays in various scenarios. Meanwhile, compared with an actor-critic algorithm without action branching, the BAC algorithm can effectively accelerate the training process and achieve higher final reward under a limited training budget.

Chapter 3

Summary

This thesis investigates a novel contention-based random access control solution to massive cellular IoT systems. Two control frameworks based on local communication and MARL are introduced to address the problem of congestion control and preamble selection, respectively. Moreover, an action-branching based MARL structure is developed to improve the training efficiency in high-dimensional action space. The performance evaluation is performed on various 3GPP-specified testing scenarios for mMTC traffic. The introduced frameworks outperform the conventional ACB and back-off methods and achieves much lower access latency with much higher success probability. Compared with other candidate solutions, e.g., grant-free random access, the proposed method has much better compatibility under the existing wireless protocols and infrastructures.

Chapter 4

Future Works

This thesis focuses on addressing the challenges of random access management in massive cellular IoT networks through two novel frameworks, LDS and IPS. However, for the sake of simplicity of analysis and evaluation, we made some simplifications and assumptions, which can be further explored in future works:

- The interference characterization and throughput of intra-cluster communication can be exploited using tools from stochastic geometry. This could provide a better performance evaluation when considering the clustering overhead. Meanwhile, a better parameter setting (e.g., the number of CHs and the number of participating devices in each cluster) can be expected by optimizing the analytical objectives.
- Allowing some form of communication between different CHs may facilitate better cluster-wise cooperation for improved overall performance. The communication in MARL systems [47] was successfully applied in some applications. The communication can be realized by allowing the CHs to send some additional information to eNB. The eNB will then broadcast the information in the next RAO.

Bibliography

[1] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[2] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[3] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[5] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5g era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, 2016.

[6] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2017.

[7] M. El Soussi, P. Zand, F. Pasveer, and G. Dolmans, "Evaluating the performance of eMTC and NB-IoT for smart city applications," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[8] L. Tello-Oquendo, I. Leyva-Mayorga, V. Pla, J. Martinez-Bauset, J.-R. Vidal, V. Casares-Giner, and L. Guijarro, "Performance analysis and optimal access class barring parameter configuration in lte-a networks with massive m2m traffic," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3505–3520, 2018.

[9] J. Bai, H. Song, Y. Yi, and L. Liu, "Multi-agent Reinforcement Learning Meets Random Access in Massive Cellular Internet of Things (IoT)," *Under Review to Internet of Things Journal*, 2020.

[10] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.

[11] "Number of internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030," Available at https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/, 2020.

[12] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled internet of things: Network architecture and spectrum access," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[13] R.-G. Cheng, J. Chen, D.-W. Chen, and C.-H. Wei, "Modeling and analysis of an extended access barring algorithm for machine-type communications in lte-a networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 2956–2968, 2015.

[14] J.-P. Cheng, C.-h. Lee, and T.-M. Lin, "Prioritized random access with dynamic access barring for ran overload in 3gpp lte-a networks," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 368–372.

[15] W. T. Toor and H. Jin, "Comparative study of access class barring and extended access barring for machine type communications," in *2017 International Conference on*

Information and Communication Technology Convergence (ICTC). IEEE, 2017, pp. 604–609.

[16] Z. J. Haas and J. Deng, "On optimizing the backoff interval for random access schemes," *IEEE Transactions on Communications*, vol. 51, no. 12, pp. 2081–2090, 2003.

[17] X. Lin, A. Adhikary, and Y.-P. E. Wang, "Random access preamble design and detection for 3gpp narrowband iot systems," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 640–643, 2016.

[18] S.-H. Wang, H.-J. Su, H.-Y. Hsieh, S.-p. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2013, pp. 107–111.

[19] T. P. de Andrade, L. R. Sekijima, and N. L. da Fonseca, "A cluster-based random-access scheme for lte/lte-a networks supporting massive machine-type communications," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[20] B. Han, V. Sciancalepore, O. Holland, M. Dohler, and H. D. Schotten, "D2d-based grouped random access to mitigate mobile access congestion in 5g sensor networks," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 93–99, 2019.

[21] M. Gharbieh, A. Bader, H. ElSawy, H.-C. Yang, M.-S. Alouini, and A. Adinoyi, "Self-organized scheduling request for uplink 5g networks: A d2d clustering approach," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1197–1209, 2018.

[22] L. Xu, R. Collier, and G. M. O'Hare, "A survey of clustering techniques in wsns and consideration of the challenges of applying such to 5g iot scenarios," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1229–1249, 2017.

[23] H. Jiang, D. Qu, J. Ding, and T. Jiang, "Multiple preambles for high success rate of grant-free random access with massive mimo," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4779–4789, 2019.

[24] A. Destounis, D. Tsilimantos, M. Debbah, and G. S. Paschos, "Learn2mac: Online learning multiple access for urllc applications," *arXiv preprint arXiv:1904.00665*, 2019.

[25] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[26] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[27] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.

[28] J. Perolat, J. Z. Leibo, V. Zambaldi, C. Beattie, K. Tuyls, and T. Graepel, "A multi-agent reinforcement learning model of common-pool resource appropriation," in *Advances in Neural Information Processing Systems*, 2017, pp. 3643–3652.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[31] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[32] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. C. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 212–217, April 2020.

[33] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1938–1948, 2018.

[34] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Caching placement and resource allocation for cache-enabling uav noma networks," *IEEE Transactions on Vehicular Technology*, 2020.

[35] X. Lin, Y. Tang, X. Lei, J. Xia, Q. Zhou, H. Wu, and L. Fan, "Marl-based distributed cache placement for wireless networks," *IEEE Access*, vol. 7, pp. 62 606–62 615, 2019.

[36] H. H. Chang, L. Liu, and Y. Yi, "Deep echo state q-network (deqn) and its application in dynamic spectrum sharing for 5g and beyond," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2020.

[37] L. Li, L. Liu, J. Bai, H. H. Chang, H. Chen, J. D. Ashdown, J. Zhang, and Y. Yi, "Accelerating model-free reinforcement learning with imperfect model knowledge in dynamic spectrum access," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7517–7528, 2020.

[38] H. Song, L. Liu, H. Chang, J. Ashdown, and Y. Yi, "Deep q-network based power allocation meets reservoir computing in distributed dynamic spectrum access networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 774–779.

[39] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches," *IEEE communications Magazine*, vol. 51, no. 6, pp. 86–93, 2013.

[40] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[41] 3GPP, "Study on RAN Improvements for Machine-type Communications," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 37.868, 08 2011, version 11.0.0.

[42] L. Liu, P. Parag, and J.-F. Chamberland, "Quality of service analysis for wireless user-cooperation networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3833–3842, 2007.

[43] N. Gunaseelan, L. Liu, J.-F. Chamberland, and G. H. Huff, "Performance analysis of wireless hybrid-arq systems with delay-sensitive traffic," *IEEE Transactions on Communications*, vol. 58, no. 4, pp. 1262–1272, 2010.

[44] F. Xiangning and S. Yulin, "Improvement on leach protocol of wireless sensor network," in *2007 international conference on sensor technologies and applications (SENSORCOMM 2007)*. IEEE, 2007, pp. 260–264.

[45] D. Wierstra and J. Schmidhuber, "Policy gradient critics," in *European Conference on Machine Learning*. Springer, 2007, pp. 466–477.

[46] A. Tavakoli, F. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[47] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, pp. 2137–2145, 2016.