IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 March 2024, accepted 10 April 2024, date of publication 15 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3388498

RESEARCH ARTICLE

Dynamic Transmission and Delay Optimization Random Access for Reduced Power Consumption

JIHA KIM[1], YONGHO KIM[1], SEUNGHYUN PARK[2], AND JAYH PARK[1], (Member, IEEE)
[1]Department of Information and Communication Engineering, Myongji University, Yongin-si 17058, South Korea
[2]Department of Computer Engineering, Hansung University, Seoul 02876, South Korea

Corresponding author: Jayh Park (hhpark@mju.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) and Institute for Information & Communications Technology Planning & Evaluation (IITP) funded by Korean Government [Ministry of Science and ICT, South Korea (MSIT)]; in part by the Acquisition Indication-Media Access Control (AI MAC) Protocol on Distributed Machine Learning for Intelligent Flying Base Station under Grant 2022R1A2C2005705; and in part by the Research on Advanced Core Technologies for Wireless Local Area Network (WLAN) Based on eXplainable Artificial Intelligence (AI) under Grant 2021-0-00990. The work of Seunghyun Park was supported by Hansung University.

ABSTRACT Today's communication networks, especially those involving massive Machine-Type Communications (mMTC), encounter challenges like channel congestion and high power consumption. This paper introduces the Dynamic Transmission and Delay Optimization Random Access (DTDO-RA) scheme to address these issues. The presented approach adjusts the backoff indicator (BI) value based on the number of transmissions to optimize the success rate of the RA procedure. This is achieved while concurrently reducing channel congestion and power consumption. The strategy employs reinforcement learning algorithms, notably Q-learning and Deep Deterministic Policy Gradient (DDPG). Methods based on these algorithms play a critical role in fine-tuning the Backoff Indicator (BI) value and the maximum number of preamble transmissions (Max TX), thereby enabling efficient and responsive channel access management. This paper highlights the critical influence of the BI value on preamble transmission delay and overall access delay. The DTDO-RA method effectively manages the number of preamble transmissions during the RA procedure, expanding the BI range in response to heavy RA traffic from UEs. Reducing channel congestion and power consumption is crucial for enhancing the success rate of the RA procedure. The simulation results in this paper evaluate the power consumption and channel congestion in relation to the number of UEs and the number of Random Access Opportunities (RAOs), providing a comparative perspective on the variations in the BI values achieved by the DTDO-RA scheme. The results of the paper provide insights into the achievement of higher network efficiency, reduction of network congestion and lower power consumption.

INDEX TERMS Power consumption, congestion avoidance, massive machine-type communications (mMTC), random access (RA), access delay.

I. INTRODUCTION

The Machine-Type Communication (MTC) requirements are service-specific and outlined in the 3rd Generation Partnership Project (3GPP) standards document [1]. This document distinguishes between MTC and human-to-human communication. The costs and efforts described in this document, as well as the widespread use of communication terminals, are further explored in [2]. The massive MTC (mMTC) environment within 5th-generation new radio (5G NR) communication highlights limitations stemming from the restricted resources of the Physical Random Access Channel (PRACH) and Physical Uplink Shared Channel (PUSCH) in current cellular networks. In response to these limitations,

The associate editor coordinating the review of this manuscript and approving it for publication was Adamu Murtala Zungeru.

the application of reinforcement learning to communication mechanisms has recently been on the rise, paralleling advancements in hardware capabilities. [3], [4], [5]

Typically, User Equipment (UE) utilizes the Grant-Based Random Access (GBRA) procedure, starting with the transmission of a preamble signal to access a channel. This system allocates communication resources and uplink channel access privileges to UEs. However, not all UEs can receive grant successfully. In such cases, the UE is configured to select a random backoff delay from a specified Backoff Indicator (BI) range before attempting retransmission. This random backoff delay plays a pivotal role in preventing channel collisions when multiple UEs try to connect simultaneously. If the UE continues to send preamble requests with fixed delay, power consumption rises, and channel congestion ensues. To mitigate this, the UE modifies the preamble after a pre-defined number of retransmissions to balance reduced collision probability with system efficiency.

In this paper, we propose that a longer BI value and a maximum number of preamble transmissions (Max Tx) should be adopted in situations where a UE transmitting a preamble in a congested channel satisfies certain constraints relative to the number of retransmissions. This approach effectively allows the UE to escape from congested channel conditions and improves the performance of the overall RA procedure. To select the optimal BI value and number of transmissions for performance improvement, we apply the reinforcement learning model with constraints presented in this paper. The agent that performs the reinforcement learning is set as the BS. The BS checks the status of UEs in its coverage and learns to set the BI value and number of transmissions for all UEs appropriately to increase the success rate. We apply Q-learning [6] and deep deterministic policy gradient [7] as learning techniques for this purpose. The main goal of this paper is to reduce power consumption during the RA procedure while improving metrics such as success rate, access delay, and channel congestion without requiring high computational complexity on the UE.

In Section II, we review previously proposed methods to improve the performance of the RA procedure. Section III provides a comprehensive overview of the system model and the reinforcement learning algorithm to be applied in this paper. Section IV describes in detail the method proposed in this paper, the dynamic transmission and delay optimization (DTDO) RA procedure. We also analyze the constraints between the power consumption in the idle and transmission states used in the DTDO-RA procedure, and describe the learning methods of Q-learning and DDPG, which use them to find the optimal BI value and Max Tx. In Section V, we compare metrics such as success rate, average access delay, power consumption, and channel congestion of the RA procedure for different number of UEs, BI values, and Max Tx values under experimental conditions. Section VI concludes the paper by providing insights and conclusions and suggesting areas for future research. The simulation code for 5G Random Access will be available at https://github.com/yaki-toki/5G-NR-RandomAccess-ReinforcementLearning. It was written in Python.

II. RELATED WORKS

This section discusses the fundamental aspects and advancements of GBRA schemes, focusing on their application in Long-Term Evolution (LTE) MTC and 5G NR environments. The discussion starts with [8] and [9], which lays the foundation for GBRA schemes and provides essential simulation results within LTE MTC settings. Building on these foundations, [10] further elaborates on the fundamental procedures and analytical frameworks required for performance evaluation.

An important aspect of this discourse is the efficiency of power consumption during RA procedures, especially in the context of multiple retransmissions. In this vein, [12] proposes an innovative approach using Non-Orthogonal Multiple Access (NOMA) to mitigate collision problems prevalent in Massive MTC (mMTC) environments. This is complemented by the study in [13], which investigates the power consumption dynamics associated with RA schemes in the 5G NR framework and presents experimental results aligned with the 5G standard Key Performance Indicators (KPIs).

The literature also explores the interplay between channel schedules, coverage classes and their impact on device battery life. The pivotal work of [16] extends the findings of [17] by elucidating the relationship between wider coverage, increased battery drain, and latency in IoT devices. This section also addresses the constraints and trade-offs in physical channel design, emphasizing the need for a balanced optimization strategy.

The role of small data transmission (SDT) schemes in minimizing radio resource control (RRC) overhead during RA procedures is explored in [18]. The paper contextualizes the relevance of this technology for IoT devices, which typically transmit smaller data packets, highlighting the challenges of significant RRC overhead and subsequent power inefficiency. In addition, the evolution of UE power saving technologies from 3GPP Release 15 to Release 18 is outlined in [21], with [22] discussing various power saving techniques across multiple domains and their impact on data transmission processes.

The adoption by 3GPP of the Access Class Barring (ACB) method for RA procedures within the mMTC environment is highlighted in [23] and [24]. While this strategy is effective in managing PRACH congestion and prioritizing UEs, it is known to increase access latency. To address this, [25] introduces the Distribution Queue (DQ) mechanism, which provides improvements in access latency and load management.

Finally, advances in preamble detection and the redesign of RA procedures are specifically addressed in [26], [27], [28], [29], [30], and [31]. These studies focus on improving

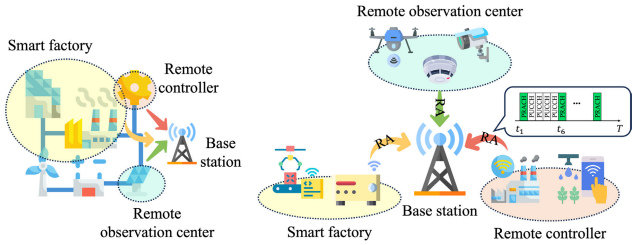


FIGURE 1. Network model.

detection efficiency and system accessibility. For example, [26] highlights the need for new PRACH preamble and protocol designs in 5G NR, recognizing the complexities introduced by varying subcarrier spacing and analogue beam usage. References [27] and [28] contribute designs aimed at reducing collision frequencies, thereby improving success rates in dense network environments. Reference [29] introduces a multi-stage detection process to negate false peaks resulting from difficult channel conditions. In particular, [30] integrates machine learning algorithms such as Decision Tree Classification, Naive Bayes and K-Nearest Neighbor to improve system efficiency, especially in scenarios where multiple devices share the same preamble. Finally, [31] explores the use of device-to-device (D2D) communication to facilitate network access, demonstrating its effectiveness in collision reduction and delay minimization, while identifying the need for further improvements in energy efficiency.

III. SYSTEM ENVIRONMENT

A. SYSTEM MODEL

Fig. 1 illustrates the system environment discussed in this paper. We assume that the base station (BS) is centrally located, and that N_{UE} devices are randomly distributed at locations (x, y) within the cellular boundary, which is a subset of $\mathbb{R}^2$. The system is operated using a remote controller. Remote observation centers and smart factories are connected to the cellular networks and provide data regarding operational state of the system, control outcomes, and environmental inconsistencies under user supervision. Remote observation centers frequently access the network as they are required to transmit data in real-time. In contrast, smart factories engage in network communication only upon the occurrence of specific events or upon receiving instructions from the remote control device. Consequently, smart factory attempts to access the network are sporadic and event-driven.

B. ANALYTICAL MODEL

The probability of preamble transmission, represented by $P_{TX}(i, N_{TX})$, indicates the likelihood that each device will initiate a random access attempt. This probability is influenced by several factors: M_i is the average number of access attempts made by the UE within a certain period; R, representing the number of RAOs; N_{TX}, denoting the number of preamble transmissions; and BI, which stands for the backoff indicator.

$$P_{TX}(i, N_{TX}) = \frac{M_i}{R} e^{-\frac{M_i}{R}} \frac{1}{1 + (N_{TX} \cdot BI)} \tag{1}$$

The successful transmission probability defined by $P_S(i, N_{TX})$ implies the likelihood that each device will successfully complete a random access attempt. As described in equation (2), this probability of success is a function containing the parameter being changed. Specifically, this relies on the average number of access attempts M_i, the available random access slot R, the number of transmissions N_{TX}, and the parameter $B(N_{TX})$ indicating the range of backoff values that the UE selects for transmission. Here, $B(N_{TX})$ is calculated in the form of $BI \times N_{TX}$.

$$P_S(i, N_{TX}) = \frac{M_i}{R} e^{-\frac{M_i}{R}} \cdot B(N_{TX}) \tag{2}$$

The probability defined as $P_I(i, N_{TX})$ represents the likelihood that each UE is in an idle state. This is defined by equation (3). Idle probability is crucial because it reflects the frequency of UEs not participating in transmission, which in turn indicates the operational efficiency of the network. A high idle probability suggests that network congestion can be minimized, while a low idle probability may imply that the frequency of preamble transmission is high due to multiple UEs attempting the RA procedure.

$$P_I(i, N_{TX}) = e^{-\frac{M_i}{R}} (1 - B(N_{TX})) \tag{3}$$

The value $B(N_{TX})$ is normalized between 0 and 1, taking into account the value of the maximum number of transmissions $N_{PT_{max}}$ and the backoff length BI. i.e., in a system environment with a BI value of 20 and a $N_{PT_{max}}$ value of 10, values from 0 to 200 are normalized to values between 0 and 1. In particular, as the backoff interval increases, the success probability of $P_S(i, N_{TX})$ increases, while the idle probability of UEs $P_I(i, N_{TX})$ decreases because the number of successful UEs increases. In addition, for $M_i[n]$ where n is 1, the channel access probability of UEs follows a beta distribution. The equation (4) represents the average number of successful random access attempts for a particular UE, represented by UE_i.

$$M_S(i) = \sum_{N_{TX}=1}^{N_{PT_{\max}}} M_i[n] \frac{M_i}{R} e^{-\frac{M_i}{R}} B(N_{TX}) \tag{4}$$

These probabilities are summed up from 1 to the $N_{PT_{max}}$ to obtain the overall average number of successful random access attempts $M_S(i)$ for UE_i. In summary, $M_S(i)$ represents the overall likelihood that a particular UE_i will successfully complete a RA procedure attempt, considering various parameters such as the number of transmissions, BI, available RAO slots, and so on.

C. TRAFFIC MODEL

The number of UEs initiating RA procedures follows a probabilistic model described by a beta distribution. The beta distribution is suitable for characterizing a wide variety of

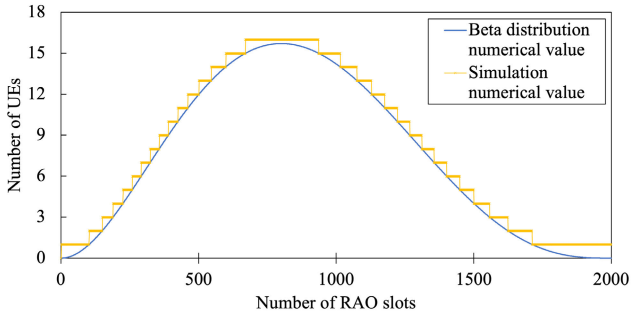


FIGURE 2. Beta distribution (UE: 15,000).

system scenarios because it can adapt to a myriad of shapes. The probability density function (PDF) of this distribution is defined as follows:

$$f(t;\alpha,\beta) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}Beta(\alpha,\beta)}. \tag{5}$$

In this context, t denotes the point in time when a UE attempts the RA procedure within a determined total time T. Here, the parameters α and β act as variables that shape the beta distribution. Furthermore, the following equation (6) represents the fraction of UEs that are activated at a given time t:

$$r(t) = f(t;\alpha,\beta). \tag{6}$$

Here, integrating the beta distribution over time into t_{RAO} allows us to calculate the fraction of UEs initiating the RA procedure at that time. Thus, the value N_{RAO}, which represents the number of UEs attempting access, can be obtained by multiplying the total number of UEs, N_{UE}, by the integral of the beta distribution at that specific time.

$$N_{RAO} = \int r(t)N_{UE}\,dt. \tag{7}$$

Fig. 2 shows the number of UEs accessing each RAO slot in a system environment with 15000 UEs. The approach based on these traffic model allows for a more detailed specification of UE activation timing and serves as a foundation for system simulation.

D. REINFORCEMENT LEARNING ALGORITHM

Reinforcement learning is a branch of machine learning in which an agent learns to interact with its environment in a way that maximizes its reward. This learning process is achieved through a series of decisions, each of which affects not only the immediate reward but also the long-term reward. In reinforcement learning, the agent observes the state (s) of the environment and selects an action (a) based on the state. The selected action (a) is applied to the environment, and the environment returns the reward ($\mathcal{R}$) for the action (a) and the new state (s) to the agent. The agent learns from these experiences, and over time it learns a method (or policy) for choosing an action (a) that maximizes reward ($\mathcal{R}$).

1) Q-LEARNING

Q-learning [6] is one of the classic algorithms in reinforcement learning, where an agent learns an optimal action value function (Q-function) by interacting with its environment. It is a model-free and off-policy learning method, where the agent learns by finding a balance between exploration and exploitation. Q-learning uses a table called a Q-table to store values for each state and action pair. The Q-table is then updated using a learning rate (α) to reflect learning and a discount factor (γ) to determine how much to apply future rewards to current values. The equation (8) below shows the process of updating the Q-table.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[\mathcal{R} + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \tag{8}$$

where s is the current state, a is the selected action, $\mathcal{R}$ is the reward received, s' is the next state caused by the currently selected action a, and a' is all possible actions in the next state s'.

2) DEEP DETERMINISTIC POLICY GRADIENT

Deep deterministic policy gradient (DDPG) [7] is a reinforcement learning algorithm suitable for problems with continuous action spaces. DDPG uses an actor-critic structure and uses deep learning to approximate policies and value functions in complex environments. DDPG combines ideas from Q-learning with gradient descent to learn policies that directly output continuous actions. Actor approximates a policy function that determines an action based on the current state. Critirc approximates the Q-value for a given state and action. Target actor and target critic are networks that are introduced for learning stability, slowly tracking the weights of the original actor and critic. The replay buffer stores and randomly samples experiences to reduce the correlation of the data used for training. Actors and critics are updated in the following steps.

- Critic update:

$$\begin{aligned} y &= \mathcal{R} + \gamma \text{TargetCritic}(s', \text{TargetActor}(s')) \\ Loss_{\text{Critic}} &= \text{MSE}(y, \text{Critic}(s,a)) \end{aligned} \tag{9}$$

Consider y as the target Q-value designated for the update of critic network. This value amalgamates the immediate reward with the anticipated future rewards. Define $\mathcal{R}$ as the reward instantly procured upon executing a chosen action in the prevailing state. The term TargetCritic(s', TargetActor(s')) represents the anticipated Q-value for the subsequent state s', as approximated by the target critic network. Notably, the action in this context is proposed by the target actor network. Furthermore, s' signifies the state transitioned into as a result of the action taken in the current state.

- Actor update:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{\mathcal{s}\sim\mathcal{D}}[\nabla_a Q(\mathcal{s}, a|\theta^Q)|_{a=\mu(\mathcal{s})} \nabla_{\theta^\mu} \mu(\mathcal{s}|Q^\mu)] \tag{10}$$

The symbol ∇ represents the gradient, which is a vector indicating the partial derivative of a function with respect to its input vector, given a scalar output. The notation θ^μ signifies the parameters (or weights) of the actor network. The term J stands for the objective function of actor network, and in the context of DDPG, the algorithm seeks to learn the optimal policy by maximizing this function. The symbol $\mathbb{E}$ denotes expectation, essentially reflecting the average over states $\mathcal{s}$ sampled from the replay buffer $\mathcal{D}$. The expression $\nabla_a Q(\mathcal{s}, a|\theta^Q)$ conveys the gradient of the Q-value (output of the critic network) with respect to the action a, where θ^Q is a parameter set of the critic network. The function $\mu(\mathcal{s})$ designates the action determined by the actor network for the current state $\mathcal{s}$. Lastly, $\nabla_{\theta^\mu} \mu(\mathcal{s}|\theta^\mu)$ represents the gradient of the action (output of the actor network) with respect to the actor network's parameter θ^μ.

- Target network update:

$$\theta' \leftarrow \alpha\theta + (1-\alpha)\theta' \tag{11}$$

where θ is the weight of the original network, θ' is the weight of the target network, and α is the learning rate.

IV. PROPOSED METHOD: DTDO-RA PROCEDURE

A. RANDOM ACCESS PROCEDURE

Before initiating the RA procedure, UEs receive a pilot signal from the BS according to the schedule established by the 3GPP standard. Upon receiving the pilot signal, the UE activates and schedules a RACH (Random Access Channel) procedure to attempt the RA process autonomously. The UE then transmits MSG 1 (RACH preamble) to the BS via PRACH. In this phase, the UE that has transmitted MSG 1 waits for MSG 2 (RAR, Random Access Response) for a preset RAR window time. The BS sends MSG 2 to those UEs that have transmitted MSG 1 without experiencing preamble collisions.

UEs that successfully receive the RAR in MSG 2 proceed to request resources for MSG 3 (RRC Connection Request) after preprocessing the message. Once MSG 3 is transmitted, UEs wait for an RRC (Radio Resource Control) connection response, denoted as MSG 4, and remain in this state for a predefined connection resolution time. Upon receiving MSG 3, the BS performs an internal preprocessing step and responds with MSG 4, allocating resources to the UEs within a predefined range. Once this procedure is complete, the UE is granted the necessary resources and gains access to the channel for data transmission. Fig. 3 illustrates the process of the RA procedure and the associated delay for successful completion.

UEs that fail to receive MSG 2 within the allocated RAR window time select a random backoff (BO) count

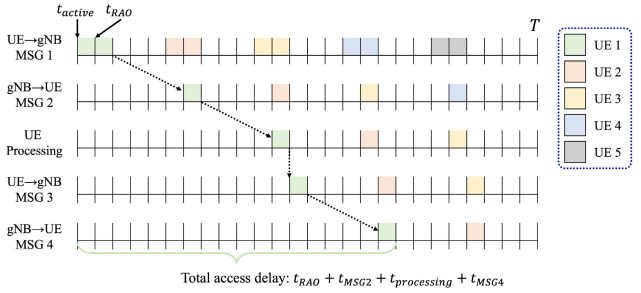


FIGURE 3. The time flow of RA procedure.

from a predefined fixed value range specified by the BI. This selected BO count serves as a crucial factor in the transmission process. Specifically, it significantly impacts the delay that the UE experiences during transmission attempts. The delay associated with preamble retransmission is calculated as:

$$t_{delay} = t_{RAO} + W_{RAR} + BO + \mathcal{L} \tag{12}$$

Here, t_{RAO} represents the time when the PRACH is scheduled, W_{RAR} is the RAR window timer that waits for the RAR to be received, and BO represents the randomly determined BO count. $\mathcal{L}$ denotes the remaining time until the next RAO is scheduled, calculated from the sum of $W_{RAR} + BO$. Fig. 4 illustrates the operational process of the UE that failed to receive MSG 2 due to a preamble collision.

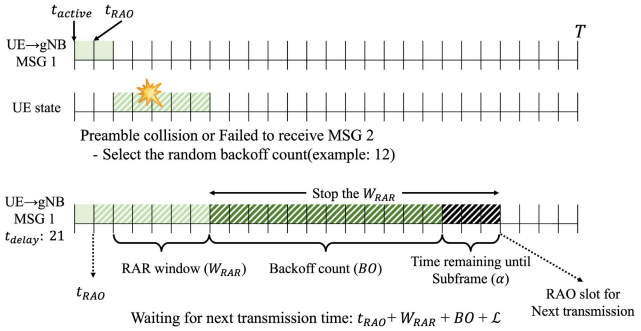


FIGURE 4. Preamble collision and RARs not received.

B. DTDO-RA PROCEDURE

According to the RA procedure, the UE sends MSG 1 (preamble) to the BS. If the BS does not receive MSG 1 or if there is a preamble number collision, the UE must retransmit MSG 1. In this process, the traditional RA procedure mechanism selects a random BO value from a range of fixed BI values. In other words, the BI value is an important value that determines how long the UE should wait before attempting to retransmit after a failed MSG 1 transmission in the RA procedure. This is reducing channel congestion and optimizing power consumption can be effective.

To calculate the power consumption, we consider both standby power and power consumption due to retransmissions. First, during idle periods, the UE operates in a low-power state. Power consumption is calculated as the

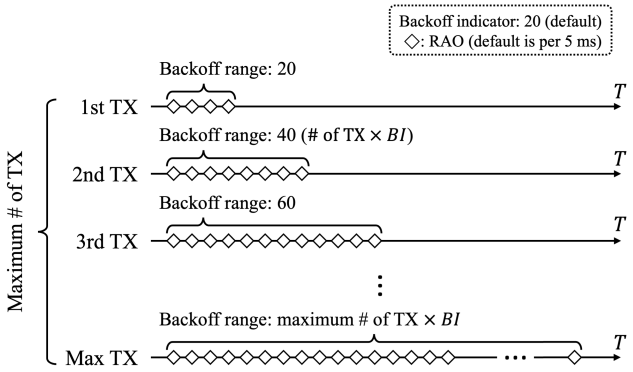


FIGURE 5. Adaptive backoff range with increasing retransmissions.

product of the power consumed per unit time in this low-power state (ω_{Idle}) and the waiting time (T_{Idle}). Therefore, the total power consumption during standby is $\omega_{Idle} \cdot T_{Idle}$. The waiting time T_{Idle} can be calculated as the product of the BI value and the number of transmissions, with a random value selected from within this range. In other words, the maximum power consumption during the idle time can be expressed as indicated in equation (13).

$$\omega_{Idle} \cdot N_{TX} \cdot BI \quad , N_{TX} \in N_{PT_{max}} \tag{13}$$

Subsequently, the power consumption for retransmissions is calculated. Preamble retransmission occurs in UEs that have not received MSG 2 and MSG 4, and for this, the UE must perform additional transmission. Therefore, the total power consumption for preamble transmission is calculated by multiplying the power consumption per unit time (Ω_{TX}), the transmission time (T_{TX}), and the number of retransmissions (N_{TX}). This is equivalent to equation (14).

$$N_{TX} \cdot (\omega_{TX} \cdot T_{TX}) \quad , N_{TX} \in N_{PT_{max}} \tag{14}$$

This paper explores the relationship between power consumption during idle periods and power consumption during preamble retransmissions to optimize overall power usage. To achieve this, we compare equations (13) and (14), which respectively represent power consumption due to latency and power consumption due to preamble retransmission. Equation (15) demonstrates that power consumption due to delay time must be lower than that due to preamble retransmission.

$$\omega_{Idle} N_{TX} BI < \omega_{TX} N_{TX} T_{TX} \quad , N_{TX} \in N_{PT_{max}} \tag{15}$$

Equation (15) can be used to balance the power consumed during transmission with the power consumed during idle time while waiting for the next transmission opportunity. This balance can improve power efficiency and reduce channel congestion. Equation (15) can be reformulated as equation (16) by reducing it to an inequality for BI.

$$BI < \frac{\omega_{TX} T_{TX}}{\omega_{Idle}} \tag{16}$$

This suggests that the delay time should be greater than a specific multiple of the preamble transmission time. This multiple is dictated by the ratio of power consumed per unit time during preamble transmission to that during standby. This enables the UE to minimize power consumption by selecting an appropriate delay time. Building on the above equation, the optimization problem aimed at minimizing power consumption is defined as follows.

$$\begin{aligned} \min_{T_{Idle}, N_{TX}} \quad & \mathcal{P}_{UE} = \frac{1}{N_{UE}} \sum_{i=1}^{N_{UE}} \left\{ N_{i,TX}(\omega_{i,TX} T_{i,TX}) + \omega_{i,Idle} T_{i,Idle} \right\} \\ \text{s.t.} \quad & T_{i,Idle},\ N_{i,TX} \geq 0, \quad N_{i,TX} \in N_{PT_{max}} \\ & BI < \frac{\omega_{TX} T_{TX}}{\omega_{Idle}} \end{aligned} \tag{17}$$

Equation (17) represents the average power consumption due to preamble transmissions and power consumption during the standby state for all UEs in the set $i \in N_{UE}$. The optimization problem aims to identify the most efficient number of preamble transmissions N_{TX} and the BI value to manage the delay time. By doing this, the UE can determine the most efficient way to utilize power while maintaining communication performance. The effect of adjusting the number of preamble transmissions and the delay time is as follows:

1) BI: By adjusting the delay time, the power $\omega_{Idle} N_{TX} BI$ consumed during the waiting period for each UE can be altered. Longer delay times reduce the number of preamble transmissions, but they can also increase the overall system delay; therefore, an optimal balance is needed.
2) N_{TX}: When the number of preamble transmissions is adjusted, the power consumption $N_{TX}(\omega_{TX} T_{TX})$ due to preamble transmissions can be altered for each UE. Reducing the number of preamble transmissions decreases power consumption, but if the network is in poor condition, the reliability of data transmission may be compromised.

Therefore, we aim to minimize the power consumption $\mathcal{P}$ by tuning the two variables. This allows us to find the right balance between low power consumption and network performance. Equation (17) represents the average power consumption of the UE, from which the power consumption per RAO slot can be calculated. Equation (18) represents the average power consumption for all RAO slots. Specifically, for each slot $r \in N_{RAO}$, $\mathcal{P}_{UE,r}$ means the power consumption of the UE activated in the rth RAO slot.

$$\mathcal{P}_{RAO} = \frac{1}{N_{RAO}} \sum_{r=1}^{N_{RAO}} \mathcal{P}_{UE,r} \quad , r \in N_{RAO} \tag{18}$$

Algorithm 1 outlines the pseudocode of the RA process and the DTDO-RA procedure proposed in this paper. The procedure consists of four steps: Send_PREAMBLE, Receive_RAR, Send_MSG_3, Receive_MSG_4.

1) **TRANSMIT_PREAMBLE (lines 1-9):** Here, the UE selects a preamble for random access. If BACKOFF_COUNT is zero, the selected preamble is transmitted to the BS. If BACKOFF_COUNT is not zero,

Algorithm 1 Pseudo Code

```
1: procedure Transmit_Preamble
2:     Select random preamble
3:     if BACKOFF_COUNT == 0 then
4:         Transmit the selected preamble to the BS
5:     else
6:         BACKOFF_COUNT -= 1
7:         Return to TRANSMIT_PREAMBLE procedure
8:     end if
9: end procedure
10: procedure Receive_RAR
11:     if Random Access Response (RAR) is received within the RAR windows then
12:         Uplink resource allocation from the RAR
13:     else
14:         Apply the backoff count indicated by the BI
15:         RETRANSMIT_COUNT += 1
16:         BI = RETRANSMIT_COUNT * BI
17:         Select random BACKOFF_COUNT in BI range
18:         Return to TRANSMIT_PREAMBLE procedure
19:     end if
20: end procedure
21: procedure Transmit_MSG_3
22:     Construct a MSG 3 including the necessary control information
23:     Transmit the MSG 3 to the BS using the uplink resources received in RAR
24: end procedure
25: procedure Receive_MSG_4
26:     if MSG 4 is received within the predefined time (Contention Resolution timer) then
27:         The contention resolution is successful
28:     else
29:         Apply the backoff count indicated by the BI
30:         RETRANSMIT_COUNT += 1
31:         BI = RETRANSMIT_COUNT * BI
32:         Select random BACKOFF_COUNT in BI range
33:         Return to TRANSMIT_PREAMBLE procedure
34:     end if
35: end procedure
```

BACKOFF_COUNT is decremented by one, and the TRANSMIT_PREAMBLE step is repeated.

2) **RECEIVE_RAR (lines 10-20):** The RECEIVE_RAR step checks if the UE has received the RAR within the RAR window. If the RAR is successfully received, the UE is allocated uplink resources in the RAR. There are two cases in which the UE may not receive the RAR: the first case occurs when the selected preamble is also chosen by another UE, leading to a conflict; the second case happens when the BS is unable to respond due to a lack of grant. If the UE does not receive the RAR, it increments the RETRANSMIT_COUNT by 1, and the BI is updated as the product of RETRANSMIT_COUNT and a predefined BI. Then, a random BACKOFF_COUNT within the BI range is selected, and the process returns to the TRANSMIT_PREAMBLE step.
3) **TRANSMIT_MSG_3 (lines 21-24):** The UE constructs an MSG 3 containing the required control information. It transmits the MSG 3 to the BS using the uplink resource received from the RAR.
4) **RECEIVE_MSG_4 (lines 25-35):** The UE checks if it has received MSG 4 within the predefined connection resolution timer. If MSG 4 is received, the RA procedure is successful, and data transmission can proceed. If MSG 4 is not received, RETRANSMIT_COUNT is incremented by 1, and the BI is updated as the product of RETRANSMIT_COUNT and a predefined BI, following the same procedure as in lines 12-16 of the RECEIVE_RAR step. After that, a random BACKOFF_COUNT within the BI range is selected, and the process returns to the TRANSMIT_PREAMBLE step.

C. DTDO-RA WITH REINFORCEMENT LEARNING

Among the elements used in Q-learning and DDPG, the agent is assumed to be the BS. The BS calculates the reward using the success rate ($\mathcal{R}_S$) and average access delay ($\mathcal{R}_D$) for UEs attempting RA procedures in an episode $n(n \in \mathcal{N})$, and the average power consumption $\mathcal{P}_{RAO}$ for all RA slots. Since each factor is composed of different values, a logarithmic scale is applied. The equation (19) below shows how the reward is calculated.

$$\mathcal{R}_n = W_S \log(\mathcal{R}_S) - W_D \log(\mathcal{R}_D) - W_P \log(\mathcal{P}_{RAO}) \quad (19)$$

where W_S, W_D, W_P are the weights of success rate, average access delay, and mean power consumption, respectively. The state $\mathcal{s}_n$ observed in episode $n \in \mathcal{N}$ is the selected BI value, $N_{PT_{max}}$ value, and the observed results $\mathcal{R}_S$, $\mathcal{R}_S$, $\mathcal{R}_D$, $\mathcal{P}_{RAO}$ for $\mathcal{s}_n(BI, N_{PT_{max}}, \mathcal{R}_S, \mathcal{R}_D, \mathcal{P}_{RAO})$. The Fig. 6 shows the learning process for each episodes of a BS that is the subject of learning.

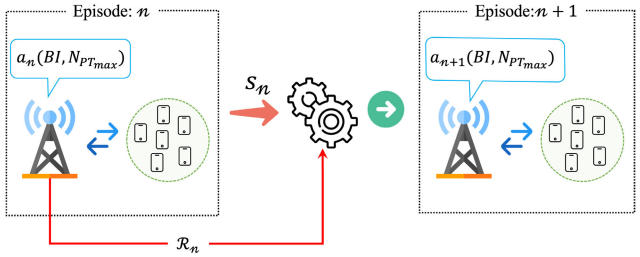


FIGURE 6. Reinforcement learning process of BS.

The actor-critic model of the DDPG algorithm employs a deep neural network (DNN) [33]. Technical terms are explained upon first use. Logical structure and a clear and balanced presentation are emphasized. To reduce the computational complexity of the system, this study adopts a ResNet model [34] with five residual blocks. The input for

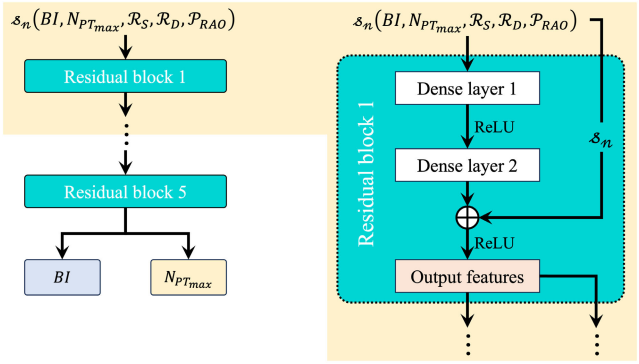


FIGURE 7. Model architecture of actor network.

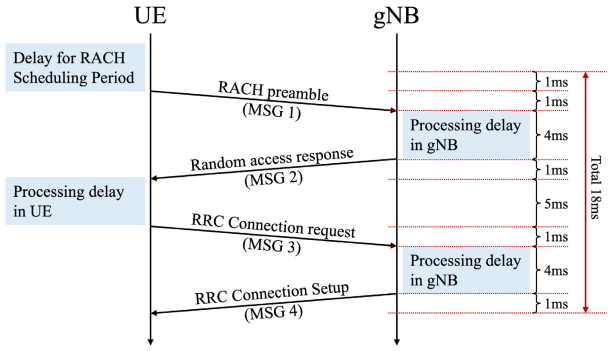


FIGURE 8. RA procedure delay from Talbe B.1.1.1-1 in [32].

the actor network is the state information from the previous episode. The citation style adheres to the chosen style guide. Grammatical correctness is ensured. The language is objective, precise, and free from filler words, biased language, and informal expressions. The model comprises a multi-output layer that yields values for BI and $N_{PT_{max}}$, respectively. The Critic model leverages state information, together with action and reward data randomly selected from the replay buffer, to compute the error. The model architecture of Actor network and residual block configuration are depicted in Fig. 7.

V. SIMULATION RESULTS AND DISCUSSION

A. PARAMETERS AND ENVIRONMENT

As described in Table 1, the system parameters are configured according to the specifications provided in the 3GPP standards document [1]. For the traffic model, a beta distribution is chosen to simulate a scenario with sudden channel access requests. The simulation runs for 10 seconds. RAO slots for RA procedures are provided at 1 ms, 6 ms, and 5 ms intervals for subcarriers, according to the PRACH configuration index of 6. The number of grants available in each RAO slot is limited to 3 uplink grants per RAR and 4 CCEs per PDCCH. This means that the BS can allocate a total of 12 grants per RAO slot for UEs attempting RA procedures, assuming a scenario with limited resource availability.

TABLE 1. Simulation parameters from Table 6.2.2.1.2 in [1].

Parameters	Value
α, β	3, 4
Simulation time	10s * 1000ms
PRACH configure index	$PRACHConfigIndex = 6$
RAO subframe	1, 6
Total number of preambles	54
Number of UL grants per RAR	3
Number of CCEs per PDCCH	4
Total number of UL grants per RAO slots	12
RA-Response window size	5ms
Contention Resolution timer	48ms
MSG 3 Reception Probability	0.9
Maximum number of preamble transmission	10, 8, 6, 4, 2
Backoff indicator	20, 40, 60, 80, 100ms

In the simulation, the propagation delay is configured to comply with the regulations specified in the 3GPP standard document [32]. According to this configuration, the UE sets a scheduling window of 1 ms to initiate the RA procedure. During this procedure, each message transmission incurs a propagation delay of 1 ms. It is worth noting that, according to the guidelines set forth in the aforementioned 3GPP standard document [32], the processing delay times are established at 5 ms for the UE and 4 ms for the BS. Fig. 8 shows an RA procedure that reflects this.

Table 2 shows the power consumption used in the RA procedure steps. The power values are in microjoule (μJ) and refer to [12]. One thing to note from the Table 2 is the high power consumption in the transmission phase. The MSG 3 and Data transmission phases each consume 100 μJ, accounting for the largest portion of the total power used. It can be seen that the power consumed during the transmission of the MSG 1 preamble is 14 μJ, which is significantly lower than the power consumption for MSG 3 and data transmission. Both MSG 2 and MSG 4 received by the UE require the same power consumption of 20 μJ. Subsequently, the power consumption for processing the received MSG 2 and MSG 4 is 12.5 and 30 μJ, respectively. As a result, the minimum power consumption for the RA procedure can be calculated as 296.5 μJ.

TABLE 2. The amount of power used by the RA procedure [12].

Message/data	UE power consumption (μJ)
MSG 1 preamble transmission (Format A1 0.07ms)	14
MSG 2 reception	20
UE processing for MSG 2	12.5
MSG 3 transmission (0.5ms)	100
MSG 4 reception	20
UE processing for MSG 4	30
Data transmission (0.5ms)	100
SUM	296.5

In each simulation, we evaluate multiple performance metrics. These include the success rate of UEs that successfully complete the RA procedure and the average access delay. Both of these metrics are analyzed as functions of the BI value and the Max TX. Additionally, we compare power consumption, varying it according to the number of UEs and the RAO slots, as well as analyze its relationship with channel congestion.

1) **Success Rate:** The success rate is defined as the percentage of UEs that successfully complete the RA procedure out of the total number of UEs. This reflects

the efficiency of the RA procedure and the capacity of the system to handle a large number of simultaneous access attempts.

2) **Average Access Delay:** Average access delay measures the average time it takes for a UE to successfully complete the RA procedure. The lower the delay, the faster the UE can start transmitting data, which reflects the responsiveness of the system and the user experience.
3) **Power Consumption (mJ):** Power consumption is an important aspect of wireless networks, especially when devices often rely on battery power. Power consumption can be affected by a number of factors, including the number of UEs (more devices means more power consumption) and RAO slot usage (inefficient RAO slots use increases power consumption). We also consider the size of the resulting graph, which is expressed in μJ to mJ.
4) **Network Congestion:** Congestion measures the load on the network based on the number of UEs trying to access the system at the same time. High congestion can cause RA procedures to fail due to preamble collisions and increase power consumption due to preamble transmissions.

By comparing these metrics, we can evaluate the effectiveness of the DTDO-RA procedure in terms of minimizing power consumption, managing network congestion, and improving RA performance.

The performance of reinforcement learning is evaluated by the total reward per episode. The reward per episode is the sum of the immediate feedback that the agent receives as it interacts with the environment, and is a direct indication of the performance of the learning process. To evaluate the effectiveness of a reinforcement learning algorithm, we typically use two main metrics

1) **Raw parameter value:** The raw parameter value received by the agent in each episode, represented as time series data, captures the immediate performance of the algorithm. This value can vary significantly from episode to episode and may reflect instability during the initial learning process.
2) **Moving Average:** Calculate a moving average of the raw parameter value to assess the long-term performance trend. Moving averages smooth out short-term volatility and clearly show the stability of the learning process and the overall improvement trend.

As shown in Table 3, the learning parameters for Q-learning. The agent in Q-learning learns the Q-table to select the optimal behavior for a given state. The value of learning rate, which determines how quickly the agent accepts new information when updating the Q-table, is set to 0.001 to enable stable convergence. Discount factor, which determines the value of future rewards, is set to 0.99. The exploration rate is initially set to 1.0 to allow the agent to explore all possible actions, and then decreases to 0.01 over time to allow for optimal behavior as learning progresses. Exploration decay rate is set to 0.001, which acts as a rate to decrease the exploration rate as the agent learns.

TABLE 3. Q-learning parameters.

Parameters	Value
Learning rate	0.001
Discount factor	0.99
Exploration rate	1.0
Max exploration rate	1.0
Min exploration rate	0.01
Exploration decay rate	0.001
W_S	0.3
W_D	0.35
W_P	0.35

TABLE 4. DDPG parameters.

Parameter	Value
Replay buffer size	10,000
Batch size	32
Discount factor	0.99
Learning rate of Actor	0.001
Learning rate of Critic	0.001
Soft update rate	0.001
W_S	0.3
W_D	0.35
W_P	0.35

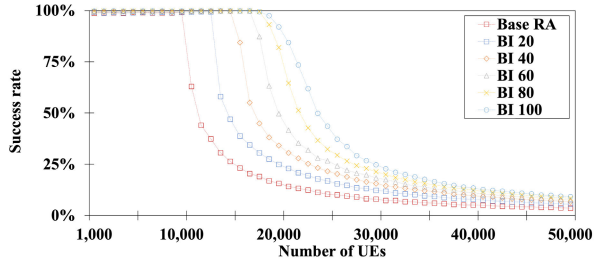
As discussed in Table 4, the parameters used in the training environment of DDPG. The DDPG is a policy-based reinforcement learning algorithm that uses an actor-critic model and is effective in continuous action space. Set the Replay buffer to 10,000, which indicates the size that the agent can store past experiences and reuse them for learning. Set the Batch size to 32 to indicate the size of the batch of experiences used for training. This allows the agent to use mini batch gradient descent to increase the efficiency of learning. Set the discount factor to 0.99, the same as for Q-learning. For the learning rate, we use 0.001 for the actor model and 0.001 for the critic model. Soft update rate is set to 0.001, which acts as a soft update parameter used to update the target network of the actor-critic network.

B. PERFORMANCE EVALUATION

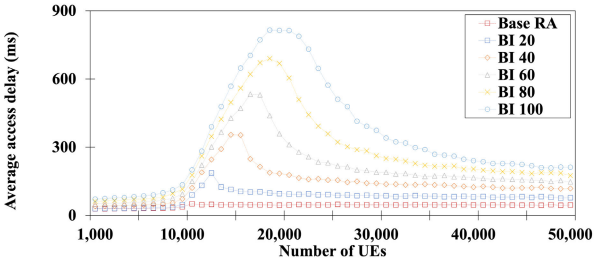
1) SUCCESS RATE AND AVERAGE ACCESS DELAY

Fig. 9 compares the simulation results when following the traditional RA procedure (Base RA) with the simulation results obtained by changing the BI and Max Tx values.

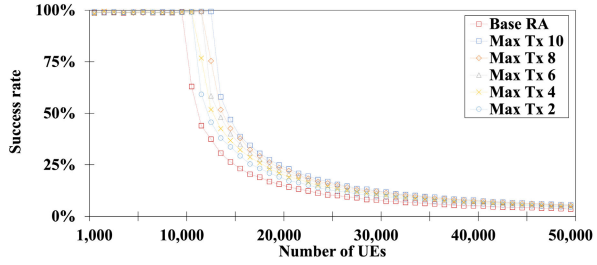
Subfigures (a) and (b) in Fig. 9 present the impact of varying the BI value on the success rate and average access delay. When relying on the Base RA procedure, the success rate sees a significant decline at 10,000 UEs. In contrast, using the DTDO-RA procedure, the success rate starts to decrease when the number of UEs reaches 15,000 with a BI of 20. This results suggests that adjusting the backoff range can effectively reduce congestion. Specifically, setting



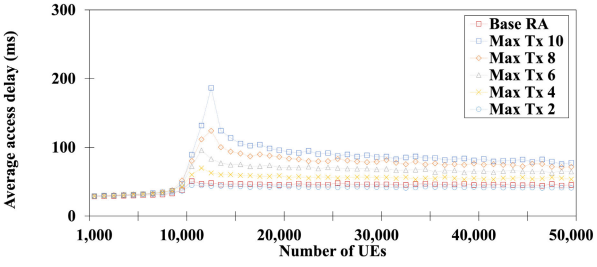
(a) Success rate with varying BI



(b) Average access delay with varying BI

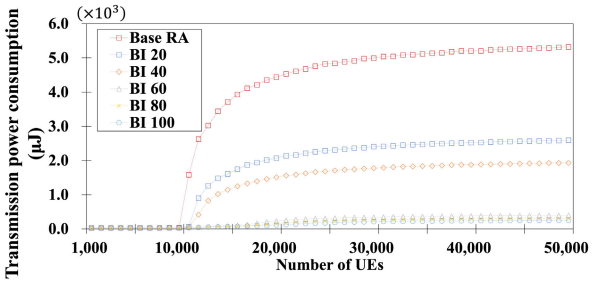


(c) Success rate with varying Max Tx

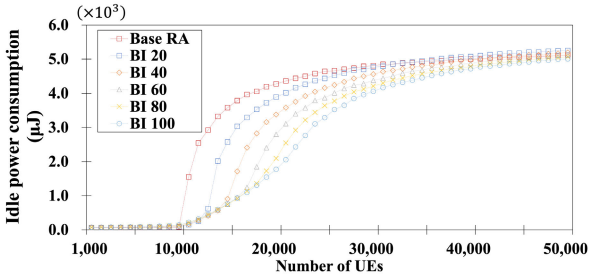


(d) Average access delay with varying Max Tx

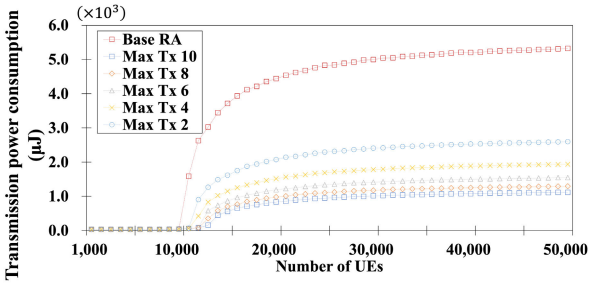
FIGURE 9. Compare success rate and average access delay by BI and Max Tx.



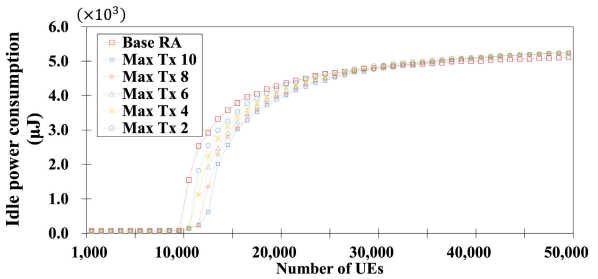
(a) Power consumption in transmission with varying BI



(b) Power consumption in idle state with varying BI



(c) Power consumption in transmission with varying Max Tx



(d) Power consumption in idle state with varying Max Tx

FIGURE 10. Compare power consumption by BI and Max Tx.

the BI value to 100 maintains a high success rate even with 20,000 UEs. However, it should be noted that a higher BI value also leads to an increase in the average access delay, which can escalate to approximately 812.8173 ms when there are 20,000 UEs.

Subfigures (c) and (d) in Fig. 9 illustrate the effects of altering the Max Tx value. Reducing the number of preamble transmissions naturally limits the scalability of the randomized backoff range. In other word, when the Max Tx is set to 2, the maximum backoff range is 40, which similar with the specifications of the Base RA procedure. However, as in the Max Tx 10 graphs in subfigures (c) and (d), allowing various backoff ranges helps UE better explore highly congested areas, an advantage similar to that observed when adjusting BI. Moreover, the average access delay is reduced by up to 4.3 times compared to adjustments in the BI, given that the backoff range does not substantially increase.

2) POWER CONSUMPTION PER UES

Fig. 10 (a), (b) shows the power consumption as the BI value changes. At low numbers of UEs (especially below 10,000), the transmit power consumption for different BI values shows a similar trend. For instance, Fig. 10-(a) reveals that at 10,000 UEs, the power consumption of Base RA is 0.041 mJ, while it is 0.038 mJ for BI 100—a negligible difference of 0.003 mJ. This means that the power consumption is

constant at low numbers of UEs. However, as the number of UEs increases to 15,000, there is a sharp difference in transmit power consumption. If you compare the transmit power consumption of Base RA and BI 100 with increased BI value at 15,000 UEs, you can see a reduction of about 98.06%, which is 3.71 *mJ* and 0.072 *mJ*, respectively. This shows that adjusting the BI value can reduce the amount of preamble transmissions in congestion channels, thereby reducing collisions and saving transmission power. However, in Fig. 10-(b), the power consumption in the idle state continues to increase as the BI value increases. In particular, for the more long time simulations, the DTDO-RA procedure consumes more standby power in the idle state than Base RA. Based on this result, it can be seen that a proper adjustment of the BI value is required rather than unconditionally increasing the BI value to achieve a high success rate and reduce channel congestion.

Fig. 10 (c), (d) shows the variation of power consumption with Max Tx value. In an environment with less than 10,000 UEs, the difference in transmit power consumption between Base RA and Max Tx is negligible, similar to (a) in Fig. 10. Starting from the results for more than 15,000 UEs in Fig. 10-(c), the difference in transmission power consumption becomes significant. It can be seen that when the value of Max Tx is reduced using the DTDO-RA procedure, the results of Max Tx 10 and Max Tx 2 increase the transmission power consumption by about 192.73%, from 0.55 *mJ* to 1.61 *mJ*. In other words, this result shows that the smaller the value of Max Tx, the higher the transmission power consumption can be. The power consumption at idle state in Fig. 10-(d) shows the lowest power consumption of 2.56 *mJ* when applying the technique proposed in this paper at a number of 15,000 UEs with Max Tx of 10. However, as the number of UEs increases, the power consumption in the idle state continues to increase. For example, when the number of UEs is 50,000, the power consumption is 5.11 *mJ* for the Base RA, but it is higher at 5.24 *mJ* with the technique proposed in this paper and Max Tx set to 10. These results show that reducing the maximum number of transmissions to reduce the transmission power leads to an increase in power consumption in the idle state. Therefore, it is necessary to adjust the number of Max Tx appropriately, and if it is set to too small a number, it is difficult to obtain a gain in the success rate as shown in Fig. 9-(c).

3) COMPARISON OF POWER CONSUMPTION PER RAO SLOTS

Fig. 11 shows the average power consumption per RAO slot during the RA procedure. Fig. 11 (a) and (b) reflect the variation in power consumption per RAO slot with BI value, while (c) and (d) reflect the variation associated with different Max Tx counts. A clear observation from Fig. 11-(a) is that for RAO slots fixed at 2,000, there is a power consumption difference of about 3.64706 *mJ*. Referring to the data described in Table 5 for Fig. 11 (a) and (b), we can see that as the BI value increases, the power consumption

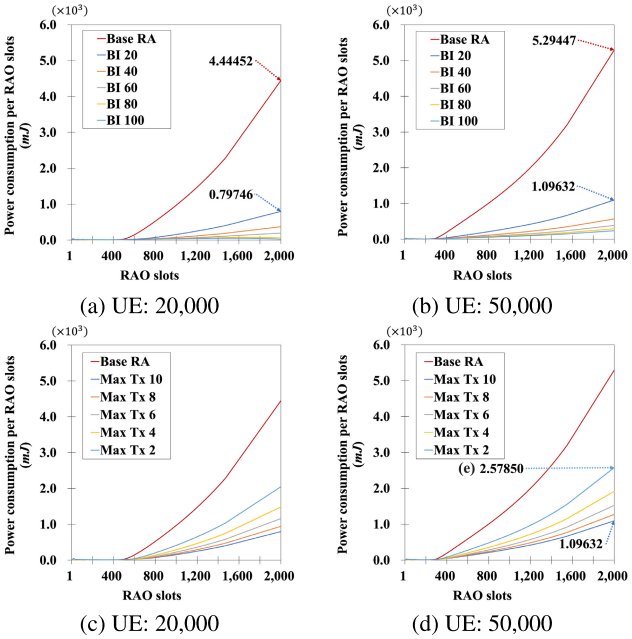


FIGURE 11. Power consumption per RAO slot with varying BI and Max Tx for UE 20,000 and 50,000.

TABLE 5. Average power consumption per RAO slot according to BI.

UEs	RAO slots	Base RA	BI 20	BI 40	BI 60	BI 80	BI 100
20,000	400	0.00718	0.00591	0.00717	0.00550	0.00644	0.00836
	800	0.52249	0.07218	0.03893	0.02888	0.02670	0.02379
	1,200	1.47184	0.24889	0.11571	0.06787	0.04978	0.04120
	1,600	2.81139	0.49840	0.23175	0.12900	0.07443	0.04942
	2,000	4.44452	0.79746	0.36948	0.19105	0.05880	0.02315
50,000	400	0.19345	0.04332	0.02915	0.02396	0.02151	0.02010
	800	1.00539	0.21290	0.11671	0.08394	0.06872	0.05943
	1,200	2.00261	0.42137	0.22381	0.15629	0.12233	0.10180
	1,600	3.43398	0.71576	0.37429	0.25714	0.19766	0.16208
	2,000	5.29447	1.09632	0.56831	0.38659	0.29443	0.23886

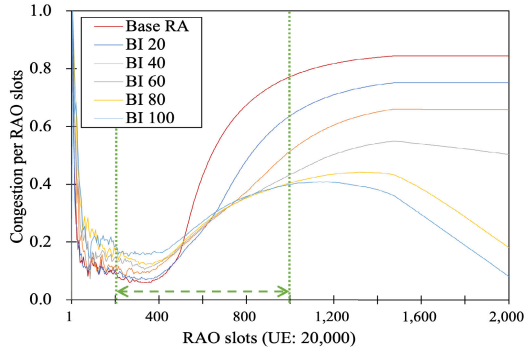
decreases in parallel. This confirms that improving BI tasks an important role in optimizing power efficiency.

In Fig. 11 (c) and (d), the results show the relationship between the decrease in the Max Tx count and the power consumption per RAO slot. In particular, the power consumption increases as the Max Tx count decreases. This trend can be attributed to the limited ability to adjust the backoff range as the Max Tx count decreases. In general, it is expected that the power consumption will decrease as the number of transmissions decreases, but it can be seen that this is due to the continuous transmission of the preamble as the success rate of the RA procedure decreases. Fig. 11-(e) shows the power consumption in an environment with a Max Tx count of 2, where the backoff range is limited to BI 20 * 2, requiring a select of ranges up to 40 ms. This causes the UE to initiate approximately 1.4822 *mJ* more preamble transmissions compared to when Max Tx is 10, which increases power consumption. With this reduced range, the UE will struggle to operate efficiently in congested channel conditions. This can be correlated to the lower success rate of the RA procedure, as shown by the results in Fig. 9-(c).

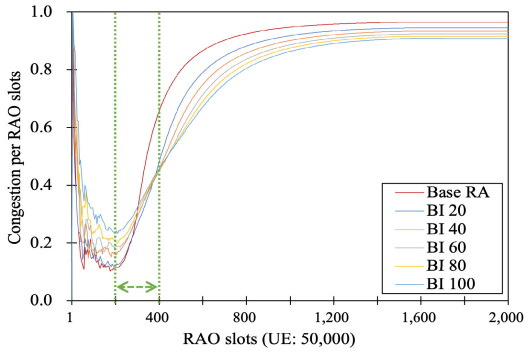
TABLE 6. Average power consumption per RAO slot according to Max Tx.

UEs	RAO slots	Base RA	Max Tx 10	Max Tx 8	Max Tx 6	Max Tx 4	Max Tx 2
20,000	400	0.00718	0.00591	0.00683	0.00865	0.00566	0.00665
	800	0.52249	0.07218	0.08335	0.10503	0.13816	0.20392
	1,200	1.47184	0.24889	0.29250	0.36189	0.46547	0.65152
	1,600	2.81139	0.49840	0.58840	0.72415	0.92759	1.28345
	2,000	4.44452	0.79746	0.94346	1.15997	1.48451	2.04803
50,000	400	0.19345	0.04332	0.04343	0.04686	0.05924	0.08331
	800	1.00539	0.21290	0.24148	0.28412	0.35628	0.48175
	1,200	2.00261	0.42137	0.48519	0.57660	0.72248	0.97159
	1,600	3.43398	0.71576	0.82891	0.99043	1.24301	1.67095
	2,000	5.29447	1.09632	1.27418	1.52586	1.91727	2.57850

The data presented in Table 6 offer more details about power consumption per RAO slot for configurations with 20,000 and 50,000 UEs. A consistent trend emerges: power consumption increases as the Max Tx count decreases. For instance, when we examine the results for 20,000 UEs, the power consumption in RAO slot number 2,000 is 0.7975 *mJ* with a Max Tx count of 10. This figure stands in contrast to the 2.0480 *mJ* observed when the Max Tx count is set to 2. Similarly, for 50,000 UEs, power consumption in the same RAO slot rises from 1.0963 *mJ* with a Max Tx count of 10 to 2.5785 *mJ* when the Max Tx count is set to 2. These observations indicate that the Max Tx count plays a significant role in power efficiency, emphasizing the need for careful parameter selection in the RA procedure. Thus, it becomes evident that excessively reducing the Max Tx count in an attempt to decrease transmission power consumption could have adverse effects on the goal.



(a) UE: 20,000



(b) UE: 50,000

FIGURE 12. Compare congestion per RAO slot by BI for UE 20,000 and 50,000.

4) COMPARISON OF CONGESTION PER RAO SLOTS

Fig. 12 shows the congestion level of each RAO slot for an environment of two UEs, 20,000 and 50,000. The results compare between the conventional RA procedure and the DTDO-RA procedure under various BI settings. Congestion is defined by the ratio of UEs attempting the RA procedure to the total number activated in a specific RAO slot. The initial RAO slots show high congestion values due to the few UEs that complete the RA procedure. As an example, if 20 UEs are activated in one RAO slot and only 2 succeed, the congestion ratio becomes 18/20.

In Fig. 12-(a) for 20,000 UEs, the Base RA experiences a sudden increase in channel congestion level after the 500th RAO slot. This rise in congestion is attributed to the limited resources that prevent UEs from completing the RA procedure. Furthermore, UEs that are unsuccessful in preamble transmission attempt preamble transmission in subsequent RAO slots, which further affects UEs trying the RA procedure in later slots. However, with the DTDO-RA procedure, UEs can select from a wider range of RAO slots for preamble transmission compared to the Base RA procedure. Thus, one can avoid congested channel environments made by limited resources. The findings from Fig. 12-(a) indicate that adjusting wait times is an effective way to address the congestion problem. While using a higher BI value might lead to increased access delay, it is advantageous when a significant number of UEs initiate the RA procedure simultaneously. But, in Fig. 12-(b) for 50,000 UEs, a mere adjustment of the BI is not enough to alleviate congestion. Nevertheless, when applying a BI value of 20 using the DTDO-RA procedure, the increase in congestion is tempered than Base RA. These results indicate that the adjustment of BI values has a positive effect in a scenario where the UE is massive. The efficacy of the DTDO-RA procedure, particularly with a BI of 100, stands out in handling congestion during peak traffic times.

In traffic model 2 based on section II, the simulation environment is characterized by a sudden increase in UEs attempting to access at a certain point in time. This sudden increase in the number of UEs attempting access can be seen when the RAO slot of the simulation approachs 200. In Fig. 12, the change in congestion can be seen from the point where the RAO slot is 200, as shown by the green arrow (<—>) of each graph. In this area, it can be seen that the congestion increases rapidly and its performance is reversed. When the BI is set to 100, the congestion level during the initial RAO slot (before 200th RAO slot) is measured higher than when the Base RA is used. This increases the backoff range and can inevitably result in a longer delay during preamble transmission. As a result, fewer UEs successfully complete the initial RA procedure in an environment with a BI value of 100. However, as the density of UEs increases, different results are shown. Looking at the previous simulation result Fig. 9, a larger BI value significantly increases the success rate in the RA procedure.

Therefore, congestion tends to gradually decrease because the success rate in the RA procedure increases as the BI value increases. These observations show that in a scenario where UEs suddenly increase, optimizing BI can be important to improve both congestion and success rate of RA procedures, and can alleviate congestion in other RAO slots.

C. COMPARATIVE ANALYSIS OF Q-LEARNING AND DDPG RESULTS IN DTDO-RA

Both Q-learning and DDPG algorithms perform learning up to 4,000 episodes in a network environment with 50,000 UEs. The learning agent dynamically configures the environment of the RA procedure according to the network state by selecting the BI value and Max Tx value respectively. The BI value is selected between 10 and 400 ms, and the Max Tx value is selected between 2 and 40. The results of the learning are summarized in the form of reward accumulation, success rate, and power consumption as shown in Fig. 13, 14, 15, 16 and Tables 7, 8, 9, 10.

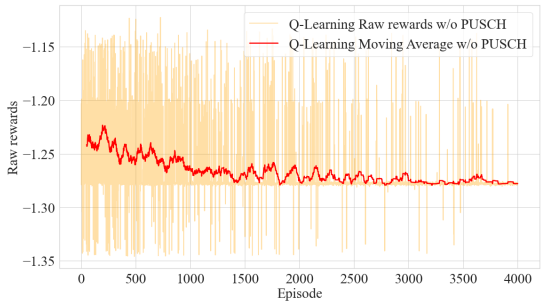
TABLE 7. Summary statistics: rewards.

	Q-learning w/o PUSCH	DDPG w/o PUSCH	Q-learning w/ PUSCH	DDPG w/ PUSCH
Mean	-1.268302	-0.587082	-1.444927	-1.012743
Std	0.032066	0.125521	0.155341	0.150928
Min	-1.345927	-0.779995	-2.010621	-1.139072
25%	-1.278392	-0.735048	-1.378065	-1.100824
50%	-1.277695	-0.548587	-1.377613	-1.098995
75%	-1.276899	-0.464281	-1.377293	-0.947662
Max	-1.122634	-0.447376	-1.376057	-0.510639

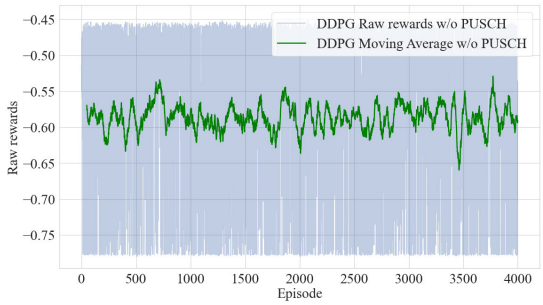
Fig. 13 (a), (b) and Table 7 'w/o PUSCH' show the evolution of the rewards obtained by the agents of each learning algorithm in without PUSCH. We can see that the average reward of Q-learning is lower than the average reward of DDPG. This indicates that DDPG obtained a higher reward for the given problem. As the standard deviation of the reward of DDPG is higher than the standard deviation of Q-learning, the DDPG algorithm shows a higher variability in its reward. This may indicate that DDPG is better suited to more complex environments with greater policy diversity.

Fig. 13 (c) and (d), as well as Table 7 'w/ PUSCH' show the rewards obtained by the agents using each learning algorithm when considering PUSCH. The results show that DDPG consistently achieves higher rewards than Q-learning, indicating a more efficient policy in the given context. However, the inclusion of PUSCH resources adds an additional layer of complexity. The simulations involving PUSCH indicate that the Q-learning algorithm encounters difficulties in adjusting to the expanded action space, resulting in a decrease in average rewards. In contrast, DDPG demonstrates robust performance even with PUSCH, although the variability in its rewards suggests susceptibility to the dynamic access conditions introduced by PUSCH.

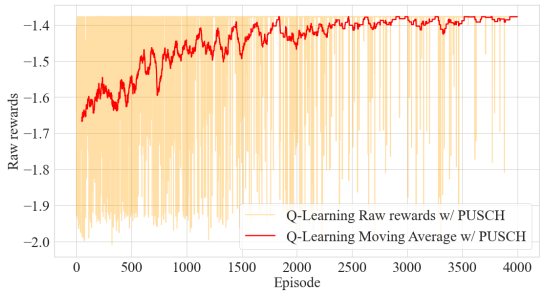
Fig. 14 (a), (b) and Table 8 'w/o PUSCH' show the trend of the success rate for two learning algorithms as the network environment changes when not considering PUSCH. The findings simulation that the DDPG algorithm has a



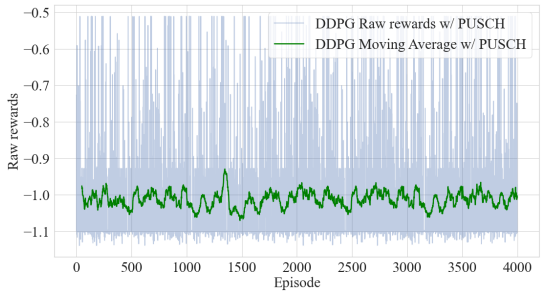
(a) Moving average of reward in Q-learning



(b) Moving average of reward in DDPG



(c) Moving average of reward in Q-learning with PUSCH

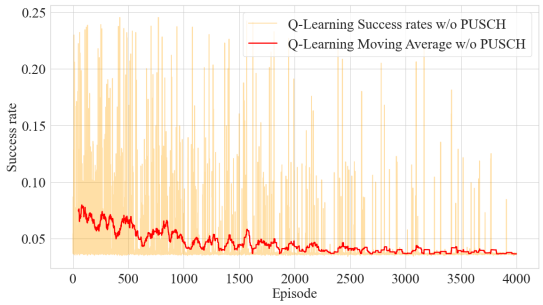


(d) Moving average of reward in DDPG with PUSCH

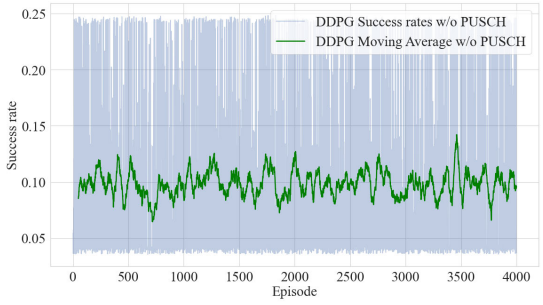
FIGURE 13. Comparison of Q-learning and DDPG performance metrics: rewards.

TABLE 8. Summary statistics: success rate.

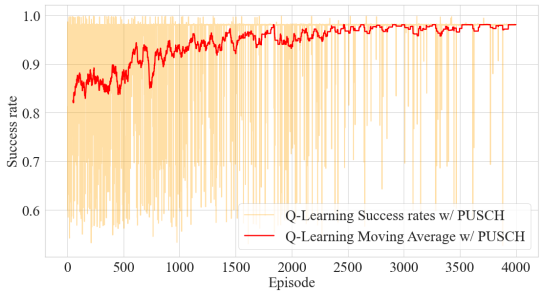
	Q-learning w/o PUSCH	DDPG w/o PUSCH	Q-learning w/ PUSCH	DDPG w/ PUSCH
Mean	0.0448	0.0984	0.944633	0.657204
Std	0.0283	0.0753	0.099261	0.121745
Min	0.0343	0.0353	0.527240	0.526280
25%	0.0364	0.0402	0.980120	0.567560
50%	0.0368	0.0434	0.980720	0.595880
75%	0.0373	0.1433	0.981200	0.702150
Max	0.2454	0.2491	0.998360	0.998200



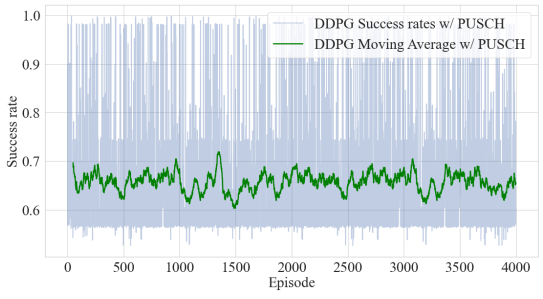
(a) Moving average of success rate in Q-learning



(b) Moving average of success rate in DDPG



(c) Moving average of success rate in Q-learning with PUSCH



(d) Moving average of success rate in DDPG with PUSCH

FIGURE 14. **Comparison of Q-learning and DDPG performance metrics: success rate.**

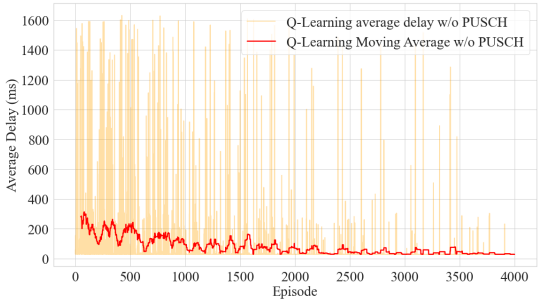
mean success rate that is over twice as high as Q-learning. Specifically, Success rate of DDPG distribution appears more widely spread, indicating a higher likelihood of achieving greater success rates. Q-learning has a comparatively low success rate; however, the standard deviation remains small, and the median, 25th, and 75th percentiles have similar values, demonstrating that the success rate is reasonably consistent. The consider of PUSCH in the simulation has a significant impact on the performance metrics.

As demonstrated in Fig. 14 (c), (d) and Table 8 'w/ PUSCH', the success rate is significantly increased, particularly for the Q-learning algorithm. This improvement can be explained by the improved data transmission capability and more efficient channel utilization with PUSCH. The standard deviation of the success rate with PUSCH still indicates variability in performance, but the range has narrowed, suggesting a more stable and reliable communication process. The increase in the average success rate for Q-learning with PUSCH implies that the algorithm adapts well to the added complexity and benefits from the additional transmission opportunities provided by PUSCH. The near-optimal maximum success rate highlights the effectiveness of PUSCH in supporting the RA procedure. However, for DDPG, the success rate performance is lower than that of Q-learning, despite the high reward performance. This is attributed to the weight of the parameters that determine the reward of the learning algorithm. It is suggested that each weight used to calculate the reward should be adjusted appropriately according to the network environment.

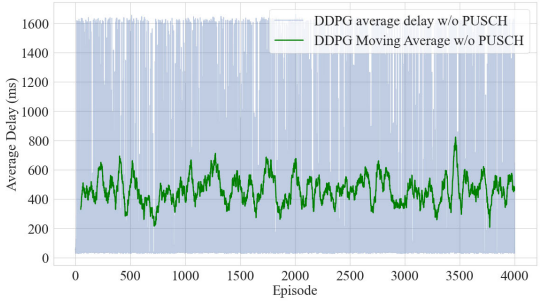
TABLE 9. **Summary statistics: average access delay.**

	Q-learning w/o PUSCH	DDPG w/o PUSCH	Q-learning w/ PUSCH	DDPG w/ PUSCH
Mean	74.97ms	461.84ms	140.19ms	1190.09ms
Std	206.12ms	607.12ms	344.37ms	608.96ms
Min	27.99ms	28.19ms	35.28ms	35.34ms
25%	28.80ms	35.15ms	35.43ms	539.12ms
50%	28.98ms	39.46ms	35.48ms	1589.64ms
75%	29.23ms	897.29ms	35.54ms	1609.38ms
Max	1631.55ms	1660.16ms	1989.25ms	2047.18ms

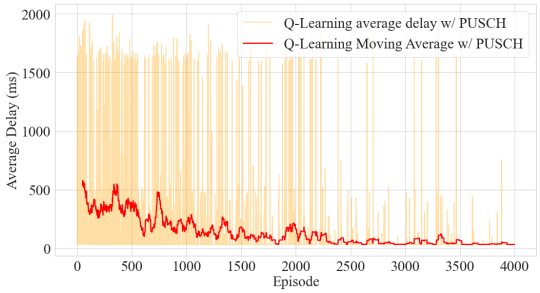
Fig. 15 (a), (b) and Table 9 'w/o PUSCH' show the trend of the average access delay for both algorithms as they learning when PUSCH is not considered. The significant finding from this result is that Q-learning has a notably lower average access delay than DDPG. This demonstrates that Q-learning has superior efficiency in decision-making for certain environments. On the other hand, DDPG exhibits a high standard deviation in access delays, which implies that certain episodes could experience excessively high access delays. Consequently, DDPG may have prolonged access delays in more intricate environments, but it could also display brief access delays in some cases. By comparing the median and interquartile range, it becomes apparent that Q-learning exhibits consistently low access delays, whereas DDPG demonstrates a higher median, but very high access delays in the upper quartile. This implies that DDPG may have unstable performance in certain situations and therefore, Q-learning may be a more reliable choice for network environments where access delay must be consistently low. Fig. 15 and Table 9 demonstrate the progression of average access delay for Q-learning and DDPG algorithms, considering PUSCH. Q-learning maintains its superior efficiency with PUSCH, resulting in a lower average access delay compared to DDPG. This consistency in Q-learning highlights its robustness in environments where PUSCH is a significant factor. The use of PUSCH reveals the variability in performance



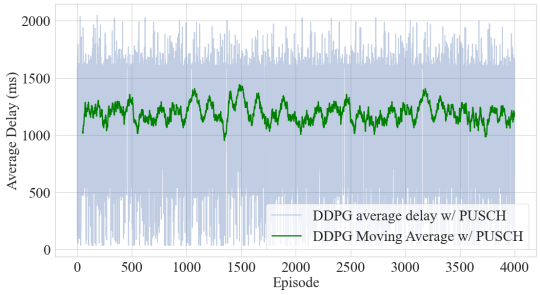
(a) Moving average of average access delay in Q-learning



(b) Moving average of average access delay in DDPG

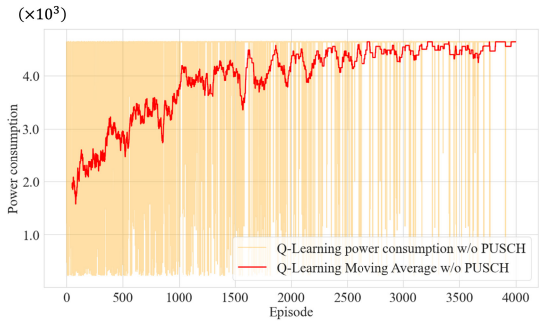


(c) Moving average of average access delay in Q-learning with PUSCH

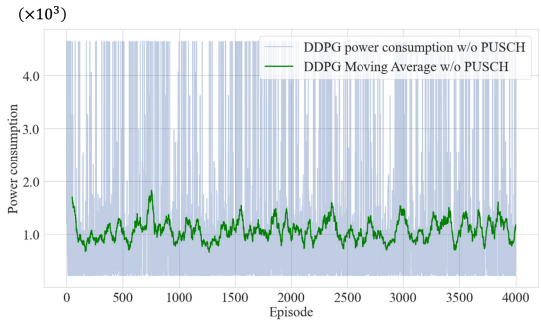


(d) Moving average of average access delay in DDPG with PUSCH

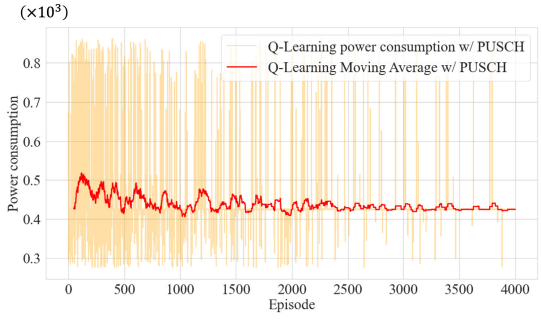
FIGURE 15. Comparison of Q-learning and DDPG performance metrics: average access delay.



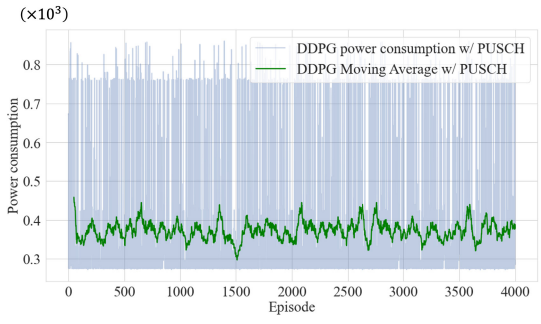
(a) Moving average of power consumption in Q-learning



(b) Moving average of power consumption in DDPG



(c) Moving average of power consumption in Q-learning with PUSCH



(d) Moving average of power consumption in DDPG with PUSCH

FIGURE 16. Comparison of Q-learning and DDPG performance metrics: power consumption.

of DDPG, as its higher standard deviation suggests the potential for both significantly high and reasonably low access delays depending on the network scenario. The median and interquartile ranges indicate that Q-learning remains consistently reliable with PUSCH, while DDPG performance is more variable, with a higher median delay and peak delays, particularly in the upper quartile. When deploying DDPG in network environments where PUSCH plays a crucial role, careful consideration is necessary to avoid unpredictability in access delays.

The power consumption results are displayed in Fig. 16 and Table 10. The comparison indicates in Fig. 16 (a) and (b) that Q-learning needs more average power per episode

TABLE 10. Summary statistics: power consumption (*mJ*).

	Q-learning w/o PUSCH	DDPG w/o PUSCH	Q-learning w/ PUSCH	DDPG w/ PUSCH
Mean	4.0398	1.0835	0.4367	0.3728
Std	1.4022	1.2146	0.0899	0.1643
Min	0.2278	0.2275	0.2748	0.2714
25%	4.6354	0.2496	0.4232	0.2751
50%	4.6394	1.0675	0.4239	0.2947
75%	4.6423	1.0849	0.4246	0.3698
Max	4.6568	4.6498	0.8607	0.8607

compared to DDPG with the average power consumption being almost four times higher when without PUSCH. Nevertheless, power consumption of Q-learning converges as the training progresses. Conversely, DDPG exhibits a low mean power consumption but a high standard deviation, suggesting that its power consumption fluctuates under varying network conditions. DDPG may be more suitable for power-critical scenarios, whereas Q-learning would be more apt for environments with fewer power constraints.

Fig. 16 (c) and (d) show the results considering PUSCH. The consider of PUSCH results in high power consumption during the initial stages of learning when employing Q-learning. However, the use of PUSCH leads to a significant reduction in power requirements, suggesting that Q-learning benefits from channel optimization. This indicates an adaptive capability within Q-learning that, when leveraged with PUSCH, can lead to substantial energy savings. The convergence of power consumption in Q-learning suggests a stabilization in its decision-making process, optimized for energy efficiency when PUSCH is considered. On the other hand, DDPG consistently exhibits lower mean power consumption, with and without PUSCH, indicating a natural efficiency within the DDPG algorithm for power management. However, the higher standard deviation suggests that the performance of DDPG is sensitive to different network conditions. The addition of PUSCH appears to improve the performance of DDPG by reducing the variability and further reducing the power consumption, making it more suitable for power critical scenarios.

D. DISCUSSION

1) EFFECT OF BACKOFF INDICATOR

The simulation results presented in Fig. 9, as well as in Fig. 10 and 11, demonstrate that adjusting the BI value plays a pivotal role in managing both the success rate and the power consumption during the RA procedure. Furthermore, the impact of the BI value grows increasingly significant as the number of UEs escalates, when compared to the Base RA method. Specifically, the data in Fig. 12 reveal that in congested channel conditions, elevating the BI value can serve as an effective strategy to alleviate channel congestion. This is attributed to the fact that a higher BI value expands the range of RAO slots that a UE can opt for, thereby effectively reducing the likelihood of preamble collisions. Consequently, the UE gains a higher probability of steering clear of congested channels. Viewed from a theoretical perspective, this suggests that the BI value can be harnessed not merely as a parameter for backoff time management, but also as a crucial tool for optimizing power usage and averting channel congestion.

2) EFFECT OF MAXIMUM PREAMBLE TRANSMISSIONS

The results of controlling the Max Tx value provide insight into power consumption. Comparing the success rate and average access delay in Fig. 9, we can see that there is not much difference from the basic RA. However, looking at the power consumption in Fig. 10 and Fig. 11, we can see that reducing the Max Tx value to an extreme extent does not significantly reduce the transmission power consumption. In other words, the DTDO-RA procedure requires more transmissions to successfully complete the RA procedure when the number of preamble transmissions is reduced, which increases the power consumption.

3) MINIMIZATION OF POWER CONSUMPTION

From the simulation results, we can see in ig. 10 and 11 that the power consumption of the mMTC network can be reduced by adjusting the BI value and Max Tx. As the mMTC environment evolves, especially due to the proliferation of devices and the resulting resource constraints, the need for energy savings becomes more and more important. From the results in Fig. 10, we can compare the power consumption in transmission and idle for each parameter, and we can see that the power consumption obviously decreases with the increase of the BI value. When adjusting the Max Tx value, setting it too low will result in increased transmit power, but the increase in idle power is not significant. Changes in BI and Max Tx play an important role in reducing power consumption during network operation, as shown in Fig. 11. An important implication is that BI, which has only been recognised as a parameter for managing backoff time, can be an important factor in determining power consumption, i.e. under congested channel conditions, careful selection of BI values can significantly reduce power consumption. Similarly, setting the Max Tx value correctly can balance the number of preamble transmissions with power consumption.

4) CHANNEL CONGESTION IN MMTC SCENARIOS

As the number of UEs increases, the challenge of effectively managing channel congestion becomes increasingly important, especially in mMTC environments where many devices must compete for limited channel resources. Efficient channel access mechanisms are essential to minimize delay time and reduce power consumption while maintaining reliable communications. One of the main factors that contribute to channel congestion in mMTC environments is the increasing frequency of preamble collisions. As shown in Fig. 2, as the number of UEs attempting RA procedures simultaneously increases, the preamble collision rate increases accordingly. This can also be seen in Fig. 12, where we see a sharp increase

in channel congestion at a certain point. If the RA procedure fails due to preamble collisions, the UE will retry in the next RAO slot, which is likely to further exacerbate the congestion. To address this issue, in this paper we propose that channel congestion can be effectively mitigated by expanding the range of selectable RAO slots by adjusting the BI value.

5) INSIGHTS INTO APPLYING REINFORCEMENT LEARNING

Reinforcement learning applied to the RA procedure shows that DDPG generally outperforms Q-learning in achieving higher rewards and success rates, but with greater variability in success rates, indicating that it is effective in a variety of environments. The ability of DDPG to discover the optimal action policy, along with its efficiency in complex scenarios, helps in the selection of learning algorithms where power is critical due to the low power consumption required to maintain performance. With its stability, Q-learning is well suited for consistent environments, while DDPG is more adaptive to complex and changing conditions. However, the integration of PUSCH significantly improves the success rate of Q-learning, allowing it to better respond to the dynamic network environment despite its initially high power consumption, and reduces power consumption by optimizing channel usage, increasing the adaptability of Q-learning for energy efficiency.

As a result, the apply of reinforcement learning has a positive impact on RA procedures in bursty traffic environments. For example, in an environment where power is sufficient but the success rate needs to be increased, Q-learning can work effectively to increase the number of retransmissions by choosing a low BI value to increase the success rate by increasing the number of retransmissions. On the other hand, when there is not enough power and the network environment is more complex and constrained, DDPG is an effective learning method and can adjust to increase the BI value to increase access delay and avoid congested channel environments. In this way, the application of reinforcement learning plays an important role in optimizing RA procedures to balance power consumption and success rate.

VI. CONCLUSION

The DTDO-RA method was presented in this study as a means of minimizing power consumption and channel congestion in mMTC networks. The approach focuses on optimizing the BI and Max Tx values in the RA procedure. By dynamically adjusting these parameters, the method effectively mitigates interference and channel congestion, enhancing the efficiency of RA procedures. And also, the use of Q-learning and DDPG algorithms allows for adaptive optimization of BI and Max Tx values, taking into account the dynamic features of the network environment.

The efficiency of the DTDO-RA scheme has been validated through extensive simulations, showing significant improvements in power efficiency and RA success rates, especially in congested network conditions. Optimizing Max Tx helps to balance network reliability and efficiency. However, it is important to calibrate it carefully to prevent increased power consumption due to excessive transmissions. These findings provide insights for future improvements of the DTDO-RA procedure, with the aim of customizing it to various network scenarios and requirements.

REFERENCES

[1] *Service Requirements for Machine-Type Communications (MTC) Stage 1 (Rel. 16)*, 3GPP, document (TS) 22.368, v16.0.0, Jul. 2020.

[2] L. Miuccio, D. Panno, and S. Riolo, "A new contention-based PUSCH resource allocation in 5G NR for mMTC scenarios," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 802–806, Mar. 2021.

[3] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled Internet of Things: Network architecture and spectrum access," *IEEE Comput. Intell. Mag.*, vol. 15, no. 1, pp. 44–51, Feb. 2020.

[4] M. V. da Silva, R. D. Souza, H. Alves, and T. Abrão, "A NOMA-based Q-learning random access method for machine type communications," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1720–1724, Oct. 2020.

[5] N. Jiang, Y. Deng, and A. Nallanathan, "Traffic prediction and random access control optimization: Learning and non-learning-based approaches," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 16–22, Mar. 2021.

[6] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992.

[7] 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," 2015, *arXiv:1509.02971*.

[8] *Study on RAN Improvements for Machine-Type Communications*, 3GPP, document TS 37.868, v11.0.0, Sep. 2011.

[9] I. Leyva-Mayorga, C. Stefanovic, P. Popovski, V. Pla, and J. Martinez-Bauset, "Random access for machine-type communications," in *Wiley 5G Ref: The Essential 5G Reference Online*. Hoboken, NJ, USA: Wiley, 2019, pp. 1–21.

[10] O. Arouk and A. Ksentini, "General model for RACH procedure performance analysis," *IEEE Commun. Lett.*, vol. 20, no. 2, pp. 372–375, Feb. 2016.

[11] S.-S. Yoo, S.-H. Lee, J. Shin, S.-M. Oh, and J.-H. Kim, "Performance evaluation of LTE—A random access procedure for IoT service," *J. Korean Inst. Commun. Inf. Sci.*, vol. 41, no. 8, pp. 965–973, Aug. 2016.

[12] Y. Piao, Y. Kim, and T.-J. Lee, "Random power back-off for random access in 5G networks," *IEEE Access*, vol. 9, pp. 121561–121569, 2021.

[13] Y. R. Li, M. Chen, J. Xu, L. Tian, and K. Huang, "Power saving techniques for 5G and beyond," *IEEE Access*, vol. 8, pp. 108675–108690, 2020.

[14] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.

[15] *Quantitative Analysis on UL Data Transmission in Inactive State*, ZTE Microelectronics, 3GPP Technical Document, document R2-1701932, Feb. 2017.

[16] A. Azari, C. Stefanovic, P. Popovski, and C. Cavdar, "On the latency-energy performance of NB-IoT systems in providing wide-area IoT connectivity," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 1, pp. 57–68, Mar. 2020.

[17] A. Azari, G. Miao, C. Stefanovic, and P. Popovski, "Latency-energy tradeoff based on channel scheduling and repetitions in NB-IoT systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[18] H. Zhou, Y. Deng, L. Feltrin, and A. Höglund, "Analyzing novel grant-based and grant-free access schemes for small data transmission," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2805–2819, Apr. 2022.

[19] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[20] S. Han, X. Xu, Z. Liu, P. Xiao, K. Moessner, X. Tao, and P. Zhang, "Energy-efficient short packet communications for uplink NOMA-based massive MTC networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12066–12078, Dec. 2019.

[21] T. Islam, D. Lee, and S. S. Lim, "Enabling network power savings in 5G-advanced and beyond," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1888–1899, Jun. 2023.

[22] *Study on Network Energy Savings for NR*, 3GPP, Standard (TR) 38.864, v18.0.0, Dec. 2022.

[23] 3GPP, "Medium access control (MAC) protocol specification," Sophia Antipolis, France, Tech. Rep. TS 36.321, V9.3.0, Jun. 2010.
[24] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "On the accurate performance evaluation of the LTE—A random access procedure and the access class barring scheme," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 7785–7799, Dec. 2017.
[25] A. H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "A comprehensive distributed queue-based random access framework for mMTC in LTE/LTE—A networks with mixed-type traffic," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12107–12120, Dec. 2019.
[26] Q. Xiong, B. Yu, C. Qian, X. Li, and C. Sun, "Random access preamble generation and procedure design for 5G-NR system," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–5.
[27] G. Schreiber and M. Tavares, "5G new radio physical random access preamble design," in *Proc. IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 215–220.
[28] J. Youn, J. Park, J. Oh, S. Kim, S. Ahn, S. Cho, S. Park, and C. You, "CeRA-eSP: Code-expanded random access to enhance success probability of massive MTC," *Sensors*, vol. 22, no. 20, p. 7959, Oct. 2022.
[29] T. A. Pham and B. T. Le, "A proposed preamble detection algorithm for 5G-PRACH," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2019, pp. 210–214.
[30] S. S. Zehra, M. Magarini, R. Qureshi, S. M. N. Mustafa, and F. Farooq, "Proactive approach for preamble detection in 5G-NR PRACH using supervised machine learning and ensemble model," *Sci. Rep.*, vol. 12, no. 1, May 2022, Art. no. 8378.
[31] A. Rodriguez Medel and J. M. C. Brito, "Random-access accelerator (RAA): A framework to speed up the random-access procedure in 5G new radio for IoT mMTC by enabling device-to-device communications," *Sensors*, vol. 20, no. 19, p. 5485, Sep. 2020.
[32] *Feasibility Study for Further Advancements for E-UTRA (LTE-Advanced)*, 3GPP, document TR 36.912, V17.0.0, Mar. 2022.
[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

JIHA KIM received the B.S. degree in computer software engineering from Korea Bible University, Seoul, South Korea, in 2020. He is currently pursuing the master's and Ph.D. combined course with the Department of Information and Communication Engineering, Myongji University. He served as a soldier in the Republic of Korea Marine Corps from 2014 to 2016. He holds a patent in the field of deep learning. His research interests include mobile communication, network analysis, deep learning, and reinforcement learning to improve the efficient communication environment.

YONGHO KIM was born in Seoul, South Korea. He received the B.S. degree in computer software engineering from Korea Bible University, Seoul, in 2021. He is currently pursuing the master's and Ph.D. combined course with the Department of Information and Communication Engineering, Myongji University. His research interests include AI/ML modeling and prediction, intelligent system development, and computer vision.

SEUNGHYUN PARK received the Ph.D. degree in cybersecurity from Korea University, in 2021. He joined as a Faculty Member of Hansung University, in September 2021, where he is currently an Assistant Professor with the Division of Computer Engineering. From 2014 to 2021, he was with Hyundai Motor Company and Kia in connected vehicles and enterprise cybersecurity for security risk assessment, remediation, and incident response. Before that, he was a Senior Researcher with the Advanced Institute of Technology, Korea Telecom Corporation, where he researched media technology and data analytical methods, from 2011 to 2014. He also developed mobile software in LG Electronics Inc., from 2008 to 2011. His current research interests include automotive cybersecurity and risk assessment, web framework, and data analytics models.

JAYH (HYUNHEE) PARK (Member, IEEE) received the Ph.D. degree in electronics and computer engineering from Korea University, Seoul, South Korea, in 2011. She joined as a Faculty Member of Myongji University, in 2020, where she is currently an Associate Professor with the Department of Information and Communication Engineering. She is also a Supervisor of the Data Analysis and Networking (DAN) Laboratory. From March 2017 to February 2020, she was with the Department of Computer Software, Korean Bible University, Seoul, as an Assistant Professor. Since November 2014, she has been with LG Electronics as a Senior Researcher in Wi-Fi standardization (IEEE 802.11ax, Wake Up Radio, and Wi-Fi Alliance). From January 2013 to October 2014, she joined the INRIA Research Center as a Postdoctoral Researcher, where she works with the DIONYSOS Research Group and Telecom Bretagne as a Postdoctoral Researcher. She undertakes the system implementation for QoE on wireless networks with DIONYSOS Research Group and Telecom Bretagne. From September 2011 to February 2013, she joined Korea University as a Research Professor. Her research interests include wireless networks, mobile edge/cloud computing, and big data analysis. She served as the Program Co-Chair for the IMIS 2020; the Organizing Committee for the ICTC2020, ICTC2021, and BWCCA 2021; and the Workshop Organizer for FINGNet 2019 and 2020. She is a Guest Editor of the *Electronics* and *Journal of Advanced Transportation*.