IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 July 2023, accepted 24 July 2023, date of publication 27 July 2023, date of current version 8 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299340

RESEARCH ARTICLE

AoI-Aware Resource Management for Smart Health via Deep Reinforcement Learning

BEINING WU[1], (Student Member, IEEE), ZHENGKUN CAI[2], WEI WU[3], AND XIAOBIN YIN[1]
[1]School of Mathematics and Statistics, Anhui Normal University, Wuhu 241002, China
[2]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[3]College of Mechanical and Aerospace Engineering, Jilin University, Changchun 130025, China

Corresponding author: Xiaobin Yin (xbyinzh@ahnu.edu.cn)

This work was supported in part by the Natural Science Foundation of Anhui Province under Grant 2008085MA06.

ABSTRACT The freshness of information is critical for patient vital signs and physiological parameters in the healthcare system because changes in these parameters can indicate a patient's overall health status and guide treatment decisions. In this paper, we consider an edge device-aided smart healthcare system that relies on a resource management scheme. The medical center requires patient information, and edge nodes process the latest measurements received by each wearable device. Our goal is to find the optimal strategy to minimize the worst case of information freshness, i.e., the peak AoI age of information (PAoI). Firstly, we model the problem as a Markov Decision Process (MDP). Then, we design two separate Reinforcement Learning (RL)-based algorithms to find the optimal strategy that minimizes energy consumption and the average PAoI. To minimize energy consumption, we propose a pair of sleep mechanisms, including the N policy and p wake-up policy, to improve the energy efficiency of each wearable device. Simulation results show that the proposed wake-up strategy and the proposed RL algorithm make a better trade-off between the average PAoI and power dissipation compared to the baseline schemes.

INDEX TERMS Smart health, age of information (AoI), sleep-scheduling, deep reinforcement learning (DRL), deep deterministic policy gradient (DDPG).

I. INTRODUCTION

Smart health empowers sophisticated diagnostic tools to deliver advanced treatment for patients and smart health-based equipment to improve the quality of care by providing real-time vital indicators [1], [2]. Specifically, smart health is capable of providing an efficient and fast flow of information to patients and caregivers, thus enhancing the efficiency of the healthcare sector [3]. In smart health system, there is a large amount of data to be transmitted, processed and stored, but the traditional cloud computing architectures cannot cope with the needs of running a smart health system [4]. Therefore, mobile edge computing (MEC) has been proposed as a new approach for smart health [5], [6]. which can be utilized to boost the effectiveness of information transmission and reduce power dissipation. Reference [5] uses MEC nodes to meet the requirements of deep neural network algorithm training, thus providing low latency and high performance information transmission services for intelligent medical systems. Reference [6] investigates the multi-edge server MEC system and uses deep reinforcement learning (DRL) algorithms to obtain the optimal policy, thus optimising the duration of information transmission between servers based on non-orthogonal multiple access (NOMA) and providing a better solution to the information transmission problem of smart health systems.

MEC-based smart health architecture is composed of wearable devices, a patient data aggregator (PDA), mobile/infrastructure edge nodes (MEN), an edge cloud, and a medical center, respectively. Specifically, wearable devices are responsible for sensing the patient's status through the body's local sensor network, while the PDA acts as a communication hub that transmits the information to the infrastructure. The MEN processes and stores the data and

The associate editor coordinating the review of this manuscript and approving it for publication was Paulo Mendes.

forwards it to the cloud, and the edge cloud stores and analyzes the patient's data to enable the medical center to take timely medical care [7]. Due to the intricate nature of the information, in order to ensure precise evaluation of various aspects of a patient's physical condition, each update is assumed to be independent and identically distributed (i.i.d.) within the healthcare center.

During the transmission of patient information, the freshness of the information has a significant influence on the timely implementation of medical treatments, especially for surgical reliability in every hospital. For example, a highly contagious strain that attacks the upper respiratory tract of the human body and puts the respiratory system at risk or even paralyzes it, makes real-time feedback on the patient's physical condition essential, as poor timeliness of information on the patient's condition can lead to complete deterioration. Similarly, chronic diseases (such as heart and lung-related diseases) require emergency measures within 12 hours of the onset, and the timeliness of the information can determine whether the procedure can be performed in time, thus affecting the patient's vital signs.

In order to effectively quantify the concept of information freshness [8], the age of information (AoI) is proposed as a metric that indicates the time elapsed since the moment of generation when the information was updated [9]. References [10] and [11] have both derived expressions for the average Age of Information (AoI) and Peak Age of Information (PAoI) based on an M/G/1 queueing system. On the other hand, research [12], in comparison to [10] and [11], evaluates the freshness of information in the system by employing three different scheduling policies. Within the realm of intelligent healthcare systems, edge nodes are employed for the reception, storage, and subsequent transmission of patient data conveyed by PDA, forwarding it to the cloud. Throughout this process, the "freshness" of the data directly influences the real-world performance of edge-assisted smart healthcare systems [13].

Furthermore, wearable and implantable medical devices (IMDs) are extensively utilized in smart healthcare scenarios to monitor chronic diseases. However, the non-rechargeable batteries used in most of these devices have a limited lifespan, which fails to meet the demand for continuous monitoring. Therefore, besides optimizing the battery material and structure, the working mechanism of the device is also a crucial factor in reducing battery energy consumption and extending its working life. Thus, the optimization of the device's working mechanism to minimize energy consumption and prolong its working life is a key issue addressed in this study.

II. RELATED WORKS

The issue of minimizing the AoI is particularly important in the healthcare architecture of smart health, as prompt and accurate assessment of the user's vital signs is crucial. In previous research on AoI, the main consideration has been to investigate the optimization of the average AoI metric. However, average AoI does not accurately represent the dynamics of AoI over continuous time [14]. Therefore, to accurately represent the long-term behavior of AoI, we introduce the PAoI metric, which represents the worst-case delay in the freshness of the information being used [15].

Several approaches have been adopted to optimize the AoI metric. Lv et al. [16] proposed an online auction mechanism called PreDisc to optimize the metric of AoI. PreDisc leverages dynamic programming to greedily allocate resources in each time slot while considering a preemptive factor to balance the newly arrived tasks and the ongoing tasks. Sharan et al. [17] considered th e energy-saving scheduling problem of AoI minimization, which was solved using a segmented linear approximation method. While these methods mainly aim to minimize AoI at the level of the optimization algorithm, the dynamics of AoI as a continuous time process has not been fully considered.

To fully characterize the dynamic process of AoI over time, Wu et al. [18] formulated the AoI minimization problem as a Markov decision process (MDP). Moreover, taking into account the intricate nature of the MDP, it has been further subdivided into an alternative near-optimal strategy based on the Lyapunov drift function. Additionally, the aspect of user fairness has been considered, and a greedy policy has been proposed to minimize the maximum expected AoI for users.

In addition to the heuristic algorithms employed above, there has been some research into the use of reinforcement learning (RL) algorithms to optimally solve the problem of minimising AoI [29], [30], [31]. Deep RL (DRL) is a combination of deep learning and RL, which solves the problem of large state-action space or continuous state-action space by fitting Q-tables or direct fitting strategies with the powerful representational power of deep neural networks, and the convergence speed is faster [32]. Deep Q learning is a typical example in DRL, in which deep Q networks are widely used, such as resource allocation in the NOMA system in [33], and drone path planning in [34]. However, the output decisions of DQN can only be discrete, a drawback that leads to quantization errors for continuous action tasks. However, deep deterministic policy gradient (DDPG) methods are proposed to better solve these problems, and DDPG is based on the actor-critic structure, which is an enhanced version of the deterministic policy gradient (DPG) algorithm.

In the transmission of information, in addition to information timeliness, energy consumption is also an important issue that cannot be ignored and is a challenge to extend the working life of edge nodes [35]. In order to reduce the number of redundant nodes, ensure the efficiency of the nodes and thus reduce the overall energy consumption of the work, the sleep/wake mechanism is therefore widely used. Reference [35] proposes an $M/M/1/C$ queuing model based on the N policy, where nodes in the dormant state are switched to the wake-up state for data transmission work when the packet volume reaches a threshold N, effectively extending

TABLE 1. Contrasting our contribution to the literature.

Feature \ Ref	[19]	[15]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	Our work
Learning based algorithm	×	×	✓	✓	×	×	✓	×	×	✓	✓	✓
Multiple sensors	×	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓
Multiple uesrs	×	✓	×	×	×	×	×	×	×	×	×	✓
AoI optimization	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
MDP Modeling	×	×	✓	✓	×	✓	×	✓	×	×	✓	✓
Sleep-scheduling	×	✓	×	×	✓	×	×	×	×	×	×	✓

the working life of the nodes. Reference [36] then models a dynamic N policy based on the different arrival rates of the packets. In [37], the authors modelled the working state of the nodes according to the sleep/wake mechanism, i.e. the probability of a node being woken up is p and the probability of being in a dormant state is $1-p$. The energy consumption of the nodes was modelled according to a MDP and the optimal wake-up probability p^* was derived. Reference [38] then proposes an effective energy management mechanism based on the MDP model based on p wake-up probability, the parameters in the study are determined by simulation verification.

In summary, our research stands out from other works by considering resource management solutions in edge-assisted intelligent healthcare systems and utilizing RL-based algorithms to attain optimal solutions. Our study not only builds upon the dormant mechanism of edge devices but also takes into account the simultaneous optimization of average PAoI and energy consumption, which are two distinct performance metrics. Additionally, we have devised two separate DRL algorithms tailored to different action spaces, effectively tackling the resource management problem.

A. CONTRIBUTIONS

This study considers MEC-based smart health architecture that consists of wearable device, PDA, MEN, edge cloud and medical center, as shown in Figure 1.

Our objective is to discover the optimal information acceptance strategy for MEN that minimizes the average PAoI, which more effectively reflects the long-term characteristics of AoI compared to traditional AoI. To address the challenge of minimising average PAoI, we propose using RL algorithms to learn the optimal policy for different environments. Furthermore, due to the limitations of MEN's battery size, we also propose a sleep scheduling algorithm to extend the lifetime of the system and ensure its robustness. In summary, our contributions are summarized as follows:

- Firstly, we formulate the information status update as a Markov decision problem, since MDP is a powerful tool that fully characterizes the dynamic process of average PAoI over time, and allows for the design of an optimal resource management scheme.
- Secondly, we design a DRL-based algorithms to find the optimal strategy that minimizes energy consumption and the average PAoI, and introduces a pair of sleep mechanisms to improve the energy efficiency of each wearable device. The N policy combined with the p wake-up policy.
- Thirdly, to optimize the N policy and p wake-up policy, we propose a Deep Reinforcement Learning (DRL) approach based on a joint DQN-DDPG network to effectively optimize the sleep scheduling strategy, because the state and action spaces are large and involve both discrete and continuous actions. The proposed DRL method has been validated by simulation to be more effective in optimising the objective function than other algorithms.

The contributions made in this paper are clearly contrasted with the literature in Table 1.

B. ORGANIZATION

The remainder of this article is structured as follows: In Section III, we introduce the system model and problem formulation. In Section IV, we describe two sleep scheduling frameworks based on DRL. Section V presents the simulation results, and in Section VI, we draw our conclusions.

III. SYSTEM MODEL

A. QUEUE DESCRIPTION BASED ON N-POLICY AND p WAKE-UP POLICY

We consider a smart healthcare network consisting of a patient, wearable device, PDA, edge node, edge cloud and medical centre, as shown in Figure 1. The markings of the arrows in Figure 1 indicate the sequential process of transmission of the raw data. The wearable device continuously monitors the patient's status and transmits it to the PDA, the PDA aggregates the collected data and transmits it to the MEN, which performs the intermediate processing and storage functions of the data and transmits it to the edge cloud. The edge cloud analyses the patient data and transmits it to the medical centre to provide further medical services. In the realm of authentic systems, prior knowledge can be acquired through predetermined assumptions or prolonged surveillance and statistical analysis.

The concepts of "queue awakening" highlight the threshold N and wake-up probability p, which can be utilized to regulate the operational frequency of edge nodes and the latency of buffered data packets. The N strategy entails the server entering an awakened state to receive and process data packets once the queue reaches the threshold value N. On the other hand, the p wake-up policy involves the server switching between sleep and awakenen policy states based

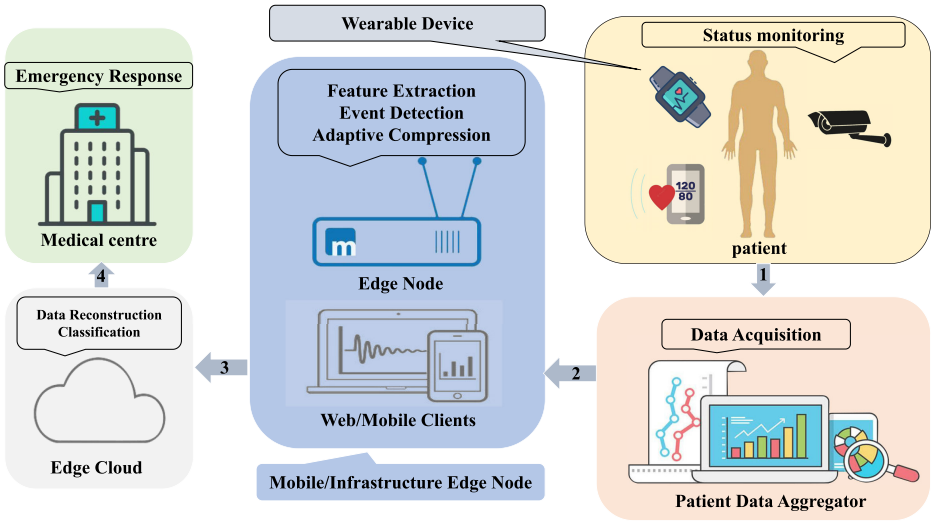


FIGURE 1. Smart healthcare network.

TABLE 2. Key notations.

Notation	Definition
N, p	Packet Optimal Threshold, optimal wake-up probability
λ_0, λ_1	Dormant packet transfer rate, packet transmission rate in wake-up state
$\mu, E[I_N]$	Average packet processing rate, idle period
$E[B_N], E[T_N]$	Busy period, busy cycle
P_I, P_B	Probability that an edge node is idle, probability that an edge node is in a wake-up state
L_0, L_1	Number of packets expected in the dormant state, number of packets expected in the wake-up state
$L, H_{ub,n}$	Expected total number of packets, the upper limit of channel capacity
$D, \psi_{t,n}$	Available bandwidth, the transmit power of the edge nodes
$G_0, \lvert r_{t,n}(t)\rvert$	Noise spectral density, the channel coefficient of the wireless link between the medical centre and the nth edge node at time t
θ_0, d_n	Small-scale fading parameter, link length
a_0, A_{peak}	The path loss of the link, average PAoI
E_{SR}, E_{RS}	The average energy consumption of an edge node switching from a dormant to the wake-up state per unit time, the average energy consumption of switching from an excited to a dormant state
e_{sr}, e_{rs}	The power consumption of the edge node each time it switches from the dormant to the wake-up state, the power consumption of the edge node each time it switches from the excited to the dormant state

on a probability of p, with a probability of $1-p$ for awakening [37], [39]. According to [37] and [39], both policies are effective in reducing energy consumption. Drawing inspiration from their findings, we have integrated these two awakening policies in our queuing model to minimize energy consumption and enhance information freshness.

In this paper, we investigate the problem of minimising PAoI while minimising energy consumption in an $M/M/1$ queue under an N policy with a p wake-up policy. We assume that the edge node is not in a completely dormant state but can provide self-state information to the device. This enables the device to determine when the node is in a dormant or awakened state and adjust its packet transmission rate accordingly. Due to the varying workload between the dormant and awakened states, the edge node has different sampling rates in these two states. In the dormant state, the arrival rate of data packets is denoted as λ_0. When the number of packets reaches a threshold N, the node starts to enter the wake-up state with probability p. When the nodes enter the wake-up state, the packet arrival rate is λ_1 while the packet processing rate satisfies the negative exponential distribution of μ. The individual edge node operating states are shown in Figure 3.

B. M/M/1 QUEUING MODEL WITH N-POLICY AND p WAKE-UP POLICY

In this subsection, we present the Markov queuing model based on the N policy with the p wake-up policy and the results of its steady-state analysis. The state of the system is represented by the pair $\langle \delta, i \rangle$, $\delta = 0$ and 1, $i = 1, 2, \ldots$, where $\delta = 0$ and $\delta = 1$ indicates that the node is in a sleep and wake-up state. i is the number of packets queued at the edge node. When analysing the state of a system, we use the following notation:

- $P_{\langle 0,0 \rangle}$ = the probability of no packets within the edge node. (dormant state)
- $P_{\langle 0,i \rangle}$ = the probability of having n packets inside the edge node. (dormant state), where $i = 1, 2, \ldots$
- $P_{\langle 1,i \rangle}$ = the probability of having n packets inside the edge node. (wake-up state), where $i = 1, 2, \ldots$

The $M/M/1$ queuing model based on the N policy and the p wake-up policy is shown in Figure 4, where the circular

chain at the top and the circular chain at the bottom represent the dormant and wake-up states of the edge nodes respectively. λ_0 and λ_1 denote the packet arrival rates of nodes in the dormant and wake-up states respectively, while μ denotes the average service rate of the node. We may wish to remember: $\rho_0 = \lambda_0/\mu$, $\rho_1 = \lambda_1/\mu$. The steady state equations for $P_{\langle 0,i\rangle}$ and $P_{\langle 1,i\rangle}$ are as follows:

$$\begin{cases} P_{\langle 0,0\rangle}\lambda_0 = P_{\langle 1,1\rangle}\mu, \\ P_{\langle 0,i\rangle} = P_{\langle 0,0\rangle}, & (i \leqslant N), \\ P_{\langle 0,i\rangle}\lambda_0 = P_{\langle 0,i-1\rangle}\lambda_0(1-p), & (i > N), \\ P_{\langle 1,1\rangle}\,(\lambda_1+\mu) = P_{\langle 1,2\rangle}\mu, \\ P_{\langle 1,i\rangle}\,(\lambda_1+\mu) = P_{\langle 1,i-1\rangle}\lambda_1 + P_{\langle 1,i+1\rangle}\mu & (i \leqslant N+1), \\ P_{\langle 1,i\rangle}\,(\lambda_1+\mu) = P_{\langle 1,i-1\rangle}\lambda_1 + P_{\langle 1,i+1\rangle}\mu \\ +P_{\langle 0,i-1\rangle}\lambda_0 p & (i > N+1). \end{cases} \tag{1}$$

Since the model considered in this study is difficult to solve by probability generating function (PGF), we consider a recursive approach. It can be concluded that in the dormant state:

$$\begin{aligned} P_{\langle 0,i\rangle} &= P_{\langle 0,0\rangle} \quad (i \leqslant N), \\ P_{\langle 0,i\rangle} &= (1-p)^{i-N} P_{\langle 0,0\rangle} \quad (i > N). \end{aligned} \tag{2}$$

The transition probabilities for the wake-up state are shown below for $i \leqslant N+1$ and $i > N$:

$$\begin{aligned} P_{\langle 1,i\rangle} &= \frac{1-\rho_1^i}{1-\rho_1}\rho_0 P_{\langle 0,0\rangle} \quad (i \leqslant N+1), \\ P_{\langle 1,N+k\rangle} &= \left[\frac{(1-p)-\rho_1^{k-1}\left(\rho_1^{N+1}-p\right)}{1-\rho_1}\right. \\ &\quad \left.-(1-p)+(1-p)^{k-1}\right]\rho_0 P_{\langle 0,0\rangle} \quad (k \geqslant 2). \end{aligned} \tag{3}$$

Meanwhile, based on the property that the sum of transition probabilities is 1, i.e., $\sum_{n=1}^{i}(P_{\langle 0,i\rangle} + P_{\langle 1,i\rangle}) = 1 (N \ll i \ll \infty)$, we can obtain the probability of no packets within the edge node, which is $P_{\langle 0,0\rangle}$:

$$P_{\langle 0,0\rangle} = \frac{1}{\left[N - \frac{(1-p)^{i-N+1}+(1-p)^i-2(1-p)}{p} + \frac{N+i+1-\rho_1^{N+1}-i\rho_1^i}{1-\rho_1}\rho_0\right]}. \tag{4}$$

We define idle period $E[I_N]$, busy period $E[B_N]$, and busy cycle $E[T_N] = E[I_N] + E[B_N]$, respectively, with P_I denoting the probability of a node being in an idle state and P_B denoting the probability of a node being in a wake-up state. The dormant state probability P_I can be found:

$$\begin{aligned} P_I &= \sum_{n=0}^{i} P_{\langle 0,i\rangle} \\ &= \left[N + \frac{(1-p)-(1-p)^{i+1}}{p}\right] p\langle 0.0\rangle. \end{aligned} \tag{5}$$

According to the concept of total probability, the probability of an edge node being in an awakened state P_B can be expressed as follows:

$$\begin{aligned} P_B &= 1 - P_I \\ &= 1 - \left[N + \frac{(1-p)-(1-p)^{i+1}}{p}\right] P_{\langle 0,0\rangle}. \end{aligned} \tag{6}$$

By leveraging the memorylessness property of the exponential distribution, the duration of dormant time $E[I_N]$ can be represented as the sum of N random variables with a mean of $\frac{1}{\lambda_0}$ and i random variables with a mean of $\frac{1}{\lambda_0(1-p)}$. Due to the percentages of running time during idle and busy periods being given respectively by $P_I = \frac{E[I_N]}{E[T_N]}$ and $P_B = \frac{E[B_N]}{E[T_N]}$ for the edge nodes, we can use (5) and (6) to calculate the expected lengths of idle and busy periods for the nodes:

$$\begin{aligned} E[I_N] &= \frac{N}{\lambda_0} + \frac{i-N}{\lambda_0(1-p)} \\ &= \frac{i-Np}{\lambda_0(1-p)}, \\ E[T_N] &= \frac{E[I_N]}{P_I} \\ &= \frac{i-Np}{\lambda_0(1-p)\left[N + \frac{(1-p)-(1-p)^{i+1}}{p}\right] P_{\langle 0,0\rangle}}, \\ E[B_N] &= P_B E[T_N] \\ &= \frac{(i-Np)\left\{1 - \left[N + \frac{(1-p)-(1-p)^{i-N+1}}{p}\right] P_{\langle 0,0\rangle}\right\}}{\lambda_0(1-p)\left[N + \frac{(1-p)-(1-p)^{i-N+1}}{p}\right] P_{\langle 0,0\rangle}}. \end{aligned} \tag{7}$$

The expected number of packets for a node when the edge node is in the dormant and wake-up states is represented by L_0 and L_1 respectively:

$$\begin{aligned} L_0 &= \frac{\frac{(N+1)^2}{2} + N(1-p) + \frac{(1-p)-(1-p)^{i+1}}{p} - (N+1)(1-p)^{i+1}}{N - \frac{(1-p)^{i-N+1}+(1-p)^i-2(1-p)}{p} + \frac{N+i+1-\rho_1^{N+1}-i\rho_1^i}{1-\rho_1}\rho_0}, \\ L_1 &= \left[\frac{1}{(1-\rho_1)}\left[\frac{(N+2)(N+1)}{2}\right.\right. \\ &\quad - \frac{\frac{\rho_1-\rho_1^{N+2}}{1-\rho_1} - (N+1)\rho_1^{N+2}}{1-\rho_1} \\ &\quad \left.\left. + \frac{(2+i)(i-1)\left(1-\rho_1^{N+1}\right)}{2}\right.\right. \\ &\quad + \left(\frac{(2+i)(i-1)}{2}\rho_1 - \frac{\rho_1 + \frac{\rho_1^2-\rho_1^{i+1}}{1-\rho_1} - i\rho_1^{i+1}}{1-\rho_1}\right)\cdot \rho_1^N \\ &\quad \left. - \frac{(2+i)(i-1)}{2} - \frac{\rho_1 + \frac{\rho_1-\rho_1^i}{1-\rho_1} + i\rho_1^i}{1-\rho_1}\right] \end{aligned}$$

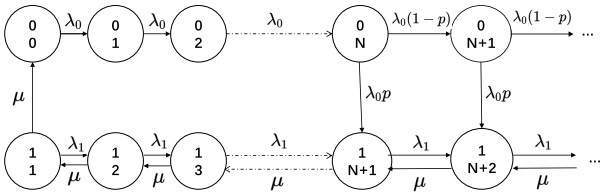


FIGURE 2. State-transition-rate diagrams for the N policy and p wake-up M/M/1 queuing system.

$$- \frac{(2+i)(i-1)(1-p)}{2} + \frac{(1-p) + \frac{(1-p)-(1-p)^i}{p} + i(1-p)^i}{p}]\rho_0 p\langle 0,0\rangle. \quad (8)$$

C. POWER COMSUMPTION

In this subsection, we define the total expected energy consumption function as F_{total}. Our goal is to establish an energy consumption function that is based on the system parameters and provides an effective measure of the energy consumption of the system.

Denote the average energy consumption of an edge node switching from a dormant to an wake-up state per unit time by E_{SR}, and by E_{RS} the average energy consumption of switching from an excited to a dormant state, where e_{sr} and e_{rs} are both system energy consumption parameters, e_{sr} is the power consumption of the edge node each time it switches from the dormant to the wake-up state, and e_{rs} is the power consumption of the edge node each time it switches from the excited to the dormant state.

$$\begin{aligned} E_{SR} &= p\sum_{k=N}^{i} P_{\langle 0,k\rangle}\lambda_0 e_{sr} \\ &= \left[1-(1-p)^{i+1}\right]\lambda_0 e_{sr} p\langle 0,0\rangle \\ &= \frac{\left[1-(1-p)^{i+1}\right]\lambda_0 e_{sr}}{N - \frac{(1-p)^{i-N+1}+(1-p)^i-2(1-p)}{p} + \frac{i+1-\rho_1^{N+1}-i\rho_1^i}{1-\rho_1}\rho_0}, \\ E_{RS} &= \mu P_{\langle 1,1\rangle} e_{rs} \\ &= \mu\rho_0 P_{\langle 0,0\rangle} e_{rs} \\ &= \frac{\mu\rho_0 e_{rs}}{N - \frac{(1-p)^{i-N-1}+(1-p)^i-2(1-p)}{p} + \frac{N+i+1-\rho_1^{N+1}-i\rho_1^i}{1-\rho_1}\rho_0}. \end{aligned} \quad (9)$$

For simplicity, we assume:

e_h = the holding power of a single packet in the system,

e_{id} = energy consumption of edge nodes in idle periods,

e_b = energy consumption of edge nodes in busy periods.

Using the power consumption parameters defined above, we can derive the expression for the energy consumption function. Our objective is to minimize the following function

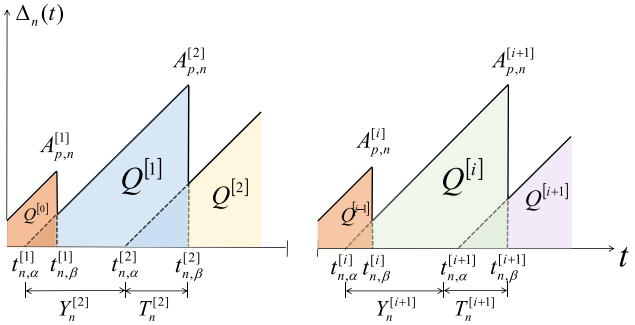


FIGURE 3. Age of information of first-in-first-out (FIFO) single queue.

by finding the threshold N and wake-up probability p.

$$F_{total} = e_h L_N + e_{id}\frac{E[I_N]}{E[T_N]} + e_b\frac{E[B_N]}{E[T_N]} + E_{SR} + E_{RS}, \quad (10)$$

where $E[I_N]$, $E[T_N]$, $E[B_N]$, E_{SR}, E_{RS} are given in (7) and (9).

D. PEAK AGE OF INFORMATION

To minimize energy consumption, we study the PAoI in the context of frequency-division multiple access (FDMA),

$$H_{ub,n}^{T,[i]}(t) = \frac{D\alpha_t^{[i]}}{G\alpha_b^{[i]}}\log_2\left(1+\frac{\psi_{t,n}^{[i]}(t)\left|r_{t,n}(t)\right|^2}{(D/G)G_0}\right), \quad (11)$$

where $H_{ub,n}$ denotes the upper limit of channel capacity and D denotes the available bandwidth. $\psi_{t,n}$ represents the transmit power of the edge nodes and G_0 is the noise spectral density, $\left|r_{t,n}(t)\right| = 10^{-3}\theta_0 d_n^{-a_0}$ is the channel coefficient of the wireless link between the medical centre and the nth edge node at time t, where θ_0 and d_n denote the small-scale fading parameter and the link length, respectively. At the same time, a_0 indicates the path loss of the link.

The AoI of the nth edge node at the i-th state update $\Delta_n(t)$ is shown in Figure 3 and can be represented as $A_n^{[i]}(t) = t - t_{n,\alpha}^{[i]}$, where $t_{n,\alpha}^{[i]}$ and $t_{n,\beta}^{[i]}$ denote the moment of generation and transmission of the ith state information, respectively. In addition to this, $A_{p,n}^{[i]}$ is the instantaneous PAoI of the nth edge node at the i-th state update. Since the optimization process for updating each state is the same, we have omitted the superscript "[i]" in subsequent equations for simplicity:

$$\mathbb{E}\left[A_p\right] = \mathbb{E}\left[t_{n,\beta}^{[i]} - t_{n,\alpha}^{[i-1]}\right] = \mathbb{E}[Y_n] + \mathbb{E}[T_n]. \quad (12)$$

$Y_n^{[i+1]}$ and $T_n^{[i+1]}$ respectively represent the inter-arrival time of data generation and the system delay of a data packet during the $i+1$th state update. Therefore, $\mathbb{E}[Y_n]$ and $\mathbb{E}[T_n]$ denote the average inter-arrival time of data generation and the average system delay per data packet, respectively. Consequently, $\mathbb{E}[Y_n] = \lim_{T\to\infty} T / \left(\sum_{i=1}^{\infty} \mathbb{1}_{\left\{t_{n,\alpha}^{[i]} < T\right\}}\right)$,

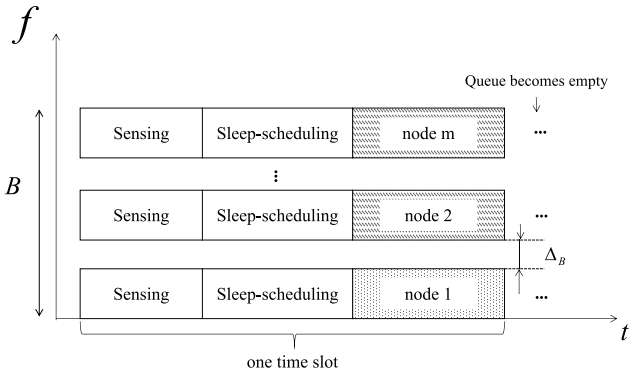


FIGURE 4. An example that illustrates the FDMA with sleep-scheduling.

where $\mathbb{1}_{\cdot}$ represents the indicator function. Additionally, we employ the term "per-packet AoI" to evaluate the performance of edge-assisted smart healthcare systems in relation to the AoI associated with each packet. In accordance with Study [40], the "per-packet AoI" for the ith status update can be determined by calculating the area of $Q^{[i]}$ in Figure 3 in the following manner:

$$\bar{A}_n^{[i]} = \frac{Q^{[i]}}{t_{n,\beta}^{[i]} - t_{n,\beta}^{[i-1]}} = \frac{Q^{[i]}}{2\left(Y_n^{[i+1]} + T_n^{[i+1]} - T_n^{[i]}\right)} = \frac{T_n + A_p}{2} \tag{13}$$

In the context of the FDMA environment, the available bandwidth is divided into m orthogonal frequency subchannels, with a guard bandwidth of $\Delta_B = 10$. Each subchannel is allocated to a single edge node, allowing them to share the same channel. The data is transmitted to the aggregator node. As depicted in Figure 4, a FDMA time slot encompasses the data sensing by the edge nodes, the implementation of the sleep scheduling policy (referred to as the N policy and the p wake-up policy), and data processing until the queue becomes empty.

As an effective measure of information freshness, the PAoI is a better indicator of the long-term behaviour of the AoI process than the AoI. PAoI denotes the average maximum duration after receiving the latest update packet, indicating the extent to which the propagation of update information is delayed. Based on the $M/M/1$ queuing model we have developed and the little formula, the average system delay based on the sleep scheduling strategy can be expressed as $E[T_n] = (\lambda_0 P_I + \lambda_1 P_B)^{-1} L_N$, where L_N represents the expected total number of data packets in the system, and $L_N = L_0 + L_1$. $E[T_n]$ represents the average system delay, λ_0 and λ_1 denote the arrival rates of different types of data packets, P_I signifies the probability of an idle slot, P_B represents the probability of a busy slot, and L_N represents the expected length of data packets. By substituting equation (8) into equation (12), we can obtain the average time interval for state updates as $E[Y_n] = (\lambda_0 P_I + \lambda_1 P_B)^{-1}$.

Furthermore, the average for the nth edge node, employing the sleep scheduling strategy, can be expressed as follows:

$$E\left[A_{p,n}\right] = E[Y_n] + E[T_n] = \frac{L_N + 1}{\lambda_0 P_I + \lambda_1 P_B}. \tag{14}$$

In light of the aforementioned discourse, we can articulate the average value of the PAoI as follows:

$$E\left[A_{p,n}\right] = A_{peak}(\lambda_0, \lambda_1, \mu, i, N, p), \tag{15}$$

where the definition of the function $A_{peak}(\lambda_0, \lambda_1, \mu, i, N, p)$ is as illustrated in Equation (14). For the sake of convenience, we shall denote the function $A_{\text{peak}}(\lambda_0, \lambda_1, \mu, i, N, p)$ as A_{peak}.

E. PROBLEM FORMULATION

In edge node-assisted smart healthcare systems, we focus on the battery's energy performance metrics and PAoI. We focus on the node's energy consumption performance metric and PAoI, so our goal is to obtain the optimal N policy and p wake-up policy that minimizes both node energy consumption and PAoI. We referred to study [34] and considered the significance of optimizing the objective function and the differences in magnitude and dimensionality. In the actual optimization process, we unified them into a single-objective form using weighting coefficients w_1 and w_2, as follows:

$$\begin{aligned} &\min_{N,p} w_1 F_{total} + w_2 A_{peak} \\ &\text{s.t. } (4) \sim (15), 0 < p < 1, N < i. \end{aligned} \tag{16}$$

We can observe that the objective function includes the energy consumption function represented by F_{total} and the average PAoI represented by A_{peak}. It is worth noting that both F_{total} and A_{peak} exhibit non-convexity, which results in a non-convex optimization problem. Additionally, the problem involves multiple objectives, making it a multi-objective optimization problem, which further increases the complexity of finding a solution.

IV. DRL-BASED SLEEP SCHEDULING FRAMEWORK

In this section, we will introduce two sleep scheduling frameworks based on DDQN and DQN-DDPG, respectively. The framework structures of both frameworks are illustrated in Figure 5, respectively.

A. DRL DESIGN IN THE SLEEP SCHEDULING SYSTEM

Reinforcement learning (RL) is a significant area of machine learning (ML) that leverages continuous interaction to gather information about an unknown system and improve its operating strategy through trial and error. As a result, RL does not necessitate a mathematical model or any prior knowledge of the system.

To solve RL problems, it is necessary to establish an MDP model, which primarily includes state space, action space, and reward. For the sleep scheduling system that we have developed, the definitions of these components are as follows.

• **Status space:** During each decision period, the edge nodes occupy a state. We represent the possible state set of peripheral nodes using S. The state variable denotes the state of the node (dormant or awake) as well as the number of data packets.

$$S = \{S : S = \langle \delta, i \rangle \quad i \geqslant 0\}, \tag{17}$$

where $\delta \in \{0, 1\}$, $\delta = 0$ indicates that the node is in a dormant state, $\delta = 1$ indicates that the node is in a wake-up state, i indicates the amount of data in the node, $i \geqslant 0$.

• **Action Space:** The action space comprises of two actions: selecting a threshold N and setting an wake-up probability p, so the action of TS t is defined as

$$a_t = \left\{a_t^1, a_t^2\right\}, \tag{18}$$

where $a_t^1 \in \mathcal{A}_1$ represents the action of adjusting the threshold N selection, and $a_t^2 \in \mathcal{A}_2$ represents the action of selecting the probability p.

• **Reward function:** The objective of this study is to minimize both energy consumption and average PAoI, so both factors are taken into account in the system's reward setting. However, the goal of reinforcement learning is to maximize the cumulative discounted reward. Therefore, the reward in time slot t is defined as follows

$$r_{t+1} = -w_1 F_{total} - w_2 A_{peak}, \tag{19}$$

where w_1 and w_2 are the corresponding weighting coefficients. Based on this, the long-term rewards are as follows

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{20}$$

where $\gamma \in [0, 1]$ is a discount factor that is used to weigh the future rewards against the current rewards.

B. DDQN BASED SLEEP SCHEDULING FRAMEWORK

In RL, in order to maximize the long-term reward, an optimal policy π needs to be found. π denotes the probability of mapping from any state s to action a, which tells the intelligence how it should choose an action in state s to achieve the desired R. Given a policy π, we use the Q function to evaluate the effect of taking action a in the current state s. The function with the highest Q-value is called the optimal Q-value function $Q_{\pi^*}(s_t, a_t)$, which is defined as

$$\begin{aligned} Q_{\pi^*}(s_t, a_t) &= \max_{\pi} Q_{\pi}(s_t, a_t) \\ &= \max_{\pi} \mathbb{E}\left[R_t \mid s_t, a_t\right] \\ &= \max_{\pi} \mathbb{E}\left[r_t + \gamma R_{t+1} \mid s_t, a_t\right] \\ &= \max_{\pi} \mathbb{E}\left[r_t + \gamma Q_{\pi}(s_{t+1}, a_{t+1}) \mid s_t, a_t\right] \\ &= \mathbb{E}\left[r_t + \gamma \max_{a_{t+1}} Q_{\pi^*}(s_{t+1}, a_{t+1}) \mid s_t, a_t\right]. \end{aligned} \tag{21}$$

The recursive formula (24) is also known as the Bellman equation. However, the Bellman equation requires the calculation of the expectation of the entire state space, which makes it impossible to calculate to obtain $Q_{\pi^*}(s_t, a_t)$. To solve this problem, function approximation is used in conventional RL to approximate the expectation, while DRL combines RL with deep neural networks (DNNs) to approximate the function. DRL has powerful computational power and performs better than RL for problems with large state space and action space.

DQNs, which are a combination of Q-learning and deep neural networks, are powerful structures that can learn from and explore their environment. Through Q-learning, the system earns rewards for each action taken, which is used to populate Q- and V-tables. Deep Q learning updates the model parameters of the Q function by iteratively calculating the loss function between predicted and target Q values, and back-propagating to improve the model.

As depicted in Figure 5, DDQN-based sleep scheduling framework consists of two DQN units. Each DQN unit is composed of two networks: a Q-network $Q(s, a; \chi)$ for estimating the Q-value of the selected action, and a target Q-network $Q(s, a; \chi^-)$ for generating the target Q-values used in training, where χ and χ^- represent the weights of their neural networks.

At the start of each TS, s_t is transmitted to the DQN unit of the threshold N selection network. Given s_t as input, the Q-network of this unit outputs the Q-estimate value $Q\left(s_t, a_t^1; \chi\right)$ for action $a_t^1 \in \mathcal{A}_1$. After obtaining Q estimates $Q\left(s_t, a_t^1; \chi\right)$ for all actions, we use $\epsilon - greedy$ policy to determine the threshold N for selecting action a_t^1, balancing the exploration of new actions with the exploitation of known ones. Specifically, we randomly select action a_t^1 from $\mathcal{A}_1$ with probability ϵ, or select the action a_t^1 with the highest estimated Q value with probability $1 - \epsilon$, as follows.

$$a_t^1 = \arg \max_{a_t^1 \in \mathcal{A}_1} Q\left(s_t, a_t^1; \chi\right), \tag{22}$$

where $0 < \epsilon < 1$, through this strategy and reward function, enables the DQN unit to explore unselected actions that may have better rewards, thereby exploring the entire action space and updating the corresponding Q values. Meanwhile, the unit responsible for selecting the awakening probability p remains active. Input the state s_t, and obtain the action a_t^2 selected based on the awakening probability using the same method.

After generating the threshold-selected and awakening probability-selected actions, execute action $a_t = \left\{a_t^1, a_t^2\right\}$. The packet threshold N and wake-up probability p for edge nodes are updated based on a_t, and the DRL network structure calculates the reward r_t according to Equation (18) and obtains a new state s_{t+1}. Using the experience replay strategy, we place each training sample (s_t, a_t, r_t, s_{t+1}) in the memory buffer O within each time step (TS). In each TS, we randomly select N samples from the buffer to update χ.

To adjust the network for the threshold N selection, we utilize randomly sampled training samples (s_t, a_t, r_t, s_{t+1}) to obtain the target Q-value generated by the unit based on the Q-network, as follows

$$y_i = r_i + \max_{a_{i+1}^1 \in \mathcal{A}_1} Q\left(s_{i+1}, a_{i+1}^1; \chi^-\right). \tag{23}$$

We train the Q-network of the unit by minimizing the loss function.

$$L(\chi) = \left(y_i - Q\left(s_i, a_i^1; \chi\right)\right)^2. \tag{24}$$

For the selection of the wake-up probability p for the unit, we utilize the same method to calculate the target Q-value and the loss function.

$$y_i = r_i + \max_{a_{i+1}^2(m) \in \mathcal{A}_2} Q\left(s_{i+1}, a_{i+1}^2; \chi^-\right),$$
$$L(\chi) = \left(y_i - Q\left(s_i, a_i^2; \chi\right)\right)^2. \tag{25}$$

For the DQN unit that selects the wake-up probability p, it updates the weights χ^- of the target Q-network by copying the weights of the Q-network in each TS. Based on the DDQN sleep scheduling network framework, the algorithm is shown in Algorithm 1.

C. DQN-DDPG BASED SLEEP SCHEDULING FRAMEWORK

In this section, we will introduce the sleep scheduling framework based on DQN-DDPG. The DQN-DDPG framework is an improvement over the DDQN framework. For the unit that selects the wake-up probability p, we utilize the DDPG network to directly output the wake-up probability. DDPG can effectively handle continuous action spaces, thus solving the dimensionality problem.

Moreover, the utilization of a DDPG network enabled us to accomplish a more sophisticated and nuanced optimization of policy selection, ultimately resulting in enhanced system performance. It is noteworthy that this approach necessitated a greater amount of training data and a more intricate network architecture as compared to the DQN method. Nevertheless, the DDPG network's superior performance endows it with great potential as a valuable tool for applications that necessitate precise control of continuous actions.

The DDPG network is comprised of four sub-networks: the actor network $\pi(s; \mu)$ which selects an action to maximize the Q value of the output, the critic network $Q(s, a; \theta)$ which predicts the Q value, and the corresponding target actor network $\pi\left(s; \mu^-\right)$ and target critic network $Q\left(s, a; \theta^-\right)$ which generate the target values for training. Here, μ, θ, μ^-, and θ^- represent their respective weights. The DDPG network operates in an actor-critic fashion where the actor network updates its parameters through the DPG, selecting the optimal action in the current state, and the critic network evaluates the action chosen by the actor network. In the sleep scheduling system, when an edge node receives a packet, it transmits status

Algorithm 1 DDQN-Based Edge Node Sleep Scheduling

1: Initialize the replay memory O
2: Initialize the Q network $Q(s, a; \chi)$ and target Q network $Q\left(s, a; \chi^-\right)$ with initial weights $\chi^- = \chi$
3: Initialize the terminating TS T_{max}, weights update interval W,
4: **for** $j = 1, 2, \ldots\ldots, Kmax$ **do**
5: DQN unit selects the action $a_t^1 \in \mathcal{A}_1$ following the $\epsilon - greedy$ policy.
6: DDPG unit selects the action $a_t^2 \in \mathcal{A}_2$ following the $\epsilon - greedy$ policy.
7: Obtain a reward r_t, and then the state is transited to s_{t+1}.
8: Store the tuple sample (s_t, a_t, r_t, s_{t+1}) into the memory O.
9: **if** O is full **then**
10: Sample a random mini-batch of N tuples (s_i, a_i, r_i, s_{i+1}) from memory O;
11: Update the weights by minimizing the loss function .
12: For the threshold N selection DQN unit, update its weights by minimizing the loss function (24).
13: For wake-up probaility p-selection unit, update their weights by minimizing the loss function (25)
14: **end if**
15: Update the status of edge nodes as $s_t \longrightarrow s_{t+1}$
16: **end for**
Output: Policy π

information to the DRL-based sleep scheduling framework. Upon receiving this information, the DDPG unit generates a deterministic wake-up probability p for the assignment action $a_t^2 = \pi(s_t; \mu)$, based on the weights μ and the current state s_t.

In order to add the exploration of new actions while ensuring the known actions. Similar to the $\epsilon - greedy$ strategy in DQN, we add the random noise to the initial output action as follows

$$a_t^2 = [\pi(s_t; \mu) + \mathcal{N}]_0^1, \tag{26}$$

where $\mathcal{N}$ denotes a random noise process that obeys a normal distribution. a_t^2 is then restricted to the (0, 1) interval.

After executing the action generated by the DDPG unit, the DRL-based sleep scheduling framework receives the reward and the system state moves to s_{t+1}. We store (s_t, a_t, r_t, s_{t+1}) in the relay memory $O(t)$.

The critic network $Q(s, a; \theta)$ can estimate the Q value of the selected action, which is equal to $Q(s, \pi(s; \mu); \theta)$. And the actor network $\pi(s_t; \mu)$ updates its weight to obtain a larger cumulative discount reward, as follows

$$\nabla_\mu J(\pi) = \mathbb{E}_{s \sim \rho^\pi}\left[\nabla_\mu \pi(s; \mu) \nabla_a Q(s, a; \theta)\big|_{a=\pi(s;\mu)}\right]. \tag{27}$$

Using the N sample sets selected in $O(t)$, we can approximate the expectation

$$\nabla_{\mu} J(\pi) \approx \frac{1}{N} \sum_{i} \left[\nabla_{\mu} \pi(s; \mu)\big|_{s=s_i} \nabla_a Q(s, a; \theta)\Big|_{s=s_i, a=\pi(s_i;\mu)} \right]. \tag{28}$$

Using the target actor network $\pi\left(s; \mu^{-}\right)$ and target critic network $Q\left(s, a; \theta^{-}\right)$, the target Q values generated based on random group training are

$$y_i = r_i + \gamma Q\left(s_{i+1}, \pi\left(s_{i+1}; \mu^{-}\right); \theta^{-}\right). \tag{29}$$

On the basis of this, the critic network $Q(s, a; \theta)$ updates its weights by minimising the loss function, which is defined as follows

$$L(\theta) = \frac{1}{N} \sum_{i} \left(y_i - Q\left(s_i, a_i^2; \theta\right)\right)^2. \tag{30}$$

To summarize, within the context of the wakeup probability assignment network, the actor network denoted by $\pi(s; \mu)$ takes the state s_t as input and produces the action a_t^2, while updating the parameter μ using equation (28). On the other hand, the critic network $Q(s, a; \theta)$ takes in the state s_i, outputs the Q value, and updates the parameter θ through equation (30). The target actor network $\pi\left(s; \mu^{-}\right)$ and target critic network $Q\left(s, a; \theta^{-}\right)$ input to the tuple in $O(t)$, and output to compute the target Q value in (29). Simultaneously, their weights μ^{-} and θ^{-} are updated in a gentle manner to ensure learning stability, as follows:

$$\begin{aligned} \theta^{-} &\leftarrow \tau\theta + (1-\tau)\theta^{-}, \\ \mu^{-} &\leftarrow \tau\mu + (1-\tau)\mu^{-}, \end{aligned} \tag{31}$$

where $0 < \tau \ll 1$. Algorithm 2 briefly outlines the details of DRL-based sleep scheduling framework.

Algorithm 2 DQN-DDPG Based Edge Node Sleep Scheduling

1: Initialize the Q-network $Q(s, a; \chi)$ as the DQN unit for selecting the threshold N with the weight a.
2: Initialize the actor network $\pi(s; \mu)$ and the critic network $Q(s, a; \theta)$ of the wake-up probability p assignment DDPG unit with weights μ and θ.
3: Initialize target Q network $Q\left(s, a; \chi^{-}\right)$, target actor network $\pi\left(s; \mu^{-}\right)$ and target critic network $Q\left(s, a; \theta^{-}\right)$ with initial weights $\chi^{-} = \chi$, $\mu^{-} = \mu$ and $\theta^{-} = \theta$.
4: Initialize the terminating TS T_{max}, weights update interval W, replay memory O, the random noise process $\mathcal{N}$.
5: **for** $t = 1, 2, \ldots, T_{\max}$ **do**
6: DQN unit selects the action $a_t^1 \in \mathcal{A}_1$ following the $\epsilon - greedy$ policy.
7: DDPG unit selects the action $a_t^2 \in \mathcal{A}_2$ according to (25).
8: Obtain a reward r_t, and then the state is transited to s_{t+1}.
9: Store the tuple sample (s_t, a_t, r_t, s_{t+1}) into the memory O.
10: **if** O is full **then**
11: For DQN unit, update its weights by minimizing the loss function (24).
12: For DDPG unit, the actor network updates μ according to (28), and the critic network updates θ according to (30).
13: Update χ^{-} of the DQN unit by copying χ in every W TS.
14: Update θ^{-} and μ^{-} of the DDPG unit according to (31).
15: **end if**
16: **end for**
Output: Policy π

D. TIME AND SPACE COMPLEXITY ANALYSIS

In this section, we have conducted a comprehensive analysis of the time and space complexity of the two proposed DRL-based sleep scheduling algorithms.

1) TIME AND SPACE COMPLEXITY ANALYSIS OF THE EDGE NODE SLEEP SCHEDULING ALGORITHM BASED ON DDQN

Time Complexity:

Step 1: Initialization of the replay memory. The time complexity of this operation can be considered negligible, O(1).

Step 2: Initialization of the weights for the Q-network and target Q-network. The time complexity of this operation can be considered negligible, O(1).

Step 3: Initializing termination threshold T_{max} and weight update interval W also has a time complexity of O(1), which can be considered negligible.

Steps 4-16: Repeat K_{max} times, performing a series of operations each time. Steps 5-7: Select action, obtain reward, and transition state. The time complexity of these operations can be considered negligible (O(1)).

Step 8: Store the sample in memory. The time complexity of this operation can be considered negligible (O(1)).

Steps 9-14: If the memory is full, perform weight updates. For the samples in memory, execute the minimization operation of the loss function. The time complexity of these operations depends on the computational complexity of the loss function, assumed to be O(n).

Step 15: Update the state of the edge node. The time complexity of this operation can be considered negligible (O(1)).

The time complexity of the entire loop, which is executed Kmax times, can be approximated as $O(K_{max}^n)$.

Based on the analysis provided, the overall time complexity of the algorithm can be approximated as $O(K_{max}^n)$.

Space Complexity:

1. The space complexity of the replay memory, denoted as O, depends on its size, assumed to be O(m), where m represents the capacity of the memory.

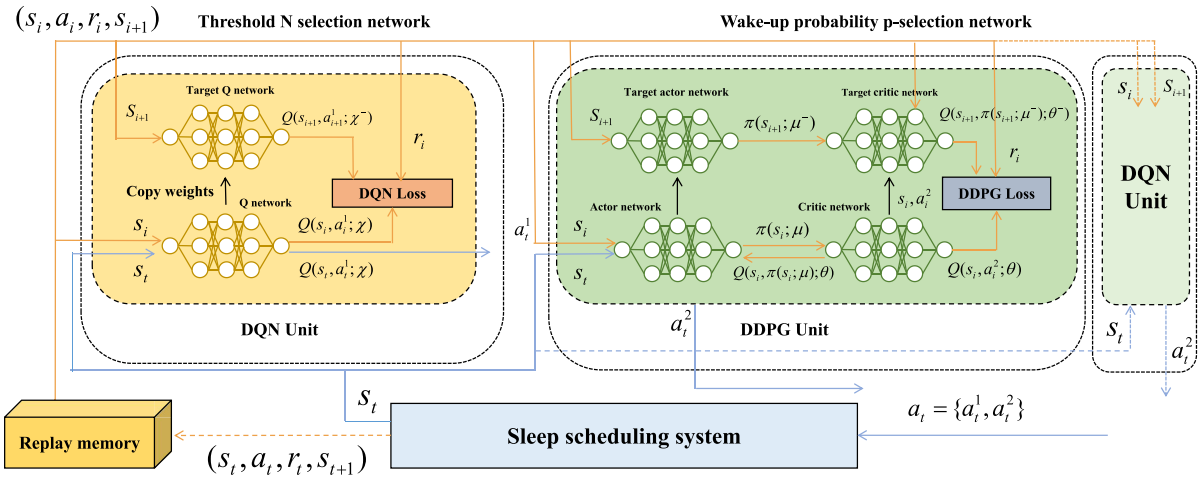


FIGURE 5. DRL-based sleep scheduling framework.

2. The space complexity of the Q-network and target Q-network depends on the network structure and the number of parameters, assumed to be O(r), where r represents the number of parameters in the network.

3. The space complexity of other variables and data structures can be considered negligible since their sizes are fixed and can be represented as O(1).

In conclusion, the overall space complexity of the algorithm can be approximated as O(m + r), considering the combined space requirements of the replay memory (m) and the Q-network and target Q-network parameters (r).

2) TIME AND SPACE COMPLEXITY ANALYSIS OF THE EDGE NODE SLEEP SCHEDULING ALGORITHM BASED ON DQN-DDPG

Time Complexity:

Step 1: Initialize the Q-network. The time complexity of this operation can be considered negligible (O(1)).

Step 1: Initialize the Q-network. The time complexity of this operation can be considered negligible (O(1)).

Step 2: Initialize the actor network and critic network. The time complexity of this operation can be considered negligible (O(1)).

Step 3: Initialize the target Q-network, target actor network, and target critic network. The time complexity of this operation can be considered negligible (O(1)).

Steps 5-16: Iterate for T_{max} times, performing a series of operations. Steps 6-8: Select action, obtain reward, and transition state. The time complexity of these operations can be considered negligible (O(1)).

Step 9: Store the sample in the replay memory. The time complexity of this operation can be considered negligible (O(1)).

Steps 10-14: If the replay memory is full, perform weight updates. For DQN units and DDPG units, this involves minimizing the loss function and updating the weights. The time complexity of these operations depends on the computational complexity of the loss function, assumed to be O(n).

The loop is executed T_{max} times, so the overall time complexity of the loop is O(T_{max}^n).

Overall, the code has an approximate time complexity of O(T_{max}^n).

Space Complexity:

1. The space complexity of the Q-network, actor network, and critic network depends on their network structure and the number of parameters, assumed to be O(r), where r represents the number of parameters in the network.

2. The space complexity of the replay memory, denoted as O, depends on its size, assumed to be O(m), where m represents the capacity of the replay memory.

3. The space complexity of other variables and data structures can be considered negligible as their sizes are fixed.

In summary, the overall space complexity of the algorithm can be approximated as O(r + m), considering the combined space requirements of the network parameters (r) and the replay memory (m).

V. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed DQN-DDPG-based resource allocation scheme implemented in smart healthcare with the assistance of edge nodes through simulation. The main simulation parameters are shown in Table 2 and the rest of the simulation parameters are mentioned in the text. The structure of the developed neural network consists of an input layer, two hidden layers and an output layer. The number of neurons in the hidden layers are 10 and 20 respectively. The Rectified Linear Unit (ReLU) function, $f(x) = \max(0, x)$, is chosen as the activation function for all the hidden layers. According to [41], τ in (31) is 0.01, and the noise process in (26) follows $\mathcal{N}(0, 1)$, while other parameters are set based on [42]: learning rate $\beta = 0.001$, $\epsilon = 0.9$, memory capacity $|\mathcal{O}| = 5000$, weights

TABLE 3. Default simulation parameters.

Parameter	Value
Packet arrival rate in the dormant state (λ_0)	0.1
Packet arrival rate in the wake-up state (λ_1)	0.3
Data processing rate (μ)	0.5
The holding power of a single packet in the system (e_h)	1
Energy consumption of edge nodes in idle periods (e_{id})	2
Energy consumption of edge nodes in busy periods (e_b)	200
The power consumption of the edge node each time it switches from the dormant to the wake-up state (e_{sr})	15μw
The power consumption of the edge node each time it switches from the excited to the dormant state (e_{rs})	0.5μw

FIGURE 6. Cumulative discounted rewards under different DRL methods.

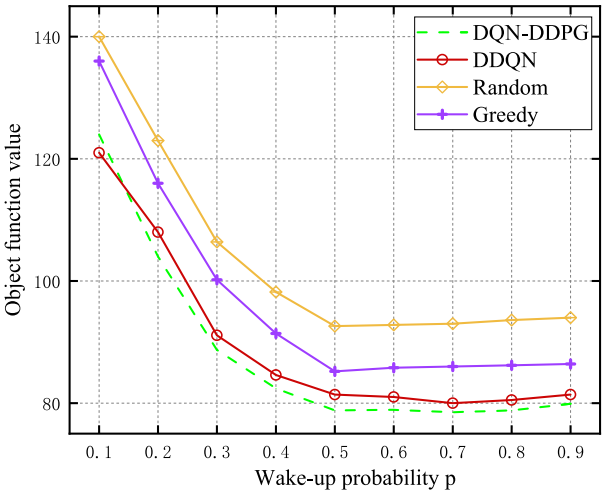


FIGURE 7. Under the condition that $N < 20$, various algorithms optimize the performance of the objective function.

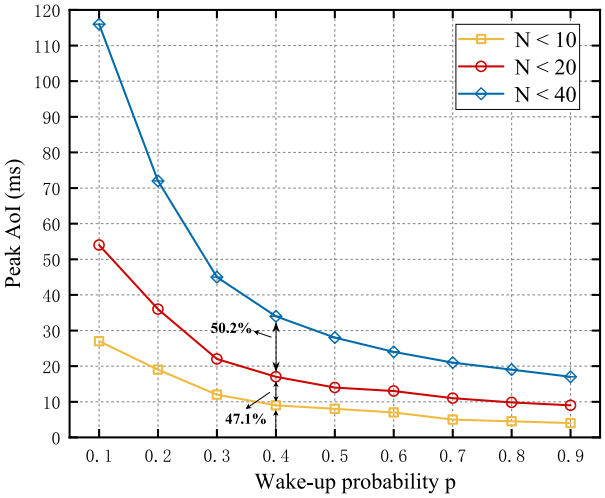


FIGURE 8. The impact of wake-up probabilities on average PAoI with varying limitation ranges at threshold N.

update interval $W = 10$, batch size $N = 32$. In order to compare the performance of different algorithms, the average test results are obtained from 2000 episodes, each consisting of 20 steps.

Figure 6 evaluates the effect of different DRL methods on reward convergence, where the threshold $N < 10$, $\lambda_0 = 0.5, \lambda_1 = 0.6, \mu = 0.7$, the learning rate $\beta = 0.001$, and $\epsilon = 0.9$. From the figure we can see that the DQN-DDPG algorithm converges faster compared to the DDQN algorithm, and its reward is better than the DDQN algorithm for obtaining a locally optimal solution, thus illustrating the optimality of the performance of our proposed DQN-DDPG algorithm. Meanwhile, Figure 7 compares the performance of greedy strategy and random strategy with DDQN and DQN-DDPG algorithms. As shown in Figure 7, we can observe that as the wake-up probability p changes, the DQN-DDPG algorithm performs better in optimizing the objective function than the DDQN algorithm. The average objective function obtained using the DQN-DDPG method compared to that obtained using the DDQN method, the greedy algorithm and the stochastic strategy was reduced by 1.8%, 9.1% and 14.9%, respectively, within the same threshold N limit. Therefore, we consider utilizing the DQN-DDPG algorithm to investigate the impact of different performance parameters on optimizing the objective.

In Figure 8, we evaluated the relationship between the average PAoI and the wake-up probability p based on the DQN-DDPG method. We considered the optimal solutions based on DRL for average PAoI and energy consumption when the data packet threshold N is limited to 10, 20, 40, and 60, respectively. After the number of data packets reaches the threshold N, data transmission and processing are performed with a wake-up probability of p. As shown in Figure 6, when the wake-up probability is 0.1, the average PAoI is the highest. When the wake-up probability reaches 0.3 and 0.5, the average PAoI correspondingly decreases. However, when the wake-up probability reaches 0.9, the difference in

average PAoI between 0.7 is not significant. When the wake-up probability is low, the nodes are mostly in a sleeping state, causing a large accumulation of data packets and resulting in lower information freshness and a larger average PAoI. When the wake-up probability increases, most nodes are in the wake-up state, and the speed of data packet transmission and processing is faster, resulting in a significant decrease in the average PAoI. However, the data processing capacity and range of edge nodes are limited, and even if almost all nodes are in a working state, timely processing of all data packets may not be possible. Therefore, when the wake-up probability is high, the average PAoI may not decrease significantly.

Meanwhile, we observed that the freshness of information decreases as the threshold N range expands. This is because a larger threshold N range causes data packets to accumulate, resulting in delayed transmission and processing of data, which increases average PAoI. As shown in Figure 8, when the wake-up probability is 0.8, average PAoI increases by 50.2% for $N < 40$ compared to $N < 20$ and by 47.1% for $N < 20$ compared to $N < 10$.

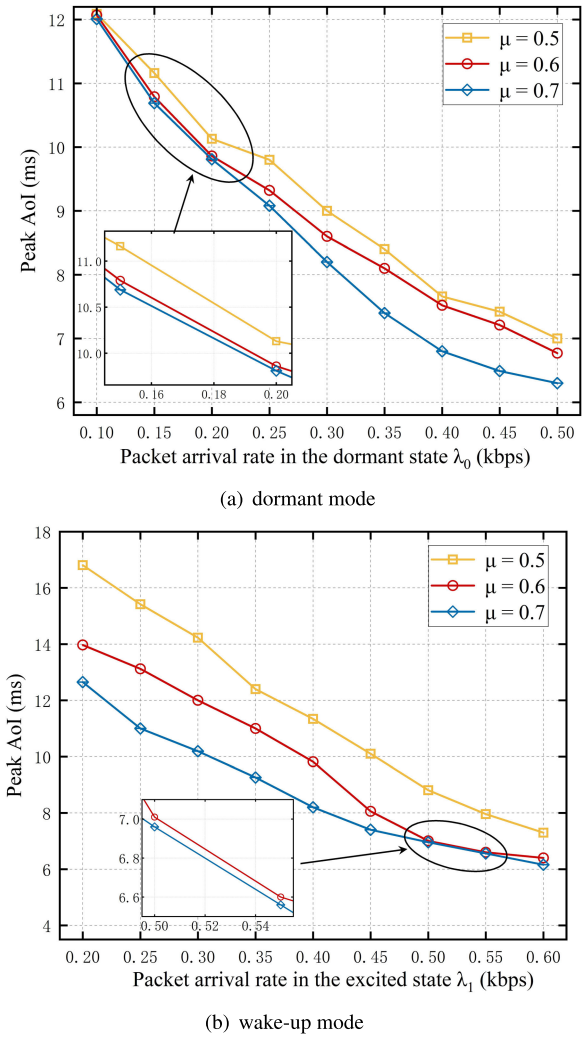


FIGURE 9. Average PAoI for different data transmission rates during dormant mode and wake-up mode.

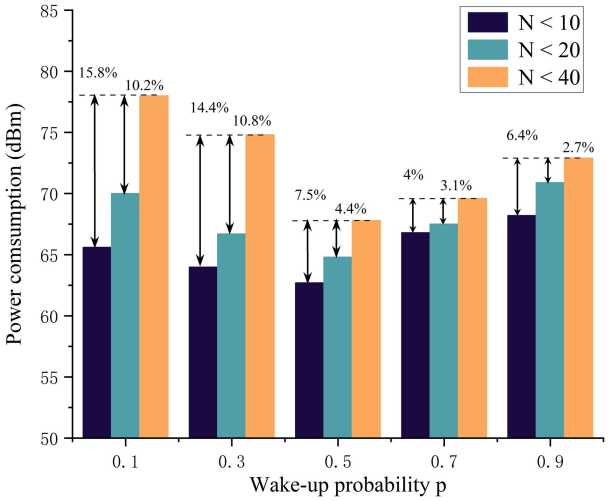
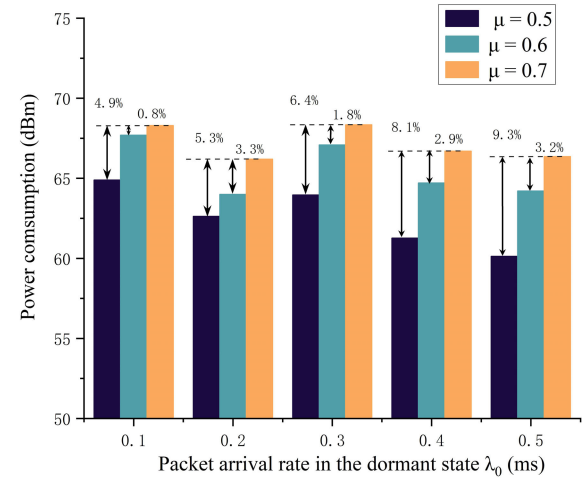


FIGURE 10. Energy consumption for various wake-up probabilities.

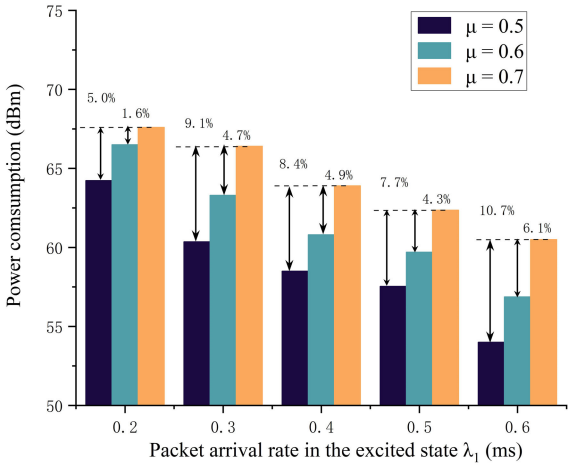
Figures 9 illustrate the impact of varying data transmission rates on the PAoI using the DQN-DDPG method in both sleep and wake-up states. As illustrated in Figure 9(a), increasing the data transmission rate in the dormant state from 0.1 to 0.4 results in a corresponding decrease in average PAoI. However, when the rate reaches 0.5, the difference in average PAoI compared to that at a rate of 0.4 is not significant. Elevating the data transmission rate during the dormant state expedites the number of data packets reaching the threshold N, ultimately increasing the probability p of the node transitioning to the wake-up state for data transmission and processing, which enhances information freshness. Nonetheless, a high data transmission rate during the dormant state is limited by the wake-up probability p and data processing rate, potentially limiting the extent to which average PAoI can decrease.

The pattern of average PAoI demonstrated in Figure 9(b) is similar to that in Figure 9(a). As the data transmission rate during the wake-up state increases from 0.2 to 0.45, the corresponding average PAoI decreases. However, the decrease in average PAoI is not significant when the rate reaches 0.6 compared to that at a rate of 0.45. Increasing the data transmission rate during the wake-up state facilitates the handling of more data packets, ultimately leading to higher work efficiency and a reduction in average PAoI. Nonetheless, a high data transmission rate during the wake-up state is limited by the wake-up probability p and the data processing rate, which may limit the extent to which average PAoI can decrease. Additionally, Figures 9 demonstrate that increasing the data processing rate can effectively enhance information freshness.

Figure 10 depicts the relationship between energy consumption and the wake-up probability p. It can be observed that energy consumption first decreases and then increases as the wake-up probability p increases. This is because the low wake-up probability at the initial stage results in a large accumulation of data packets, leading to an increase in the



(a) dormant mode



(b) wake-up mode

FIGURE 11. Energy consumption of various data transmission rates during dormant mode and wake-up mode.

required energy consumption to maintain the packets. As the wake-up probability increases, edge nodes enter the wake-up state with a probability of p, enabling data packet transmission and processing, which reduces the required energy consumption to maintain the packets. However, when the wake-up probability is too high, most nodes are in a working state, resulting in an increase in energy consumption. At the same time, it can be inferred from Figure 10 that energy consumption increases with the increase of the threshold range N, because as the threshold range N increases, the required energy for packet retention increases. When the wake-up probability is 0.5, the energy consumption of $N < 40$ is 7.5% and 4.4% higher than that of $N < 20$ and $N < 10$, respectively.

The figure depicted as Figure 11 illustrates the impact of varying data transmission rates on energy consumption through the application of the DQN-DDPG approach in both the sleep and wake-up states. The visual representation shows that elevating the data transmission rate during the dormant state barely affects energy consumption. On the other hand, during the wake-up state, a higher data transmission rate facilitates the processing of accumulated data packets, ultimately leading to a significant reduction in energy consumption.

It is worth noting that the processing rate of data packets has a significant impact on energy consumption. Energy consumption increases with an increase in the processing rate of data packets. This is because as the number of data packets decreases significantly, the system needs to increase the transmission rate of data packets to maintain queue stability, leading to an increase in energy consumption. When $\lambda_0 = \lambda_1 = 0.4$, as shown in Figure 11(a) in the dormant state, the energy consumption of $\mu = 0.5$ and $\mu = 0.6$ decreases by 6.4% and 1.8%, respectively, compared to $\mu = 0.7$. In the wake-up state, as shown in Figure 11(b), the energy consumption of $\mu = 0.5$ and $\mu = 0.6$ decreases by 8.4% and 4.9%, respectively, compared to $\mu = 0.7$. Therefore, it can be concluded that energy consumption increases with an increase in the processing rate of data packets.

Based on the research conducted by [43] and [44] regarding the impact of wearable devices on system energy consumption and AoI, as well as our analysis of the proposed operation of the smart healthcare system, we can conclude that with an increase in wearable devices, there is an increase in the amount of patient information collected. This leads to a significant increase in the number of data packets in the aggregation node, causing them to reach the threshold value N more quickly. As a result, the expected length of the busy period in the node increases, since the power consumption during the wake-up state is much higher than during the sleep state. Therefore, the energy consumption of the node increases.

Furthermore, with the increase in information volume, the expected length of data packets also increases. Since the data processing rate is fixed, the processing time for data packets becomes longer, resulting in an increase in average PAoI.

VI. CONCLUSION

In this work, since edge nodes have the advantage of low latency in information transmission, we have investigated an intelligent healthcare system assisted by edge nodes, where energy-limited edge nodes can perform real-time transmission and processing of patient vital signs information. In order to ensure timely information while minimizing energy consumption, we have studied a sleep scheduling strategy based on both threshold N and probability p wake-up using the DRL method. The simulation results have confirmed the effectiveness of the proposed DRL-based sleep scheduling strategy, as well as the impact of threshold N and wake-up probability p on the average PAoI. Furthermore, we verify that the proposed DQN-DDPG method performs better than the DDQN method, the greedy algorithm and the stochastic policy optimization, within the same threshold N limit.

REFERENCES

[1] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge computing for smart health: Context-aware approaches, opportunities, and challenges," *IEEE Netw.*, vol. 33, no. 3, pp. 196–203, May 2019.

[2] X. Hou, J. Wang, Z. Fang, Y. Ren, K.-C. Chen, and L. Hanzo, "Edge intelligence for mission-critical 6G services in space-air-ground integrated networks," *IEEE Netw.*, vol. 36, no. 2, pp. 181–189, Mar. 2022.

[3] A. M. Muhammad, A. M. Alsunbul, and A. M. Zeki, "Smart health using IoT: Challenges and solutions," in *Proc. Palestinian Int. Conf. Inf. Commun. Technol. (PICICT)*, Gaza, Palestine, Sep. 2021, pp. 1–5.

[4] Y. Xu, H. Zhang, H. Ji, L. Yang, X. Li, and V. C. M. Leung, "Transaction throughput optimization for integrated blockchain and MEC system in IoT," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1022–1036, Feb. 2022.

[5] Y. Zhao, K. Xu, H. Wang, B. Li, M. Qiao, and H. Shi, "MEC-enabled hierarchical emotion recognition and perturbation-aware defense in smart cities," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16933–16945, Dec. 2021.

[6] B. Zhu, K. Chi, J. Liu, K. Yu, and S. Mumtaz, "Efficient offloading for minimizing task computation delay of NOMA-based multiaccess edge computing," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3186–3203, May 2022.

[7] C. Roy, R. Saha, S. Misra, and D. Niyato, "Soft-health: Software-defined fog architecture for IoT applications in healthcare," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2455–2462, Feb. 2022.

[8] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1441–1456, May 2022.

[9] X. Wang, Z. Ning, S. Guo, M. Wen, and H. V. Poor, "Minimizing the age-of-critical-information: An imitation learning-based scheduling approach under partial observations," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3225–3238, Sep. 2022.

[10] J. Xu and N. Gautam, "Peak age of information in priority queuing systems," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 373–390, Jan. 2021.

[11] M. S. Kumar, A. Dadlani, M. Moradian, A. Khonsari, and T. A. Tsiftsis, "On the age of status updates in unreliable multi-source M/G/1 queueing systems," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 751–755, Feb. 2023.

[12] J. Xu, I.-H. Hou, and N. Gautam, "Age of information for single buffer systems with vacation server," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1198–1214, May 2022.

[13] H. Feng, J. Wang, Z. Fang, J. Qian, and K.-C. Chen, "Age of information in UAV aided wireless sensor networks relying on blockchain," *IEEE Trans. Veh. Technol.*, early access, Apr. 20, 2023, doi: 10.1109/TVT.2023.3268660.

[14] C. Zhou, G. Li, J. Li, Q. Zhou, and B. Guo, "FAS-DQN: Freshness-aware scheduling via reinforcement learning for latency-sensitive applications," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2381–2394, Oct. 2022.

[15] Z. Fang, J. Wang, C. Jiang, X. Wang, and Y. Ren, "Average peak age of information in underwater information collection with sleep-scheduling," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10132–10136, Sep. 2022.

[16] H. Lv, Z. Zheng, F. Wu, and G. Chen, "Strategy-proof online mechanisms for weighted AoI minimization in edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1277–1292, May 2021.

[17] B. A. G. R. Sharan, S. Deshmukh, S. R. B. Pillai, and B. Beferull-Lozano, "Energy efficient AoI minimization in opportunistic NOMA/OMA broadcast wireless networks," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 1009–1022, Jun. 2022.

[18] S. Wu, Z. Deng, A. Li, J. Jiao, N. Zhang, and Q. Zhang, "Minimizing age-of-information in HARQ-CC aided NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1072–1086, Feb. 2023.

[19] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 193–204, Mar. 2018.

[20] R. Rocchetta, L. Bellani, M. Compare, E. Zio, and E. Patelli, "A reinforcement learning framework for optimal operation and maintenance of power grids," *Appl. Energy*, vol. 241, pp. 291–301, May 2019.

[21] S. Leng and A. Yener, "Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[22] J. Wang, X. Cao, B. Yin, and Y. Cheng, "Sleep–wake sensor scheduling for minimizing AoI-penalty in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6404–6417, May 2022.

[23] S. Park, S. Jung, M. Choi, and J. Kim, "AoI-aware Markov decision policies for caching," in *Proc. 42nd IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Bologna, Italy, Jul. 2022, pp. 1274–1275.

[24] Z. Zhu, S. Wan, P. Fan, and K. B. Letaief, "Federated multiagent actor–critic learning for age sensitive mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1053–1067, Jan. 2022.

[25] J. Gong, J. Zhu, X. Chen, and X. Ma, "Sleep, sense or transmit: Energy-age tradeoff for status update with two-threshold optimal policy," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1751–1765, Mar. 2022.

[26] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, "AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1211–1223, Jan. 2021.

[27] S. Wang, M. Chen, Z. Yang, C. Yin, W. Saad, S. Cui, and H. V. Poor, "Distributed reinforcement learning for age of information minimization in real-time IoT systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 501–515, Apr. 2022.

[28] J. Huang, H. Gao, S. Wan, and Y. Chen, "AoI-aware energy control and computation offloading for industrial IoT," *Future Gener. Comput. Syst.*, vol. 139, pp. 29–37, Feb. 2023.

[29] H. B. Beytur and E. Uysal, "Age minimization of multiple flows using reinforcement learning," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Honolulu, HI, USA, Feb. 2019, pp. 339–343.

[30] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in RF-powered communication systems," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4747–4760, Aug. 2020.

[31] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8335–8351, Dec. 2021.

[32] R. Zhu, G. Li, Y. Zhang, Z. Fang, and J. Wang, "Load-balanced virtual network embedding based on deep reinforcement learning for 6G regional satellite networks," *IEEE Trans. Veh. Technol.*, early access, May 24, 2023, doi: 10.1109/TVT.2023.3279625.

[33] X. Wang, Y. Zhang, R. Shen, Y. Xu, and F.-C. Zheng, "DRL-based energy-efficient resource allocation frameworks for uplink NOMA systems," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7279–7294, Aug. 2020.

[34] Y. Peng, Y. Liu, D. Li, and H. Zhang, "Deep reinforcement learning based freshness-aware path planning for UAV-assisted edge computing networks with device mobility," *Remote Sens.*, vol. 14, no. 16, p. 4016, Aug. 2022.

[35] C. Zhang, J. Yang, and N. Wang, "Timely reliability modeling and evaluation of wireless sensor networks with adaptive N-policy sleep scheduling," *Rel. Eng. Syst. Saf.*, vol. 235, Jul. 2023, Art. no. 109270.

[36] D.-C. Huang and J.-H. Lee, "A dynamic N threshold prolong lifetime method for wireless sensor nodes," *Math. Comput. Model.*, vol. 57, nos. 11–12, pp. 2731–2741, Jun. 2013.

[37] S. T. V. Pasca, V. Srividya, and K. Premkumar, "Energy efficient sleep/wake scheduling of stations in wireless networks," in *Proc. Int. Conf. Commun. Signal Process.*, Dept. Elect. Commun. Eng., Adhiparasakthi Eng. College, Melmaruvathur, India, Apr. 2013, pp. 382–386.

[38] F. Iannello, O. Simeone, and U. Spagnolini, "Energy management policies for passive RFID sensors with RF-energy harvesting," in *Proc. IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010, pp. 1–6.

[39] F.-C. Jiang, D.-C. Huang, C.-T. Yang, and F.-Y. Leu, "Lifetime elongation for wireless sensor network using queue-based approaches," *J. Supercomput.*, vol. 59, no. 3, pp. 1312–1335, Mar. 2012.

[40] H. Zheng, K. Xiong, P. Fan, Z. Zhong, and K. B. Letaief, "Age of information-based wireless powered communication networks with selfish charging nodes," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1393–1411, May 2021.

[41] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

[42] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, Apr. 2019.

[43] G. Singh, M. Prakash, and A. Pandey, "An optimal preferred network offload scan framework for smart wearable IoT devices," in *Proc. 15th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bangalore, India, Jan. 2023, pp. 37–41.
[44] H. Xie, Y. Hu, S.-W. Jeon, and H. Jin, "Random activation control for priority AoI," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2023, pp. 443–448.

BEINING WU (Student Member, IEEE) was born in Wuhu, Anhui, China, in 2002. He is currently pursuing the degree in mathematics and applied mathematics with Anhui Normal University.

His current research interests include smart healthcare, the information age, queueing theory, and mobile edge computing.

ZHENGKUN CAI was born in Wuhu, Anhui, in 2001. He is currently pursuing the degree with the School of Information and Software Engineering, University of Electronic Science and Technology.

Currently, he is engaged in research endeavors concerning wireless communication networks and hardware communications. His current research interests include complex system theory and its applications to the optimization of the internet.

WEI WU was born in Wuhu, Anhui, China, in 2002. He is currently pursuing the degree in mechanical engineering with Jilin University.

His current research interests include AI-enhanced next-generation wireless networks, swarm intelligence, and confrontation.

XIAOBIN YIN received the Ph.D. degree in mathematics from Nanjing University, China, in 2004.

He is currently a Professor with the Department of Mathematics, College of Mathematics and Statistics, Anhui Normal University, China. He is also the Director of the Mathematics Society, Anhui. He has authored more than 40 research articles in journals. His current research interests include information freshness optimization, sparse signal processing, and machine learning for wireless networking.