Lyapunov Optimization-Based Latency-Bounded Allocation Using Deep Deterministic Policy Gradient for 11ax Spatial Reuse

AUTHOR(S):

Kotera, Shunnosuke; Yin, Bo; Yamamoto, Koji; Nishio, Takayuki

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 12, 2021, accepted November 26, 2021, date of publication December 6, 2021,
date of current version December 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133311

Lyapunov Optimization-Based Latency-Bounded Allocation Using Deep Deterministic Policy Gradient for 11ax Spatial Reuse

SHUNNOSUKE KOTERA[1], (Student Member, IEEE), BO YIN[1], (Member, IEEE), KOJI YAMAMOTO[1], (Senior Member, IEEE), AND TAKAYUKI NISHIO[1,2], (Senior Member, IEEE)

[1]Graduate School of Informatics, Kyoto University, Sakyo Ward, Kyoto 606-8501, Japan
[2]School of Engineering, Tokyo Institute of Technology, Meguro City, Tokyo 152-8550, Japan

Corresponding author: Koji Yamamoto (kyamamot@i.kyoto-u.ac.jp)

This work was supported by the Ministry of Internal Affairs and Communications/Strategic Information and Communications R&D Promotion Programme (MIC/SCOPE) under Grant JP196000002.

ABSTRACT With the growing demand for wireless local area network (WLAN) applications that require low latency, orthogonal frequency-division multiple access (OFDMA) has been adopted for uplink and downlink transmissions in the IEEE 802.11ax standard to improve the spectrum efficiency and reduce the latency. In IEEE 802.11ax WLANs, OFDMA resource allocation that guarantees latency, called latency-bounded resource allocation, is more challenging than that in cellular networks because severe unmanaged interference from overlapping basic service sets is enhanced due to the concurrent-transmission mechanism newly employed in IEEE 802.11ax. To improve the downlink OFDMA resource allocation with the unmanaged interference caused by IEEE 802.11ax concurrent transmissions, we propose Lyapunov optimization-based latency-bounded allocation with reinforcement learning (RL). We focus on the transmission-queue size for each station (STA) at the access point that determines the STA latency. Using Lyapunov optimization, we formulate the resource-allocation problem with the queue-size constraints in a form that can be solved using RL (i.e., a Markov decision process) and prove the upper bound of the queue size. Our simulation results demonstrated that the proposed method, which uses an RL algorithm with a deep deterministic policy gradient, satisfied the queue-size constraints. This means that the proposed method met the latency requirements, while some baseline methods failed to meet them. Furthermore, the proposed method achieved a higher fairness index than the baseline methods.

INDEX TERMS Basic service set (BSS) color, deep reinforcement learning, IEEE 802.11ax, Lyapunov optimization, orthogonal frequency-division multiple access (OFDMA), quality of service (QoS), resource allocation.

I. INTRODUCTION

With the rapid growth of Internet-connected devices, wireless local area networks (WLANs) have become important because of their reasonable cost and suitable specifications. Accordingly, WLAN applications have become diversified, and a demand exists for latency-bounded communications (e.g., wireless remote control) in WLANs [1].

To improve spectrum utilization and reduce transmission latency, orthogonal frequency-division multiple access (OFDMA) has been introduced in IEEE 802.11ax [2], whereby an access point (AP) can transmit frames to multiple stations (STAs) simultaneously. OFDMA has been used in cellular network systems and enables efficient spectrum utilization while satisfying latency requirements, by allocating OFDMA resources appropriately, based on the STAs' requirements; this is called latency-bounded resource allocation.

OFDMA resource-allocation mechanisms have been studied extensively for cellular networks [3]–[5]. However, resource allocation in WLANs is more challenging than that in cellular networks because WLANs adopt distributed control, and it is difficult for one AP to cooperate with others. Therefore, we must consider a resource-allocation method

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

that works well without cooperating with other APs. Moreover, overlapping basic service sets (OBSSs) cause unmanaged interference in IEEE 802.11ax WLANs, which makes resource allocation more difficult.

In addition to frame collisions caused by carrier sense multiple access/collision avoidance-based channel accesses in conventional WLANs, the concurrent transmissions by STAs associated with OBSS APs in IEEE 802.11ax WLANs—which is introduced to improve the spatial efficiency—increases the potential interference level at the receivers. Such severe unmanaged interference destabilizes the transmission rate and makes latency-bounded OFDMA resource allocation more challenging.

Some studies have addressed OFDMA resource allocation in WLANs. Reference [6] proposed a high-throughput resource-unit assignment scheme. In this scheme, the AP allocates OFDMA resources to maximize the total throughput. However, it was not designed to control the transmission latency.

Latency-aware OFDMA resource allocation in WLANs has been studied previously [7]–[9]. In these studies, traffic is classified into real-time and non-real-time. OFDMA resources are first allocated to STAs with real-time traffic, and the remaining resources are then allocated to STAs with non-real-time traffic. Although this algorithm may reduce the latency for real-time STAs, these studies do not consider concurrent transmissions from OBSSs, which could result in inappropriate allocations when the latency requirements are not satisfied. Therefore, the aim of this study is to provide a latency-bounded OFDMA resource-allocation mechanism for IEEE 802.11ax WLANs that considers OBSS concurrent transmissions.

To solve the resource-allocation problem in IEEE 802.11ax WLANs, we employ reinforcement learning (RL) and Lyapunov optimization. RL is a technique in which an agent learns an optimal strategy in a given environment by trial and error, based on its experience. However, to apply RL to solve the resource-allocation problem [5], it is necessary to formulate the problem as a Markov decision process (MDP).

To formulate the latency-bounded resource allocation as an MDP, we use the Lyapunov optimization scheme presented in our previous study [10]. In the present study, we adopt a deep deterministic policy gradient (DDPG) algorithm [11] to solve the resource allocation problem formulated as an MDP using Lyapunov optimization.

The contributions of this paper are summarized as follows:

- We provide a latency-bounded resource-allocation method for spatial-reuse operations in IEEE 802.11ax WLANs wherein an AP does not cooperate with OBSSs, and unmanaged interference is caused by concurrent transmissions from OBSSs. Most existing resource-allocation schemes assume cellular networks in which APs can cooperate with each other.
- By using Lyapunov optimization, the resource-allocation problem is formulated in a form that can be transformed into an MDP that can be solved by RL, and we prove the upper bound of the transmission latency.
- We demonstrated via simulations that the proposed method satisfies the requirement for latency in environments wherein multiple OBSS APs cause interference, although existing studies have considered a single basic service set (BSS). Furthermore, we confirmed that the proposed method achieves great fairness with OBSSs compared to the baselines.

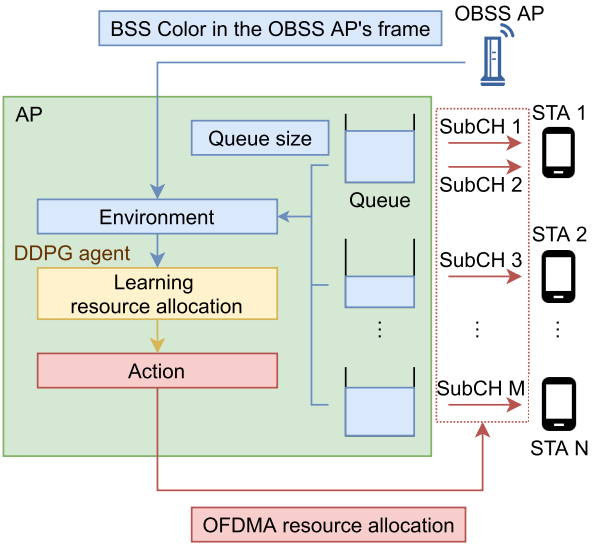

FIGURE 1. Resource-allocation framework of this study. We focus only on downlink transmissions. The considered AP receives frames from the OBSS APs and identifies the sending AP based on the basic service set (BSS) color bits in the received frame. Based on information regarding the OBSS APs and the queue sizes of the STAs, the DDPG agent at the considered AP calculates the OFDMA resource allocation.

The rest of this paper is organized as follows. Section II presents the allocation model and the formulation of the latency-bounded allocation problem. Section III presents the transformation of the problem using Lyapunov optimization. Section IV defines an MDP, and Section V introduces the proposed allocation framework comprising DDPG. Section VI presents the simulation results. Section VII presents the conclusions of this paper.

II. PROBLEM FORMULATION

A. SYSTEM MODEL

Fig. 1 presents the allocation framework of this study. We focus only on downlink transmissions. We assume that there is one AP and N STAs in the considered network, with some OBSS APs around the STAs. Let the index of the STAs be denoted by $n \in \{1, \ldots N\}$, and the subchannels be denoted by $m \in \{1, \ldots, M\}$. The OBSS APs use the same channel as the considered AP; that is, they affect the transmission between the considered AP and the associated STAs.

In IEEE 802.11ax networks, multiple transmitters can transmit simultaneously when the received interference power is below OBSS_PD. OBSS_PD is the sensitivity

threshold for the OBSS frames [2], and transmitters can set it within a fixed range. When a transmitter transmits simultaneously with other transmitters, it must reduce the transmission power as follows [12]:

$$P = P_{\text{ref}} - (P_{\text{OBSS\_PD}} - P_{\text{OBSS\_PD}_{\text{min}}})\,(\text{dBm}), \tag{1}$$

where P_{ref} is the reference power defined in the standard [12], $P_{\text{OBSS\_PD}}$ is the OBSS_PD, and $P_{\text{OBSS\_PD}_{\text{min}}}$ is the lowest OBSS_PD in a predefined range. Therefore, the higher $P_{\text{OBSS\_PD}}$ that the transmitter sets, the more opportunities it can obtain for simultaneous transmissions; however, it must transmit at a lower transmission power. When the channel is idle (i.e., no transmitters are sending frames), the transmitter can transmit frames without reducing the transmission power. When the OBSS APs transmit frames via OFDMA, the abovementioned simultaneous transmission and transmit-power decisions are made on a per-subchannel basis.

Let the transmission power in subchannel m at instant t be denoted by $P_m[t]$ and the noise power by σ^2. Moreover, let the interference power at STA n in subchannel m from OBSS APs be denoted by I_{nm}. We approximate the data rate, based on the Shannon capacity [13]; that is,

$$r_{nm}[t] = W \log_2\left(1 + \frac{P_m[t]/l(d_n)}{I_{nm}[t] + \sigma^2}\right), \tag{2}$$

where W denotes the bandwidth of one subchannel, d_n denotes the distance from the considered AP to STA n, and t denotes the time index. In the above estimation, we use the following distance-based path-loss model [14]:

$$l(d) = 20\log_{10}(f_c) - 28 + 10\,\alpha\log_{10}(\max\{d, 1\})\,(\text{dB}), \tag{3}$$

where d denotes the distance, f_c denotes the center frequency, and α denotes the path-loss factor.

In IEEE 802.11ax networks, BSS color bits are embedded in the frame header [2]. They indicate which BSS the transmitter belongs to. Therefore, in downlink transmissions, we can identify the OBSS AP under transmission using the BSS color bits. We then assume that the data rate $r_{nm}[t]$ is obtained as follows. First, we identify the transmitter from the BSS color bits in the OBSS frame's header. We then estimate the interference power by referring to the previous interference power of the transmitter. Finally, we calculate $r_{nm}[t]$ from the estimated interference power by (2).

We define the total data rate of STA n as follows:

$$R_n[t] = \sum_{m=1}^{M} x_{nm}[t]\, r_{nm}[t], \tag{4}$$

where $x_{nm}[t] \in \{0, 1\}$ denotes whether the considered AP allocates subchannel m to STA n.

We assume that the AP has queues for each STA. Let the arrival rate be denoted by $\rho_n[t]$, and the queue size be denoted by $Q_n[t]$. The queue size $Q_n[t]$ evolves as follows:

$$Q_n[t+1] = \max\{Q_n[t] - R_n[t]\,\tau, 0\} + \rho_n[t]\,\tau \quad \forall t, \tag{5}$$

where τ denotes a time-slot length and $Q_n[0] = 0$.

B. PROBLEM FORMULATION

The objective of this study is to allocate OFDMA resources for latency-bounded transmissions. To achieve latency-bounded transmissions, we introduce an allowable queue size $\overline{Q}_n$ for STA n. Note that the main transmission-delay factor in WLANs is the queuing delay [15]. Therefore, we allocate subchannels, such that the queue size Q_n is under the allowable queue size $\overline{Q}_n$. However, if we consider only the latency, many resources may be concentrated at an STA with strict constraints. Therefore, we adopt proportional fair allocation under the latency constraint. In the allocation, we set the product of the data rates $R_n[t]$ for an objective function. By maximizing the objective function, the proportional fair allocation is realized [16]. We summarize this optimization problem as follows[1]:

$$\begin{aligned}
&\underset{x_{nm}[t]}{\text{maximize}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\prod_{n=1}^{N} R_n[t] \\
&\text{subject to} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} Q_n[t] \le \overline{Q}_n \quad \forall n \\
&\qquad x_{nm}[t] \in \{0, 1\} \quad \forall n, m \\
&\qquad \sum_{n=1}^{N} x_{nm}[t] \le 1 \quad \forall m.
\end{aligned} \tag{6}$$

The aim of the objective function is to realize fair allocation. By allocating the subchannels to maximize the objective function, we can reduce the deviation of the data rates and allocate the subchannels more fairly. The first constraint presents the latency constraints. If we maintain the average queue size under the allowable value, each STA's latency requirement is guaranteed. The second and third constraints indicate the resource-allocation constraints. The second constraint shows that each subchannel can be allocated exclusively during a single time slot. The third constraint shows that each subchannel can be allocated to at most one STA.

III. LYAPUNOV OPTIMIZATION

In optimization problem (6), the queue size $Q_n[t]$ is a function of the arrival rate, as in (5). Accordingly, if we obtain the expectation of the arrival rate, we can solve optimization problem (6) directly. However, it is impossible to obtain the future arrival rate and expectation of the arrival rate. Therefore, we introduce Lyapunov optimization [17]. Lyapunov optimization is an online algorithm and does not need future information. Thus, it can be applied to optimization problem (6).

We define a virtual queue $Z_n[t]$ that changes over time, as follows:

$$Z_n[t+1] := \max\{Z_n[t] + Q_n[t+1] - \overline{Q}_n, 0\}, \tag{7}$$

[1]When $R_n[t] = 0$, we assign it an alternate constant value $R_n[t] = C$. This is because, if even one STA is not allocated, $R_n[t]$ is equal to 0, and there is no difference, irrespective of the allocation of the other STAs.

where $Z_n[0] = 0$. This virtual queue size indicates the total backlog of the gap between the actual queue size $Q_n[t+1]$ and the desirable value $\overline{Q}_n$.

Let a virtual queue vector be denoted as $\boldsymbol{Z}[t] := (Z_1[t], Z_2[t], \ldots, Z_N[t])$; we introduce the Lyapunov function as follows:

$$L(\boldsymbol{Z}[t]) := \frac{1}{2}\sum_{n=1}^{N}(Z_n[t])^2. \tag{8}$$

This function indicates the size of the virtual queue vector. Using this function, we introduce the Lyapunov drift $\Delta(\boldsymbol{Z}[t])$ as follows:

$$\Delta(\boldsymbol{Z}[t]) := \mathbb{E}\big[L(\boldsymbol{Z}[t+1]) - L(\boldsymbol{Z}[t]) \,|\, \boldsymbol{Z}[t]\big]. \tag{9}$$

This drift is the expectation of the Lyapunov function change. If we minimize $\Delta(\boldsymbol{Z}[t])$, we can minimize the virtual queue sizes $Z_n[t]$ and satisfy the first constraint in (6). However, we cannot allocate resources fairly by only minimizing $\Delta(\boldsymbol{Z}[t])$. To make a fair allocation under the latency constraint, we introduce the drift-plus-penalty [17] as follows:

$$\Delta(\boldsymbol{Z}[t]) - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right], \tag{10}$$

where $V \geq 0$ denotes an importance weight. The second term is the weighted expectation of part of objective function (6). The importance weight represents the extent to which we emphasize fair allocation against latency constraints. To help readers better understand the role of the weight V, we present the following proposition.

Proposition 1: Let us define function y as follows:

$$y[t] = \prod_{n=1}^{N} R_n[t]. \tag{11}$$

The total data rate R is upper bounded; therefore, we can assume that the expected value of y is upper bounded by a finite value $y_{\max}$; that is,

$$\mathbb{E}\big[y[t]\big] \leq y_{\max}. \tag{12}$$

We suppose that there are constants $B \geq 0, \epsilon \geq 0$, and a target value y^*, such that

$$\Delta(\boldsymbol{Z}[t]) \leq B - \epsilon\sum_{n=1}^{N}\big|Z_n[t]\big| \quad \forall t, \tag{13}$$

$$\Delta(\boldsymbol{Z}[t]) - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right] \leq B - Vy^* - \epsilon\sum_{n=1}^{N}\big|Z_n[t]\big|. \tag{14}$$

Then, the expected average y and virtual queue Z satisfy the following:

$$\liminf_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\prod_{n=1}^{N} R_n[t]\right] \geq y^* - \frac{B}{V}, \tag{15}$$

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\mathbb{E}\Big[\big|Z_n[t]\big|\Big] \leq \frac{B + V(y_{\max} - y^*)}{\epsilon}. \tag{16}$$

Proof: Provided in Appendix A. □

We can understand this proposition as follows. If we set a larger weight V, we can increase the value of y, which allows the resources to be allocated more fairly. However, an increase in V results in an increase in Z because the right-hand side of (16) increases, and the upper limit of Z is relaxed. The increase in Z indicates that the gap between the current queue size and the desirable queue size increases. Accordingly, if we increase V, the latency constraints are relaxed. In summary, the tradeoff between fair resource allocation and latency constraints can be controlled by the weight V.

Finally, we rearrange the terms in (10) and separate the uncontrollable variables from the controllable variables in the following lemma.

Lemma 2: The drift-plus-penalty (10) is upper bounded as follows:

$$\Delta(\boldsymbol{Z}[t]) - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right] \leq B - \mathbb{E}\left[\sum_{n=1}^{N} Z_n[t]\,R_n[t]\,\tau + V\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right], \tag{17}$$

where B is invariable, irrespective of the resource allocation.

Proof: Provided in Appendix B. □

In Lyapunov optimization, we minimize this upper bound instead of the drift-plus-penalty. Therefore, we set this upper bound as a new objective function. We maximize the expectation term of the right-hand side in (17) because B is a constant, and the expectation term is negative. By extracting the terms from the expectation terms on the right-hand side in (17), we can transform the original problem, (6), as follows:

$$\begin{aligned} &\underset{x_{nm}[t]}{\text{maximize}} \sum_{n=1}^{N} Z_n[t]\,R_n[t]\,\tau + V\prod_{n=1}^{N} R_n[t] \\ &\text{subject to } x_{nm}[t] \in \{0,1\} \quad \forall n, m \\ &\qquad \sum_{n=1}^{N} x_{nm}[t] \leq 1 \quad \forall m. \end{aligned} \tag{18}$$

The considered AP can calculate virtual queue sizes $Z_n[t]$ and set a coefficient V in advance. The total data rate $R_n[t]$ is a function of the allocation index $x_{nm}[t]$, as presented in (4), and the AP can set $x_{nm}[t]$. Therefore, the considered AP must determine only $x_{nm}[t]$ to maximize the objective function (18).

IV. MARKOV DECISION PROCESS FORMULATION

We formulated the allocation problem and transformed it using Lyapunov optimization. Unfortunately, the unmanaged interference from the spatial-reuse operation complicates the

allocation problem. Moreover, optimization problem (18) is 0-1 integer programming. 0-1 integer programming is proven to be NP-complete [18] and is difficult to solve directly. Therefore, we introduce the DDPG algorithm, which approximates the mapping from a state to an optimal allocation decision. Once this mapping is learned, an estimate of the solution to optimization problem (18) can be obtained in a short computation time. To apply the algorithm, we formulate resource allocation as a stochastic decision process; specifically, the MDP.

For spatial reuse, we consider one AP as a learning agent. Moreover, we consider all OBSS APs as part of the environment. We define a stochastic decision process as a quadruplet (Ω, A, q, R). In this expression, Ω denotes the state set in the environment; that is, a state space. Moreover, $A(\boldsymbol{\omega}[t])$ denotes the possible action set in one state $\boldsymbol{\omega}[t] \in \Omega$; that is, an action space. $q[t]$ denotes the probability distribution of the state transition from $\boldsymbol{\omega}[t]$ to $\boldsymbol{\omega}[t+1]$, when the selected action is $\boldsymbol{a}[t] \in A(\boldsymbol{\omega}[t])$. When this state transition occurs, the agent receives a reward $R(\boldsymbol{\omega}[t], \boldsymbol{\omega}[t+1], \boldsymbol{a}[t])$. The agent learns how to select an action to maximize the expectation of this reward. In the considered process, the next state $\boldsymbol{\omega}[t+1]$ depends on the present state $\boldsymbol{\omega}[t]$ and the selected action $\boldsymbol{a}[t]$. Therefore, the process satisfies the Markov property and is an MDP.

A. STATE

Let us denote the state space Ω as the Cartesian product of the queue-size state space Ω_{QUEUE} and the channel state space Ω_{CH}; that is,

$$\Omega := \Omega_{\mathrm{QUEUE}} \times \Omega_{\mathrm{CH}}. \tag{19}$$

The queue-size state $\boldsymbol{\omega}_{\mathrm{QUEUE}} \in \Omega_{\mathrm{QUEUE}}$ denotes the vector of the current queue sizes of the STAs; that is,

$$\boldsymbol{\omega}_{\mathrm{QUEUE}}[t] = (Q_1[t], Q_2[t], \ldots, Q_N[t]). \tag{20}$$

We assume that the queue size $Q_n[t]$ does not increase beyond the upper limit $Q_{\max}$. Note that the queue size is continuous, and the state space Ω_{QUEUE} is also continuous.

The channel state $\boldsymbol{\omega}_{\mathrm{CH}}[t] \in \Omega_{\mathrm{CH}}$ denotes the OBSS AP index that is identified from the received frame. We assume that the considered AP senses the channel continuously. If the AP detects a transmission, it checks the BSS color bits in the received frame header and immediately identifies the transmitter. BSS color is a field embedded in the IEEE 802.11ax header and indicates the BSS of the transmitter [2].

The channel state space is denoted by a tuple of the interferer index of each subchannel; that is,

$$\boldsymbol{\omega}_{\mathrm{CH}}[t] = (i_1[t], i_2[t], \ldots, i_M[t]), \tag{21}$$

where $i_m[t]$ presents the index of the OBSS AP transmitting in subchannel m. We denote $i_m[t] = 0$ when no other transmitter is transmitting in subchannel m or the transmitter is not identified, owing to a preamble error.

B. ACTION

Let $a_m \in A_m := \{1, 2, \ldots, N\}$ denote the STA to which the agent allocates subchannel m. The action space $\mathcal{A}$ is then defined using the Cartesian product of the allocation action space A_m; that is,

$$\mathcal{A} := \prod_{m=1}^{M} A_m. \tag{22}$$

For all subchannels, the agent determines the STA to which the subchannel is allocated; thus, there are N^M ways to allocate subchannels. Therefore, the action space may become large if the number of STAs or subchannels becomes large.

C. REWARD

We use objective function (18) as the reward; that is, the reward is defined as follows:

$$r = \sum_{n=1}^{N} Z_n[t]\, R_n[t]\, \tau + V \prod_{n=1}^{N} R_n[t], \tag{23}$$

where $Z_n[t]$ denotes the virtual queue size, V denotes the importance weight, and $R_n[t]$ denotes the total data rate defined in (4). Therefore, optimization problem (18) is solved when the agent learns the optimal strategy with which the reward is maximized.

V. RESOURCE ALLOCATION WITH DDPG

In this study, we seek to develop a policy that enables the fairest allocation under the latency constraints. In Section IV, we formulated the allocation problem as an MDP. Therefore, we can apply the RL algorithm using the formulated decision process. In the OFDMA resource allocation in WLANs, the number of possible allocations may become very large as the numbers of STAs and subchannels increase. In addition, as mentioned in Section IV-A, the state space is continuous. To address the problem with a continuous state space, we adopt a deep RL algorithm. Deep Q-network (DQN) [19] is one of the well-known deep RL algorithms. However, as pointed out in [11], while DQN solves problems with high-dimensional observation spaces, it can only handle discrete and low-dimensional action spaces. Instead of DQN, we use an actor-critic deep RL algorithm, DDPG [11]. DDPG is designed for problems with high-dimensional and continuous action spaces (e.g., robot operation). Therefore, it is suitable for optimization problem (18).

A. ACTOR AND CRITIC

DDPG is based on an actor-critic method, similar to the deterministic policy gradient (DPG) algorithm [20]. An actor-critic method has a parameterized actor function $\mu(\boldsymbol{\omega}|\boldsymbol{\theta}^{\mu})$ and critic function $V(\boldsymbol{\omega}, \boldsymbol{a}|\boldsymbol{\theta}^{V})$. In DDPG, the two functions are represented by neural networks. $\boldsymbol{\theta}^{\mu}$ is the weight of the actor network, and $\boldsymbol{\theta}^{V}$ is the weight of the critic network. The actor function $\mu(\boldsymbol{\omega}|\boldsymbol{\theta}^{\mu})$ is a deterministic policy in which the agent selects an action $\boldsymbol{a}$ in state $\boldsymbol{\omega}$. The critic function $V(\boldsymbol{\omega}, \boldsymbol{a}|\boldsymbol{\theta}^{V})$ is the value function in state $\boldsymbol{\omega}$ and action $\boldsymbol{a}$. The structures

of the actor and critic functions in this study are described in Section VI-A. Given a state $\boldsymbol{\omega}[t]$, action $\boldsymbol{a}[t]$, and actor function μ, we define a critic function as follows [11]:

$$V^{\mu}(\boldsymbol{\omega}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^{V}) = \mathbb{E}\left[\sum_{i=t}^{T} \gamma^{(i-t)} r(\boldsymbol{\omega}[i], \boldsymbol{a}[i])\right], \quad (24)$$

where $\gamma \in [0, 1)$ denotes the discounted factor. This function presents the expected sum of discounted rewards for one episode. The agent learns the optimal policy to achieve the highest action value. In the considered system model, the critic function $V^{\mu}(\boldsymbol{\omega}, \boldsymbol{a}|\boldsymbol{\theta}^{V})$ is the discounted sum of the weighted data-rate product and virtual-queue term. Therefore, the agent learns to allocate resources more fairly, while meeting latency constraints.

The discounted factor γ indicates how much we emphasize future rewards. If we use a large γ value, we obtain a better outcome after convergence. However, this lengthens the learning phase.

By using the Bellman equation [21], we can transform (24) as follows:

$$V^{\mu}(\boldsymbol{\omega}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^{V}) = \mathbb{E}\big[r(\boldsymbol{\omega}[t], \boldsymbol{a}[t]) + \gamma \mathbb{E}[V^{\mu}(\boldsymbol{\omega}[t+1], \boldsymbol{a}[t+1]|\boldsymbol{\theta}^{V})]\big]. \quad (25)$$

As the DDPG policy is deterministic, we can eliminate the inner expectation as follows:

$$V^{\mu}(\boldsymbol{\omega}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^{V}) = \mathbb{E}\big[r(\boldsymbol{\omega}[t], \boldsymbol{a}[t]) + \gamma V^{\mu}(\boldsymbol{\omega}[t+1], \mu(\boldsymbol{\omega}[t+1]|\boldsymbol{\theta}^{\mu})|\boldsymbol{\theta}^{V})\big]. \quad (26)$$

We optimize a critic-function parameter by minimizing the loss function. The loss function is defined as follows:

$$L(\boldsymbol{\theta}^{V}) = \mathbb{E}\left[(V(\boldsymbol{\omega}[t], \boldsymbol{a}[t]\Big|\boldsymbol{\theta}^{V}) - h[t])^2\right], \quad (27)$$

where $\boldsymbol{\theta}^{V}$ is a parameter of the approximate critic function, and $h[t]$ is a function defined as follows:

$$h[t] = r(\boldsymbol{\omega}[t], \boldsymbol{a}[t]) + \gamma V^{\mu}(\boldsymbol{\omega}[t+1], \mu(\boldsymbol{\omega}[t+1]|\boldsymbol{\theta}^{\mu})|\boldsymbol{\theta}^{V}). \quad (28)$$

Moreover, we update an actor-function parameter to maximize the expected return from the start distribution J defined as follows:

$$J = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{(t-1)} r(\boldsymbol{\omega}[t], \boldsymbol{a}[t])\right]. \quad (29)$$

The actor function $\mu(\boldsymbol{\omega}|\boldsymbol{\theta}^{\mu})$ is updated via the chain rule to the expected return as follows:

$$\nabla_{\boldsymbol{\theta}^{\mu}} J \sim \mathbb{E}\big[\nabla_{\boldsymbol{\theta}^{\mu}} V(\boldsymbol{\omega}, \boldsymbol{a}|\boldsymbol{\theta}^{V})|_{\boldsymbol{\omega}=\boldsymbol{\omega}[t], \boldsymbol{a}=\mu(\boldsymbol{\omega}[t]|\boldsymbol{\theta}^{\mu})}\big] = \mathbb{E}\big[\nabla_{\boldsymbol{a}} V(\boldsymbol{\omega}, \boldsymbol{a}|\boldsymbol{\theta}^{V})|_{\boldsymbol{\omega}=\boldsymbol{\omega}[t], \boldsymbol{a}=\mu(\boldsymbol{\omega}[t]|\boldsymbol{\theta}^{\mu})} \cdot \nabla_{\boldsymbol{\theta}^{\mu}} \mu(\boldsymbol{\omega}|\boldsymbol{\theta}^{\mu})|_{\boldsymbol{\omega}=\boldsymbol{\omega}[t]}\big], \quad (30)$$

where $\boldsymbol{\theta}^{\mu}$ is a parameter of the approximate actor function. Gradient (30) has been proven to be the policy gradient [20]; thus, we use gradient (30) to update $\boldsymbol{\theta}^{\mu}$.

B. DDPG FEATURES

DDPG adopts several features to overcome the disadvantages of neural networks [11]. When using neural networks for RL, the samples must be independent and identically distributed. However, the samples that we can obtain do not meet these conditions. To address this problem, DDPG adopts a replay buffer. A replay buffer stores the tuple $(\boldsymbol{\omega}[t], \boldsymbol{a}[t], r[t], \boldsymbol{\omega}[t+1])$. We select some samples uniformly from the replay buffer for a minibatch and update the actor and critic networks. By using the minibatch, the selected samples are independent and identically distributed.

Another disadvantage of neural networks is the convergence problem. Specifically, in DDPG, the network $V(\boldsymbol{\omega}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^{V})$ is updated, such that the loss-function (27) is minimized. However, this update is not stable because the target value $h[t]$ also uses the network $V(\boldsymbol{\omega}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^{V})$. In other words, the learning is slow and does not necessarily converge. To stabilize the learning, DDPG adopts target networks [22]. Target networks consist of a copy of the actor and critic networks, and are used to calculate the target value $h[t]$. Let $\boldsymbol{\theta}$ denote the weights of the original networks, and $\boldsymbol{\theta}'$ denote the weights of the target network. We update the weights of these networks as follows:

$$\boldsymbol{\theta}' \leftarrow \eta\boldsymbol{\theta} + (1-\eta)\boldsymbol{\theta}', \quad (31)$$

where η is a constant with $\eta \ll 1$. When using target networks, the target value $h[t]$ changes more slowly, and the learning can be stabilized.

In the learning phase, DDPG adopts exploration policy μ', defined as follows:

$$\mu'(\boldsymbol{\omega}[t]) := \mu(\boldsymbol{\omega}[t]|\boldsymbol{\theta}^{\mu}[t]) + \mathcal{N}, \quad (32)$$

where μ denotes the current policy and $\mathcal{N}$ denotes a sample of the noise process. The noise process is determined to be suitable for the environment. In this study, we selected a white Gaussian noise process.

C. DDPG APPLICATION

In this section, we explain how we apply DDPG to this study. In this study, the input of the actor network is queue sizes for N STAs and the channel states of M subchannels. Moreover, the output of the actor network is the allocation of M subchannels. Thus, the actor network can be denoted by the mapping $\mu : \mathbb{R}^{N} \times \mathbb{N}^{M} \rightarrow \mathbb{N}^{M}$.

To enhance the learning efficiency, we normalize the input vector of the actor network. The input vector is calculated as follows:

$$\boldsymbol{\omega}[t] = (\boldsymbol{\omega}_{\mathrm{QUEUE}}[t]/Q_{\max}, \boldsymbol{\omega}_{\mathrm{CH}}[t]/M), \quad (33)$$

where $\boldsymbol{\omega}_{\mathrm{QUEUE}}$ and $\boldsymbol{\omega}_{\mathrm{CH}}$ respectively denote the queue and channel states defined in Section IV. $Q_{\max}$ denotes the upper limit of the queue size and M denotes the number of subchannels.

In this study, the output of the actor network is represented by a normalized vector. Therefore, we must transform the

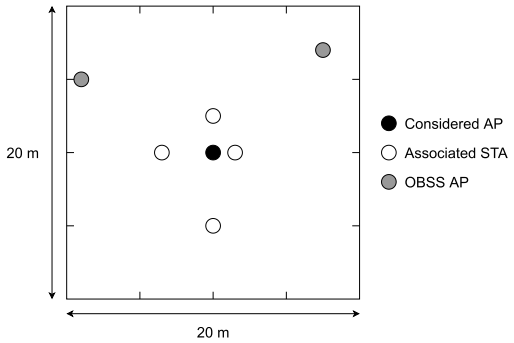

FIGURE 2. Evaluation topology. In this topology, the interference power is less than OBSS_PD. Thus, the considered AP and OBSS APs can transmit concurrently. All transmitters and receivers are fixed during the simulation.

normalized output vector into an action vector. Let the normalized output of the actor network at instant t be denoted by $\boldsymbol{o}[t]$ and the mth element of $\boldsymbol{o}[t]$ be denoted by $o_m[t]$. Moreover, let the action vector at instant t be denoted by $\boldsymbol{a}[t]$, and the mth element of $\boldsymbol{a}[t]$ be denoted by $a_m[t]$. $a_m[t]$ indicates the STA to which the agent allocates subchannel m. We then transform $o_m[t]$ into $a_m[t]$, as follows:

$$a_m[t] = \begin{cases} 1, & o_m[t] \leq 0, \\ \lceil No_m[t] \rceil, & 0 < o_m[t] \leq 1, \\ N, & 1 < o_m[t], \end{cases} \tag{34}$$

where m denotes the subchannel index, N denotes the number of STAs, and $\lceil \cdot \rceil$ denotes the ceiling function.

VI. SIMULATION EVALUATION

In this section, we validate the performance of the proposed scheme through a numerical evaluation.

A. SIMULATION SETTINGS

We consider a downlink transmission from one AP to four STAs. Two OBSS APs are near the considered network and interfere with the transmissions in it. To evaluate the performance in a spatial-reuse environment, we select the topology in which the interference power is less than OBSS_PD, and the considered AP and OBSS APs can transmit concurrently. The topology is shown in Fig. 2.

We assume that the OBSS APs use the same channel as the considered AP, and that they can detect each other. We also assume that one of the OBSS APs can start a transmission, owing to a carrier sense. We set the traffic rate according to a uniform distribution. When we determine the subchannel allocation, we assume that the data rate can be calculated from the estimated interference power based on the OBSS frame's header. The simulation parameters are presented in Table 1.

Fig. 3 presents the structures of the DDPG networks. In this figure, the "Dense" layer is the fully connected dense layer, the "ReLU" layer is one of the activation functions [23], and the "Linear" layer is a linear function. We used the "Sigmoid" layer for the output layer of the actor network to restrict the output range. The parameters of DDPG are summarized in Table 2.

We compare the queue sizes to evaluate the transmission latency. To evaluate the allocation fairness, we also use Jain's fairness index [25] to compare the data rates. Jain's fairness

TABLE 1. Simulation parameters.

Parameter	Value
Number of controllable APs	1
Number of associated STAs N	4
Number of OBSS APs	2
OBSS transmission probability	0.33
Center frequency f_c	5.18 GHz
Number of subcarriers per subchannel	52 tones [12]
Number of subchannels M	4
Upper limit of transmission power	20 dBm
Noise power	-174 dBm/Hz
OBSS_PD	-62 dBm [12]
Path-loss factor α	3 [14]
Allocation interval τ	1 ms
Simulation length	200 ms
Traffic rate	$\rho \sim \mathcal{U}(0, 4)$ Mbit/s
Allowable queue size $\overline{Q}_i$	25 kbit
Queue-size upper limit $Q_{\max}$	100 kbit
Substitute value for zero data rate C	0.001

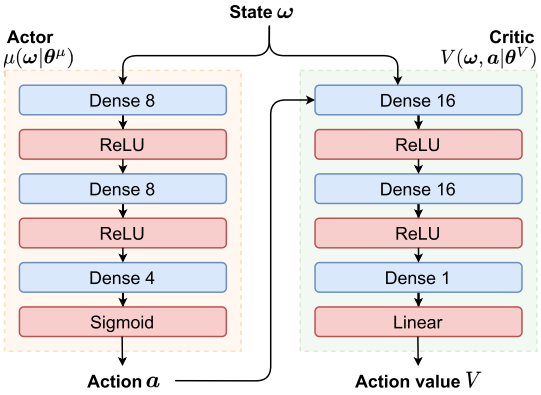

FIGURE 3. Structure of DDPG neural networks. In the actor network, the input is the state ω, and the output is the action a. In the critic network, the inputs are the state ω and action a, and the output is the action value V.

TABLE 2. DDPG agent parameters.

Parameter	Value
Number of steps in each episode	200 steps
Number of learning episodes	2000 episodes
Update period of target network	1 step
Update coefficient η	0.001
Discount rate γ	0.9
Batch size	32
Optimizer	Adam [24]
Learning rate of optimizer	0.001
Noise process $\mathcal{N}$	White Gaussian noise process
Mean of Gaussian noise	0
Standard deviation of Gaussian noise	0.33
Replay buffer size	100000

index is defined as follows:

$$f(\boldsymbol{R}) = \frac{\left(\sum_{n=1}^{N} R_n[t]\right)^2}{N \sum_{n=1}^{N} (R_n[t])^2}. \tag{35}$$

This index value is larger for more uniform data rates for all the STAs.

We evaluated the following resource-allocation schemes in the WLAN with a spatial-reuse operation.

1) PROPOSED SCHEME

The resource allocation is determined by the DDPG agent. In this scheme, the allocation index $x_{nm}[t]$ is given as follows:

$$x_{nm}[t] = \begin{cases} 1, & \text{if } n = \mu(\boldsymbol{\omega}[t], m|\boldsymbol{\theta}^{\mu}), \\ 0, & \text{otherwise}, \end{cases} \tag{36}$$

where μ is the actor function in DDPG and m is the index of the subchannel.

2) RANDOM SCHEME

The resource allocation is performed randomly.

3) RATE SCHEME

The objective of the rate scheme is to maximize the total throughput, which is the same as that in [6]. In the rate scheme, each subchannel is allocated to the STA with the highest data rate in the subchannel; that is,

$$x_{nm}[t] = \begin{cases} 1, & \text{if } n = \arg\max_{n} r_{nm}[t], \\ 0, & \text{otherwise}. \end{cases} \tag{37}$$

4) QUEUE-SIZE PRIORITY SCHEME (QUEUE SCHEME)

The objective of the queue scheme is to prioritize latency-sensitive STAs, which is the same as that in [7]. In the queue scheme, the entire channel is allocated to the STA with the largest queue size among the four STAs; that is,

$$x_{nm}[t] = \begin{cases} 1, & \text{if } n = \arg\max_{n} Q_n[t], \\ 0, & \text{otherwise}. \end{cases} \tag{38}$$

Note that although the spatial-reuse operation was not used in [6] and [7], our simulations use the WLAN spatial-reuse operation in all the resource-allocation schemes. In addition to comparing the performance of the resource-allocation schemes, we evaluated the performance of the proposed scheme without the spatial-reuse operation to emphasize its importance in WLANs. This scheme is referred to as "w/o SR scheme" in the simulation results.

B. SIMULATION RESULTS

Fig. 4 presents the average queue sizes. The queue sizes of the five schemes are smaller than the allowable queue size. In these schemes, the queue size of the proposed scheme is the smallest. In the w/o SR scheme, the average queue size is the largest. This is because the considered AP does not transmit concurrently with the OBSS APs, and the spectrum utilization is degraded. In the queue scheme, the AP considers only the queue sizes, not the data rates. Accordingly, the spectrum utilization is limited, and the queue size increases. In the proposed scheme, the considered AP considers both queue sizes and data rates. Thus, it can improve the transmission efficiency while satisfying the latency constraints.

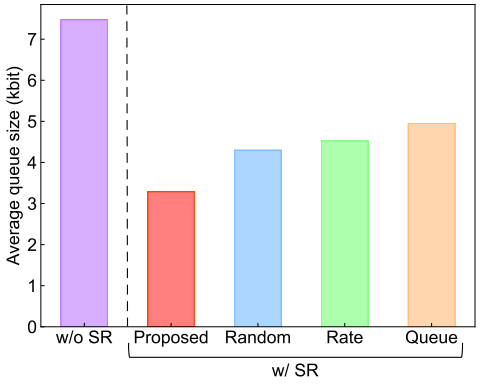

FIGURE 4. Average queue size for each scheme. The proposed scheme achieves a smaller queue size than the other schemes.

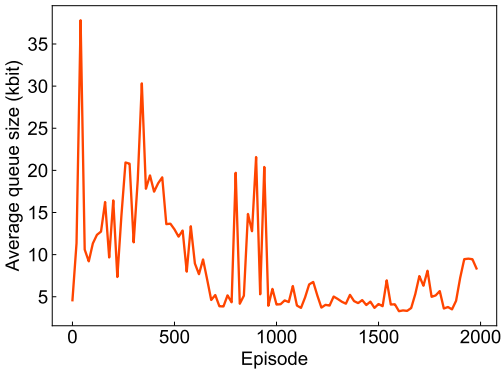

FIGURE 5. Transition of average queue size in the proposed scheme. The average queue size is the smallest at 1600 episodes.

Fig. 5 presents the transition of the average queue size of the proposed scheme during DDPG learning. Depending on the episode, the fluctuation of the average queue size becomes stable; that is, the learning is stable. The average queue size is the smallest at 1600 episodes. Therefore, we use the result at 1600 episodes in the following comparisons.

Fig. 6 presents the achievement rate for the allowable queue size. The proposed scheme realizes an achievement rate of 1; that is, the considered AP always maintains queue sizes under the allowable queue size. The queue scheme also realizes an achievement rate of 1. In this scheme, subchannels are allocated preferentially to the STA whose queue size is the largest. Thus, the queue size is kept under the allowable queue size. The achievement rate in the w/o SR scheme is the lowest among the five schemes due to the same reason as that in the average queue size.

Fig. 7 presents the standard deviations of the queue sizes. If the deviation is low, the queue size is stable, and the

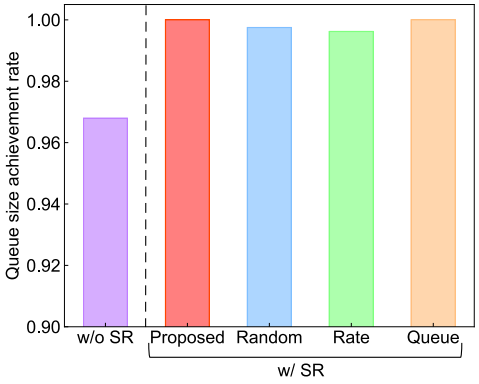

FIGURE 6. Achievement rates of queue sizes. The proposed and queue schemes achieved an achievement rate of 1.

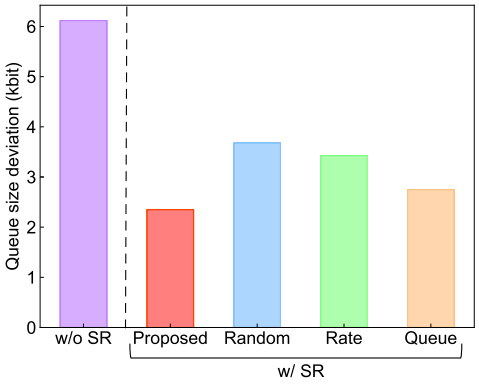

FIGURE 7. Standard deviations of the queue sizes for each scheme. The deviation obtained using the proposed scheme is lower than that of the compared schemes.

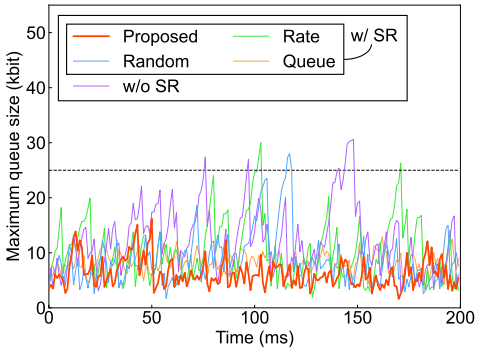

FIGURE 8. Maximum queue size in each scheme. The horizontal dashed line indicates the allowable queue size. The queue size in the proposed scheme does not exceed the allowable queue size, whereas the queue sizes in the other schemes sometimes exceed the allowable queue size.

transmission latency is also stable. The proposed scheme realizes less deviation than the compared schemes. In the proposed scheme, subchannels are allocated to keep the queue sizes small. Accordingly, the change in the queue size is small, which results in a lower deviation.

Fig. 8 presents the change in the maximum queue size for each scheme. The queue size of the proposed scheme does not exceed the allowable queue size. However, the queue sizes

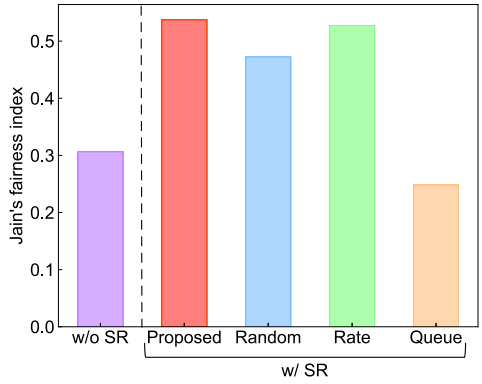

FIGURE 9. Jain's fairness index of data rates. The index in the proposed scheme is the highest among the five schemes.

of the other schemes sometimes do not meet the allowable values. In the proposed scheme, the considered AP allocates to all the STAs more frequently to control the queue sizes. According to this allocation, the queue sizes are retained under the allowable value. The same is true of the queue scheme.

Fig. 9 presents Jain's fairness index of the data rates. The fairness index in the proposed scheme is higher than that in the other schemes. Objective function (18) contains the product of the data rate, such that fair allocation is realized, and this term improves the fairness index. The fairness index in the queue scheme is the lowest among the five schemes because too many subchannels are allocated to the STA whose queue size is the largest, which degrades the fairness index.

VII. CONCLUSION

We proposed a novel latency-bounded allocation framework for spatial-reuse WLANs. Latency-bounded OFDMA resource allocation in WLANs is more challenging than that in cellular networks due to the unmanaged interference from the OBSSs. To perform latency-bounded OFDMA resource allocation under unmanaged interference, we used an RL algorithm. As one of the inputs of the algorithm, we adopted BSS color bits, which indicate the transmitter that transmits the frame. To realize latency-bounded transmissions, we focused on the queue size because the majority of the latency in WLANs is due to the queuing delay. We then formulated the OFDMA allocation problem with queue-size constraints and transformed the problem into a form such that RL could be applied via Lyapunov optimization. We proved that the queue size is upper bounded in the transformed problem. In the evaluation, we compared the queue size and fairness of the allocation of the proposed scheme with those of other schemes. The simulation results confirmed that the proposed method achieved a high Jain's fairness index while satisfying the latency constraints. Moreover, the proposed method reduced the average queue size and its deviation, which implies that the proposed method can increase the capability of WLANs to accommodate more STAs with a satisfactory latency.

APPENDIX

A. UPPER LIMIT OF DATA RATE PRODUCTS AND LOWER LIMIT OF VIRTUAL QUEUES

Proof of Proposition 1: We consider a slot t. We then obtain the expectations of both sides of (14) as follows:

$$\mathbb{E}\big[L(\boldsymbol{Z}[t+1]) - L(\boldsymbol{Z}[t])\big] - V\,\mathbb{E}\big[y[t]\big|\boldsymbol{Z}[t]\big] \le B - Vy^* - \epsilon \sum_{n=1}^{N} \mathbb{E}\Big[\big|Z_n[t]\big|\Big]. \quad (39)$$

By taking the sum over $t \in \{0, 1, \ldots, T-1\}$ for some $t > 0$ and rearranging the terms, the above inequality is transformed as follows:

$$\mathbb{E}\big[L(\boldsymbol{Z}[T]) - L(\boldsymbol{Z}[0])\big] - V \sum_{t=0}^{T-1} \mathbb{E}\big[y[t]\big|\boldsymbol{Z}[t]\big] \le (B - Vy^*)T - \epsilon \sum_{t=0}^{T-1}\sum_{n=1}^{N} \mathbb{E}\Big[\big|Z_n[t]\big|\Big]. \quad (40)$$

By dividing (40) by VT, rearranging the terms, and neglecting non-negative terms appropriately, we can obtain the following inequality:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[y[t]\big|\boldsymbol{Z}[t]\big] \ge y^* - \frac{B}{V} - \frac{\mathbb{E}\big[L(\boldsymbol{Z}[0])\big]}{VT}. \quad (41)$$

By dividing (40) by ϵT, applying (12), and rearranging the terms in the same manner as that used above, we can obtain the following inequality:

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N} \mathbb{E}\Big[\big|Z_n[t]\big|\Big] \le \frac{B + V(y_{\max} - y^*)}{\epsilon} + \frac{\mathbb{E}\{L(\boldsymbol{Z}[0])\}}{\epsilon T}. \quad (42)$$

By taking the limits of (41) and (42) as $T \to \infty$, we can prove Proposition 1. □

B. UPPER LIMIT OF OBJECTIVE FUNCTION

Proof of Lemma 2: From (7), (8), and (9), $\Delta(\boldsymbol{Z}[t])$ satisfies the following inequality, as in [17]:

$$\begin{aligned}\Delta(\boldsymbol{Z}[t]) &= \mathbb{E}[\,L(\boldsymbol{Z}[t+1]) - L(\boldsymbol{Z}[t])\,|\,\boldsymbol{Z}[t]\,] \\ &= \mathbb{E}\left[\frac{1}{2}\sum_{n=1}^{N}(Z_n[t+1])^2 - \frac{1}{2}\sum_{n=1}^{N}(Z_n[t])^2 \,\middle|\, \boldsymbol{Z}[t]\right] \\ &\le B' + \sum_{n=1}^{N} Z_n[t]\,\mathbb{E}\big[\,Q_n[t+1] - \overline{Q}_n \,|\, \boldsymbol{Z}[t]\,\big], \quad (43)\end{aligned}$$

where B' denotes a finite constant; that is,

$$\sum_{n=1}^{N}\mathbb{E}\left[\frac{1}{2}\Big[(Q_n[t+1])^2 + \overline{Q}_n^2\Big] - Q_n[t+1]\,\overline{Q}_n \,\middle|\, \boldsymbol{Z}[t]\right] = B'. \quad (44)$$

The constant B' is independent of the allocation. By applying (43) to the drift-plus-penalty (10), we obtain

$$\begin{aligned}&\Delta(\boldsymbol{Z}[t]) - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right] \\ &\le B' + \sum_{n=1}^{N} Z_n[t]\,\mathbb{E}\big[\,Q_n[t+1] - \overline{Q}_n \,|\, \boldsymbol{Z}[t]\,\big] \\ &\quad - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right]. \quad (45)\end{aligned}$$

On applying (5) to (45), we obtain

$$\begin{aligned}(45) = \sum_{n=1}^{N} Z_n[t]\,\mathbb{E}\Big[\,&\max\big\{Q_n[t] - R_n[t]\,\tau, 0\big\} + a_n[t]\tau \\ &- \overline{Q}_n \,\Big|\, \boldsymbol{Z}[t]\Big] + B' - V\,\mathbb{E}\left[\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right]. \quad (46)\end{aligned}$$

As we cannot control $Q_n[t]$ or $a_n[t]\tau$, and $\overline{Q}_n$ is invariable irrespective of the allocation, we can transform (46) as follows:

$$(46) = B - \mathbb{E}\left[\sum_{n=1}^{N} Z_n[t]\,R_n[t]\tau + V\prod_{n=1}^{N} R_n[t] \,\middle|\, \boldsymbol{Z}[t]\right], \quad (47)$$

where B is a constant that cannot be changed by resource allocation. From (45), (46), and (47), we obtain Lemma 2. □

REFERENCES

[1] Cisco Systems, San Jose, CA, USA. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*. Accessed: Feb. 12, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf

[2] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 197–216, 1st Quart., 2019.

[3] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 678–700, 2nd Quart., 2013.

[4] A. Alwarafy, M. Abdallah, B. S. Ciftler, A. Al-Fuqaha, and M. Hamdi, "Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey," May 2021, *arXiv:2106.00574*.

[5] A. T. Z. Kasgari and W. Saad, "Model-free ultra reliable low latency communication (URLLC): A deep reinforcement learning framework," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[6] M. Wu, J. Wang, Y.-H. Zhu, and J. Hong, "High throughput resource unit assignment scheme for OFDMA-based WLAN," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–8.

[7] H. Zhou, B. Li, Z. Yan, M. Yang, and Q. Qu, "An OFDMA based multiple access protocol with QoS guarantee for next generation WLAN," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Ningbo, China, Sep. 2015, pp. 1–6.

[8] T. Mishima, S. Miyamoto, S. Sampei, and W. Jiang, "Novel DCF-based multi-user MAC protocol and dynamic resource allocation for OFDMA WLAN systems," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, San Diego, CA, USA, Jan. 2013, pp. 616–620.

[9] E. Avdotin, D. Bankov, E. Khorov, and A. Lyakhov, "Resource allocation strategies for real-time applications in Wi-Fi 7," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Odessa, Ukraine, May 2020, pp. 1–6.
[10] S. Kotera, B. Yin, K. Yamamoto, T. Nishio, M. Morikura, and H. Abeysekera, "Latency-aware fair scheduling for spatial reuse in WLANs: A Lyapunov optimization approach," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–6.
[11] 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," Jul. 2019, *arXiv:1509.02971*.
[12] *IEEE P802.11ax TM/D4.0 Amendment 6: Enhancements for High Efficiency WLAN*, 802.11 Working Group of the 802 Committee, Piscataway, NJ, USA, Feb. 2019.
[13] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 4, pp. 623–656, Oct. 1948.
[14] *Propagation Data and Prediction Methods for the Planning of Indoor Radiocommunication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 450 GHz*, document ITU-R P.1238-10, International Telecommunications Union, Geneva, Switzerland, Jul. 2015.
[15] J. S. Vardakas, I. Papapanagiotou, M. D. Logothetis, and S. A. Kotsopoulos, "On the end-to-end delay analysis of the IEEE 802.11 distributed coordination function," in *Proc. 2nd Int. Conf. Internet Monit. Protection (ICIMP)*, San Jose, CA, USA, Jul. 2007, pp. 96–100.
[16] L. Massoulié and J. Roberts, "Bandwidth sharing: Objectives and algorithms," in *Proc. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 3, New York, NY, USA, Mar. 1999, pp. 1395–1403.
[17] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.
[18] R. M. Karp, *Reducibility Among Combinatorial Problems*. Boston, MA, USA: Springer, 1972.
[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[20] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn. (ICML)*, E. P. Xing and T. Jebara, Eds., Beijing, China, Jun. 2014, pp. 387–395.
[21] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.
[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," Dec. 2013, *arXiv:1312.5602*.
[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 807–814.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*.
[25] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination*. Maynard, MA, USA: Digital Equipment Corporation, 1984.

SHUNNOSUKE KOTERA (Student Member, IEEE) received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2020, where he is currently pursuing the master's degree with the Graduate School of Informatics.

BO YIN (Member, IEEE) received the B.E. degree in electrical and electronic engineering and the master's and Ph.D. degrees in informatics from Kyoto University, in 2016, 2018, and 2021, respectively. He received the VTS Japan Young Researcher's Encouragement Award, in 2017.

KOJI YAMAMOTO (Senior Member, IEEE) received the B.E. degree in electrical and electronic engineering and the master's and Ph.D. degrees in informatics from Kyoto University, in 2002, 2004, and 2005, respectively. From 2004 to 2005, he was a Research Fellow of the Japan Society for the Promotion of Science (JSPS). Since 2005, he has been with the Graduate School of Informatics, Kyoto University, where he is currently an Associate Professor. From 2008 to 2009, he was a Visiting Researcher at the Wireless@KTH, KTH Royal Institute of Technology (KTH), Sweden. His research interests include radio resource management, game theory, and machine learning. He is a Senior Member of the IEICE and a member of the Operations Research Society of Japan. He was a Tutorial Lecturer in ICC 2019. He received the PIMRC 2004 Best Student Paper Award, in 2004, and the Ericsson Young Scientist Award, in 2006. He also received the Young Researcher's Award, the Paper Award, the SUEMATSU-Yasuharu Award, and the Educational Service Award from the IEICE of Japan, in 2008, 2011, 2016, and 2020, respectively, and the IEEE Kansai Section GOLD Award, in 2012. He serves as the Symposium Co-Chair for GLOBECOM 2021 and the Vice Co-Chair for IEEE ComSoc APB CCC. He also serves as an Editor for IEEE Wireless Communications Letters, IEEE Open Journal of Vehicular Technology, and *Journal of Communications and Information Networks*.

TAKAYUKI NISHIO (Senior Member, IEEE) received the B.E. degree in electrical and electronic engineering and the master's and Ph.D. degrees in informatics from Kyoto University, in 2010, 2012, and 2013, respectively. He had been an Assistant Professor with the Graduate School of Informatics, Kyoto University, from 2013 to 2020. From 2016 to 2017, he was a Visiting Researcher with the Wireless Information Network Laboratory (WINLAB), Rutgers, The State University of New Jersey, USA. He has been an Associate Professor with the School of Engineering, Tokyo Institute of Technology, Japan, since 2020. His current research interests include machine learning-based networks control, machine learning in wireless networks, and heterogeneous resource management.