IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 9, 2022, accepted May 25, 2022, date of publication June 1, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3179581

ACK-Less Rate Adaptation Using Distributional Reinforcement Learning for Reliable IEEE 802.11bc Broadcast WLANs

TAKAMOCHI KANDA[1], (Graduate Student Member, IEEE), YUSUKE KODA[2], (Member, IEEE), YUTO KIHIRA[1], (Student Member, IEEE), KOJI YAMAMOTO[1], (Senior Member, IEEE), AND TAKAYUKI NISHIO[1,3], (Senior Member, IEEE)

[1]Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan
[2]Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland
[3]School of Engineering, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152-8550, Japan

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

This work was supported in part by the Ministry of Internal Affairs and Communications/Strategic Information and Communications Research and Development Promotion Programme (MIC/SCOPE) under Grant JP196000002.

ABSTRACT As a step towards establishing reliable broadcast wireless local area networks (WLANs), this paper proposes acknowledgement (ACK)-less rate adaptation to alleviate reception failures at broadcast recipient stations (STAs) using distributional reinforcement learning (RL). The key point of this study is that the algorithms for learning the strategy of ACK-less rate adaptation are evaluated in terms of the broadcast performance, which is composed of the data rate of the broadcast access point (AP) and the reception success rate at the recipient STAs. ACK-less rate adaptation framework was realized using the received signal strength (RSS) of the uplink frames transmitted by the non-broadcast STAs to the non-broadcast APs, which correlated with the broadcast performance with a confounding effect from the deployment of the broadcast recipient STAs. However, this rate adaptation framework has the problem that it incurs the reception failures at a part of the broadcast recipient STAs, because deep Q-learning used in the previous framework cannot deal with the wide distribution of the broadcast performance. To address this challenge, this paper further discusses the rate adaptation using distributional RL, which approximates the entire distribution of the broadcast performance. The simulations confirmed the following: 1) Using the expected broadcast performance learned by deep Q-learning improved the performance in terms of the Pareto efficiency. 2) Learning the entire distribution of the broadcast performance enabled the broadcast AP to determine the tail of the distribution using risk measure, and to alleviate reception failures while implementing the rate adaptation in the same way as the method that learns only expected broadcast performance.

INDEX TERMS Broadcast, conditional value at risk, deep reinforcement learning, distributional reinforcement learning, IEEE 802.11bc, rate adaptation.

I. INTRODUCTION

Broadcasting on wireless local area networks (WLANs) is currently being examined for distributing information across specific locations to a large number of people. As the standard for providing enhanced broadcast services (eBCS) on WLANs, IEEE 802.11bc is currently being considered [1]. A variety of use cases are considered in IEEE 802.11bc, such as multi-lingual and emergency broadcasting, eSports virtual reality video distribution, and lecture room slide distributions [2].

In the eBCS systems, the adaptive data rate control is more important issue than that in the existing broadcasting systems, e.g., TV broadcasting, and a broadcast method in the conventional WLANs. This is for the following reasons. First, WLANs, being different from other wireless broadcast systems, e.g., TV broadcasting, are generally self-deployed. This mandates the eBCS APs to be deployed at various locations without as many installation designs as the current wireless broadcast systems. Hence, for ease of deployment, as in the

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

current WLANs, autonomous adaptations of the parameters of APs to various installation locations are required. Second, the eBCS needs to satisfy the requirements of various applications, some of which require higher rates, while the current WLANs mainly use broadcast for control frames with low transmission rates. This mandates the rate control method that adapts to various environment and broadcast applications.

However, rate adaptation in the eBCS systems is more challenging than the conventional WLAN systems. This is because, in the eBCS systems, acknowledgement (ACK) mechanisms are not implemented between an eBCS AP and the recipient stations (STAs) [3] while the current unicast systems leverage them to adapt parameters to the channel conditions. This could be due to the ACK implosion problem [4], which is caused by a traffic overload while receiving many ACK frames simultaneously. Thus, being different from current WLANs, the eBCS AP is not notified whether the STAs have received the eBCS data frames successfully or not. This forces the eBCS AP to control the data rate without using current ACK-based heuristic rate adaptation algorithms such as ARF [5], SampleRate [6], and Minstrel [7]. Therefore, it is quite challenging for the eBCS AP to control the data rate adaptively to channel conditions between the eBCS AP and recipient STAs.

In our previous work [8], we addressed this challenge by developing an ACK-less rate adaptation framework, harnessing the uplink frames transmitted by the STAs that are associated with non-eBCS APs and surrounded by the eBCS recipient STAs. We referred to such STAs and eBCS recipient STAs as non-eBCS and eBCS STAs, respectively. As it is easily envisaged in eBCS applications, we assumed that the non-eBCS STAs located near the eBCS STAs transmitted uplink frames to the non-eBCS APs, as in the current applications of WLANs. The assumption on the locations of non-eBCS and eBCS STAs will easily hold, as the eBCS reception functionality is offered in addition to the current non-eBCS functionality. Therefore, almost all STAs can act as both eBCS and non-eBCS, and can occasionally switch their functionality. In this case, eBCS STAs can be placed near non-eBCS STAs as long as the people holding the STAs are clustered. The lecture room slide distribution scenario satisfies the aforementioned assumptions [2]. Students with eBCS or non-eBCS STAs may take nearby seats; some of them may upload their lecture notes by sending non-eBCS uplink frames to non-eBCS APs. Moreover, it is assumed that the number of non-eBCS STAs is much smaller than that of eBCS STAs. This assumption is not a requirement; rather, it is a strict condition for the evaluation of ACK-less rate-adaptation framework. In the rate adaptation framework, by overhearing such uplink frames, an eBCS AP surveyed the channel conditions between the eBCS AP and each eBCS STA. To validate the feasibility of utilizing the information obtained from the overheard uplink frames, we considered, as an example, that the eBCS AP measured received signal strength (RSS) of the overheard uplink frames that the non-eBCS STAs transmitted to the non-eBCS APs. Thereby, the eBCS AP implemented the rate adaptation for the eBCS STAs. For learning the rate adaptation strategy, we adopted a neural-network (NN)-based reinforcement learning. Specifically, by using deep reinforcement learning (DRL), the eBCS AP was able to learn a mapping from the RSS of the uplink frames to the expected value of the indicator of the performance of broadcasting. It was based on the idea that there would be a correlation between the RSS of the uplink frames and the broadcast performance via the confounding effect from the deployments of the eBCS STAs. For example, if the eBCS STAs were far away from an eBCS AP, both the uplink RSS and broadcast performance would be lower simultaneously. In this case, the eBCS AP was aware that a lower data rate was required to be set in order to increase the eBCS STAs with successful receptions. The broadcast performance indicator was designed as a reward, which is an objective function of reinforcement learning, such that the data rate of the eBCS AP and the reception success rate at the eBCS STAs increased simultaneously. Hence, by estimating the reward in correlation with the uplink RSS and by maximizing it, the eBCS AP was able to set a data rate, such that the number of eBCS STAs with successful reception and the data rate jointly increased for various eBCS STA deployments. However, the reception failures occur because the reward distribution becomes wider, which leads to that the eBCS AP cannot adapt the data rate to a part of the eBCS recipient STAs. Since the previous work adopted DRL as one example, the learning algorithm for obtaining the rate adaptation strategy is not evaluated. The maximizing problem of the broadcast performance indicator was not addressed as a general optimization problem, as the environmental observation was insufficient. In other words, the RSS of the overheard uplink frames does not represent the broadcast performance of all eBCS STAs. Therefore, in the previous study, RL was used as an example for acquiring this strategy.

In this paper, we studied the learning algorithms for the ACK-less rate adaptation to use high data rate while avoiding reception failures. The objective of this work is to identify which algorithm is appropriate by evaluating the impact of the difference in the targets to learn on the performance of the ACK-less rate adaptation. As the algorithm to learn the expected value of the broadcast performance indicator, we evaluated deep Q-network (DQN) [9], [10].

Furthermore, as the algorithm to learn the entire distribution of the broadcast performance instead of the expected value, we evaluated distributional reinforcement learning (RL) [11]. Specifically, we adopted quantile regression DQN (QR-DQN) [12] to approximate the reward distribution, given the RSS of the overheard uplink frames. Learning the entire distribution instead of the expected value provided eBCS AP the ability to avoid reception failures which appears in the tail of the performance distribution, and thus, the eBCS system becomes reliable. To emphasize the tail of the reward distribution, we used a risk measure [13] by referring to an established approach that uses conditional value at risk (CVaR), which is one of the risk measures associated with

QR-DQN [14]. In this way, we integrated the learning of reward distribution into the ACK-less rate adaptation framework and alleviated reception failures at the eBCS STAs by emphasizing the tail of the performance distribution. An overview of learning the statistics/distribution of the reward is shown in Fig. 1.

The contributions of this paper are as follows:

- In order to establish a reliable ACK-less rate adaptation in broadcast WLANs, we studied the benefit of applying the learned statistics of broadcast performance for data rate,[1] in correlation with the RSS of non-broadcast uplink frames transmitted by the STAs associated with non-broadcast APs. This was formulated on the idea that the RSS of the uplink frames possibly correlates with the performance measure of broadcast WLANs. Hence, the learned statistics and obtained uplink RSS would allow a broadcast AP to be aware of the rate which would lead to successful receptions. We confirmed that the rate adaptation with learning the expected performance via a standard DQN improved the success and data rates as measured in terms of the Pareto efficiency, compared with rate adaptation without learning the statistics.
- We further discussed a problem specific challenge where the distribution of the broadcast performance indicator became wider when the number of STAs that transmitted the uplink frames was much smaller than that of the broadcast recipient STAs. More specifically, we studied the effectiveness of leveraging the learned distribution of the broadcast performance itself, and not the expectation used in standard Q-learning for ACK-less rate adaptations. Simulation results showed that determining the data rate in view of the tail of the distribution enhanced the success rate while implementing the rate adaptation in the same way as method learning only expected broadcast performance.

In a nutshell, our key contribution is the demonstration of the feasibility and benefit of learning the statistics/distributions of the broadcast performance under the correlation with the RSS of the uplink frames transmitted by non-eBCS STAs to non-eBCS APs. In order to achieve these insights, throughout this paper, we have assumed the availability of the broadcast performance during learning. However, in real deployments, this performance may not be available due to the ACK-less nature of the broadcast WLANs. In order to address this challenge, our future work will be aimed at discussing sim-to-real learning as a realistic scenario such that the broadcast performance would be available during learning. However, to the best of our knowledge, we believe that the present study is sufficient for shedding light on the ACK-less rate adaptation using uplink RSS, which itself is a novel finding. Hence, an in-depth discussion of this problem is beyond the scope of this study.

[1] This broadcast performance is defined in (5) in view of the broadcast recipient STAs with the successful reception and data rate.

The rest of this paper is organized as follows: Section II describes the considered eBCS system model. Section III explains the proposed method. Section IV presents evaluation results and discussions. Section V concludes the paper.

II. SYSTEM MODEL

We considered an eBCS system, where one eBCS AP, I non-eBCS APs, N eBCS STAs, and m ($m \ll N$) non-eBCS STAs were deployed in a two-dimensional area, $\mathcal{W} \subset \mathbb{R}^2$. The proposed method in Section III can be easily extended to scenarios with multiple eBCS APs. The eBCS AP periodically broadcasted eBCS frames to the eBCS STAs. The eBCS STAs were located around the non-eBCS APs forming clusters, and the non-eBCS STAs were located near the eBCS STAs. The STA clusters were indexed by $i \in \{1, 2, \ldots, I\}$, and were formed by N_i eBCS STAs, where the total number of eBCS STAs is denoted by $N = \sum_{i=1}^{I} N_i$. The non-eBCS STAs transmitted uplink frames, e.g., ACK frames in unicast communications, to the non-eBCS APs with which they were associated. In contrast, the eBCS STAs did not transmit uplink frames, as considered in IEEE 802.11bc. Here, we assumed that these APs and STAs were densely deployed and focused at the clusters far from the eBCS AP, where part of the eBCS STAs would have failed to receive the eBCS frames. An example of a system model with one non-eBCS AP in the environment is shown in Fig. 1. We considered that the eBCS AP could receive the uplink frames transmitted by the non-eBCS STAs to survey the channel conditions between the eBCS AP and the eBCS STA clusters to learn the eBCS performance, given the RSS of the uplink frames measured at the eBCS AP.

III. LEARNING BROADCAST PERFORMANCE USING DEEP REINFORCEMENT LEARNING ALGORITHMS

This section provides an overview of the proposed method that learns the expected value or the distribution of the indicator of eBCS performance, given the RSS of the uplink frames transmitted by the non-eBCS STAs. As discussed in Section I, this is based on the idea that the deployment of eBCS STAs has a confounding effect on both the RSS of uplink frames and the eBCS performance: the two being correlated with each other. Hence, learning the statistics/distributions of broadcast performance conditioned by the RSS is useful to assess the ongoing broadcast service correctly and determine a better-performing data rate. We first propose the approximation method based on DQN, which is the most basic algorithm of DRL. Second, we propose the approximation method based on QR-DQN, which is one of the distributional RL algorithms.

A. LEARNING BROADCAST PERFORMANCE USING DEEP Q-NETWORK

In this section, we discuss the method that learns the expected value of the eBCS performance indicator, given RSS of the uplink frames of the non-eBCS STAs and data rate of the eBCS AP. To approximate the mapping from the RSS of

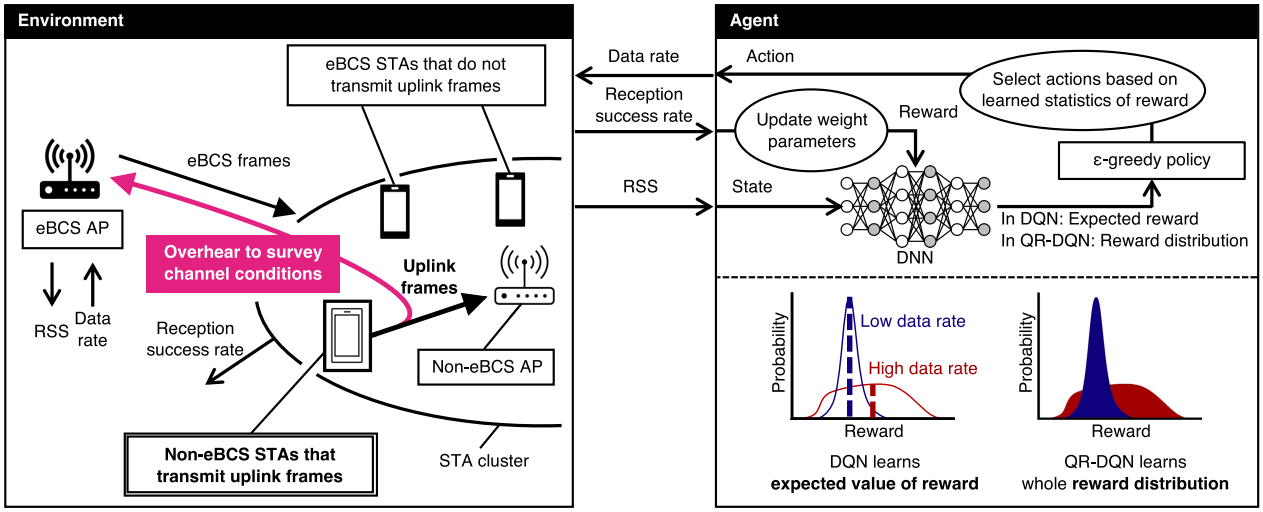


FIGURE 1. An overview of the proposed methods to learn the statistics/distribution of the broadcast performance indicator that is formulated as the reward. The agent in the eBCS AP learns the statistics/distribution exploiting the RSS of the overheard uplink frames, and selects the data rate based on the learned statistics/distribution. DQN approximates the expected value of the reward, and QR-DQN approximates the reward distribution.

the uplink frames and the data rate of the eBCS AP to the expected value of the performance indicator, we used DQN. The indicator was designed as the reward that reflected on the data rate of the eBCS AP and the reception success rate of the eBCS STAs, such that, via the maximization of the reward, both performances were enhanced. The data rate controller of the eBCS AP was defined as an agent, and the rate adaptation strategy was obtained as an optimal state-action value function and a policy. The optimal state-action value function was modeled by a deep neural network (DNN) with the input of state, s, and the output of the learned optimal state-action value function, $\hat{Q}(s, a)$, for each action a. The interactions between the agent and environment were assumed to be described by the Markov decision process (MDP). In this process, the agent observes a state, s_t and selects an action, a_t, based on the policy, π with an iteration step t. In response, the environment gives a reward, r_{t+1} to the agent. Using this reward, the agent updates the approximated optimal state-action value function, $\hat{Q}(s, a)$ based on the Q-learning algorithm [15].

1) STATE, ACTION, REWARD FORMULATION

While learning the broadcast performance, the eBCS AP overhears the uplink frames transmitted by m non-eBCS STAs associated with the non-eBCS APs. The m non-eBCS STAs are assumed to be selected uniformly at random, from a large number of non-eBCS STAs. The eBCS AP observes the RSS of the uplink frames, where the observed RSS is represented as an m-tuple given by

$$p_t := \left(p_t^{(1)}, p_t^{(2)}, \ldots, p_t^{(m)}\right), \tag{1}$$

where $p_t^{(k)}$ denotes the RSS of the kth uplink frame in dBm. Moreover, for distinguishing each cluster from the others, the eBCS AP observes the basic service set identifier (BSSID) of the uplink frames, which is a suitable assumption for current WLAN systems [16]. The observed BSSID is also given as an m-tuple:

$$q_t := \left(q_t^{(1)}, q_t^{(2)}, \ldots, q_t^{(m)}\right), \tag{2}$$

where $q_t^{(k)} \in \{1, 2, \ldots, I\}$ denotes the BSSID of the kth uplink frame. Note that p_t and q_t are sorted by BSSID.

Given the aforementioned observation, p_t and q_t, the state s_t is defined as a $2m$-tuple:

$$s_t = (p_t, q_t). \tag{3}$$

By observing the state s_t, the agent measures the channel conditions between the eBCS AP and eBCS STAs. The input size of the DNN is fixed, and if m is smaller than this size, the missing part is filled in with copied states from another part.

This action is defined as rate selection in which the eBCS AP learns the rate adaptation strategy. Following WLAN standardizations, the available data rates are discretized by $A_1, A_2, \ldots, A_K$; namely, the action space $\mathcal{A}$ is defined as follows:

$$\mathcal{A} := \{A_1, A_2, \ldots, A_K\}. \tag{4}$$

The agent selects an action, a_t from $\mathcal{A}$, for each step t, based on the observation s_t, using the policy.

To alleviate reception failures at the eBCS STAs and transmit eBCS frames at a high data rate, we designed the reward r_{t+1} as

$$r_{t+1} := \begin{cases} -\dfrac{a_t}{A_{\max}}\left(1 - \dfrac{n_t}{N}\right), & n_t < N; \\ \dfrac{a_t}{A_{\max}}, & n_t = N, \end{cases} \tag{5}$$

where n_t is the number of eBCS STAs that succeed in receiving the eBCS frames transmitted by the eBCS AP in step t,

and $A_{\max}$ is the available maximum data rate. We separated the reward by $n_t = N$, such that the reward was always less than zero when $n_t < N$, and greater than zero when $n_t = N$. Through this design, the agent would be sensitive to reception failures due to its negative reward and would, therefore, be able to avoid them. The reward was kept within the range $[-1, 1]$, which stabilized the learning of the DNN.

2) RATE SELECTION FOR DEEP Q-NETWORK DURING TRAINING

We used the ε-greedy policy that selects actions as follows:

$$a_t := \begin{cases} a_{\text{random}} & \text{w.p. } \varepsilon; \\ a_{\text{greedy}} & \text{otherwise,} \end{cases} \tag{6}$$

where a_{random} and a_{greedy} are the random action and greedy action, respectively. The random action, a_{random} is randomly selected from the available data rates, $\mathcal{A}$. The greedy action is selected to maximize the approximated optimal state-action value in each step:

$$a_{\text{greedy}} := \arg\max_{a \in \mathcal{A}} Q(s_t, a). \tag{7}$$

The greedy action, a_{greedy} is optimal in terms of maximizing the reward when the approximated optimal state-action value has already converged. However, while the agent is in learning, selecting greedy actions each time is not optimal, and hinders the convergence of the approximated optimal state-action value. Hence, with probability ε, the agent selects a random action to explore possible experiences.

B. LEARNING BROADCAST PERFORMANCE USING DISTRIBUTIONAL REINFORCEMENT LEARNING

In this section, we discuss the method that learns the distribution of the reward, given RSS of the uplink frames of the non-eBCS STAs. To approximate the mapping from the RSS of the uplink frames and the data rate of the eBCS AP to the distribution of the reward, we used QR-DQN, which is one of the distributional RL algorithms. In QR-DQN, we used the same state, action, and reward as described above. Since it is different from the method based on DQN, the rate adaptation strategy was obtained as an approximation of the reward distribution termed as value distribution and a policy. Similar to the optimal state-action value function in DQN, the value distribution was modeled using a DNN with the input of state, s, and the output of the learned value distribution, $\hat{Z}(s, a)$, for each action a.

1) DISTRIBUTIONAL REINFORCEMENT LEARNING

In this section, we introduce distributional RL and one of its algorithms, QR-DQN. Distributional RL solves the following distributional Bellman equation, which is a distributional version of the Bellman equation in value iteration algorithms, such as Q-learning:

$$Z(s, a) \overset{D}{=} R(s, a) + \gamma Z(s', a'), \tag{8}$$

where $U \overset{D}{=} V$ indicates that the random variables U and V are equal. The random variable, $Z(s, a)$ represents the return in state s with action a, which is approximated by distributional RL. The return is defined as the cumulative sum of discounted rewards:

$$z_t := \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{9}$$

where γ is the discount factor that decides the weight of future rewards, and r_t is the reward in the learning iteration t. The approximated return distribution is termed as value distribution. The expectation value of the distribution is equal to the optimal state-action value in the DQN.

QR-DQN is based on the algorithm called quantile regression temporal difference learning (QRTD) [12]. QRTD approximates the return distribution by following N_{q} parameters:

$$\theta_i := F_Z^{-1}(\hat{\tau}_i), \quad i = 1, 2, \ldots, N_{\text{q}}, \tag{10}$$

where F_Z^{-1} is the inverse cumulative distribution function of the value distribution Z. The notation, $\hat{\tau}_i$ represents midpoints of the quantiles, $\tau_i = i/N_{\text{q}}$, defined as follows:

$$\hat{\tau}_i := \frac{\tau_{i-1} + \tau_i}{2}, \tag{11}$$

where $\tau_0 = 0$. The parameter θ_i is updated by the following equation:

$$\begin{aligned} \theta_i(s_t, a_t) \leftarrow\ & \theta_i(s_t, a_t) \\ & + \eta(\hat{\tau}_i - \mathbb{1}(r_{t+1} + \gamma z_{t+1} < \theta_i(s_t, a_t))), \end{aligned} \tag{12}$$

where $r_t \sim R(s_t, a_t)$, $z_{t+1} \sim Z(s_t, a_t)$, η is the learning rate, and $\mathbb{1}$ is the indicator function.

Based on the QRTD algorithm, QR-DQN approximates the return distribution with the DNN. The input of the DNN is the state, which is the same as that of DQN. However, the output of the DNN is the value distribution, θ_i, for each iteration, while that of the DQN is the approximated optimal state-action value function. The parameter, θ_i is calculated for all available actions, a_t. QR-DQN uses an extended version of Huber loss [17], termed as quantile Huber loss, as the loss function of DNN. Quantile Huber loss is defined as

$$\rho_\tau^\kappa(u) := |\tau - \mathbb{1}(u < 0)| \frac{\mathcal{L}_\kappa(u)}{\kappa}, \tag{13}$$

where $\mathcal{L}_\kappa(u)$ is the Huber loss often used for DQN, defined as:

$$\mathcal{L}_\kappa(u) := \begin{cases} \dfrac{1}{2}u^2, & |u| \le \kappa; \\ \kappa\left(|u| - \dfrac{1}{2}\kappa\right), & |u| > \kappa, \end{cases} \tag{14}$$

where u is the temporal difference error and κ is the parameter of Huber loss.

TABLE 1. Environment settings.

Region $\mathcal{W}$	$300\,\mathrm{m} \times 300\,\mathrm{m}$
Carrier frequency f_c	$5\,\mathrm{GHz}$
Bandwidth W	$20\,\mathrm{MHz}$
Transmit power of the eBCS AP P_{eBCS}	$10\,\mathrm{mW}$
Transmit power of the non-eBCS STAs P_{STA}	$10\,\mathrm{mW}$
Number of the non-eBCS APs I	2
Number of the eBCS STAs	$N_1 = N_2 = 100$
Path loss model	Indoor model in [18]
Break point distance of the path loss	$10\,\mathrm{m}$
Noise power spectrum density	$-174\,\mathrm{dBm/Hz}$

2) RATE SELECTION POLICY FOR QUANTILE REGRESSION DEEP Q-NETWORK DURING TRAINING

We used a variant of the ε-greedy policy for QR-DQN while learning the value distribution:

$$a_t := \begin{cases} a_{\mathrm{random}} & \text{w.p. } \varepsilon; \\ a_{\mathrm{greedy}} & \text{otherwise}, \end{cases} \tag{15}$$

$$a_{\mathrm{greedy}} := \arg\max_{a \in \mathcal{A}} \frac{1}{N_{\mathrm{q}}} \sum_{j=1}^{N_{\mathrm{q}}} \theta_j(s_t, a). \tag{16}$$

This policy selects actions based on the expected value of the return distribution. Since this expected value is equal to the optimal state-action value, if DQN and QR-DQN learn the optimal value accurately, then this variant would be equivalent to the ε-greedy policy used in DQN.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed methods that learn the statistics/distribution of the reward, given the RSS of the uplink frames of the non-eBCS STAs via simulations in the considered eBCS system. First, we evaluated the learned statistics/distribution of the reward using Monte Carlo simulations. For ease of evaluation, we sampled three levels of the RSS of the uplink frames. Second, for demonstration, we simulated the rate adaptation of the eBCS AP using the learned statistics/distribution.

A. SIMULATION SETTINGS

1) ENVIRONMENT SETTINGS

The simulation environment was established as shown in Table 1. We simulated various deployments to enable the eBCS AP to learn the statistics of the reward for them. To set this environment, the non-eBCS APs were deployed according to the binomial point process in the two-dimensional area $\mathcal{W}$. Therein, the eBCS and non-eBCS STAs were deployed randomly inside circles centered at the position of the ith non eBCS AP with radius σ_i and distance B_i, as shown in Fig. 2. While analyzing the statistics of the reward, the distance $B := \max_i B_i$ and the radius $\sigma := \sigma_{\arg\max_i B_i}$, are randomly selected for each deployment.

Due to carrier sensing of the eBCS AP, non-eBCS APs, and non-eBCS STAs, it was assumed that no frame collisions occurred and there was no interference among them.

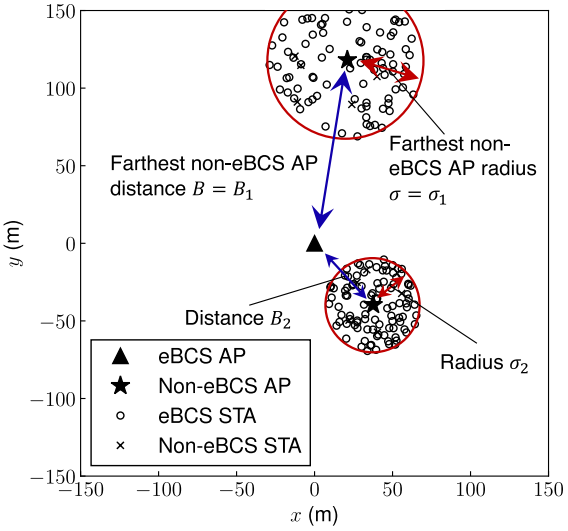


FIGURE 2. An example of deployment of the eBCS AP, non-eBCS AP, eBCS STAs, and non-eBCS STAs in the simulation.

Therefore, reception failures were only caused due to the SNR of the eBCS frame being lower than that required for the eBCS STAs to successfully decode the frame. The required SNR depends on the selected data rates of the eBCS AP. Given a data rate, a, the required SNR Γ_{req} is defined as follows:

$$\Gamma_{\mathrm{req}}(a) := 2^{a/W} - 1, \tag{17}$$

where W is the channel bandwidth. This requirement was derived from Shannon's noisy-channel coding theorem, stating that the upper limit of the data rate for error-free communication is the channel capacity.

For ease of evaluation, we used several data rates defined in IEEE 802.11ax [16]. The action space $\mathcal{A}$ is expressed as

$$\mathcal{A} = \{8.6, 51.6, 103.2, 143.4\}\,\mathrm{Mbit/s}. \tag{18}$$

2) AGENT SETTINGS

Table 2 shows the agent parameter settings of DQN and QR-DQN. Deployments of eBCS AP, non-eBCS AP, eBCS STAs, and non-eBCS STAs were randomly generated for each episode. By selecting m non-eBCS STAs uniformly at random, and subsequently calculating the RSS, information about the state of one episode is obtained. This state is transitioned randomly and independent of the previous state. Hence, the discount factor is set as $\gamma = 0$, consistent with the previous work [8]. For each state, we consider the problem of determining the data rate. This implies that the value distribution would approximate the reward distribution at the same time as that of the return.

For comparison, the same DNN structures were used for DQN and QR-DQN, except for the output shape since the output of the DNN in DQN was different from that in QR-DQN, as mentioned previously. The DNN was composed of six fully connected layers, where each hidden layer consisted of

TABLE 2. Agent settings.

Parameters	DQN	QR-DQN
Number of episodes	10000	10000
Number of steps	100	100
Policy	ε-greedy ($\varepsilon = 0.3$)	ε-greedy ($\varepsilon = 0.3$)
Learning rate η	0.0001	0.0001
Discount factor γ	0.0	0.0
Batch size	32	32
Loss function	Huber loss	Quantile Huber loss
Optimizer	Adam [19]	Adam
Replay buffer [20] capacity	10000	10000
Number of quantiles N_{q}	-	50
Huber loss threshold κ	1	1
Selection of m non-eBCS STAs	Uniformly random	Uniformly random

64 units and used rectified linear unit (ReLU) activation. This setup was an example, i.e., any DNN having more weight parameters than the ones in this setup can be applied to our approximation method with the appropriate hyperparameter tuning.

B. EVALUATED RATE ADAPTATION FRAMEWORK

In this section, we introduce the framework for rate adaptation, which was used for demonstration of the eBCS AP using the learned statistics/distribution of the reward. In this framework, the DQN agent selected the data rate based only on the greedy policy, while the QR-DQN agent selected the data rate based on a policy associated with a risk measure. Moreover, for comparison with the above two methods, we illustrated another rate adaptation method that did not use the learned statistics of the reward as a baseline.

1) RATE SELECTION USING LEARNED BROADCAST PERFORMANCE

We introduce the ACK-less rate adaptation framework [8] using the statistics learned by DQN and QR-DQN, in this section. This framework consists of learning and application phases. The agent acquires the rate-adaptation strategy using RL and simulations during the learning phase and then applies it to the real environment during the application phase. In the application phase, the agent observes the state from the environment and selects the action at every step. Notably, in the application phase, the eBCS AP cannot determine the success or failure of the eBCS STAs in real environments. That is, the agent cannot observe the reward from the environment. For the expected value of the reward learned by DQN, we simply used the greedy policy (7). The agent simply observed the state s_t and selected the data rate a_t, based on the DQN output.

For the reward distribution learned by QR-DQN, we used the risk-measure-based policy rather than the greedy policy (16):

$$a_t = \arg\max_{a \in \mathcal{A}} \mathrm{RM}(Z), \tag{19}$$

where $\mathrm{RM}(Z)$ is the risk measure for the distribution Z. As an example of risk measure, we used conditional value at risk (CVaR). CVaR is defined using the value at risk (VaR) measure:

$$\begin{aligned} \mathrm{RM}(Z) &= \mathrm{CVaR}_\alpha(Z) \\ &:= \mathbb{E}_{y \sim Y}[\, Y \mid Y < \mathrm{VaR}_\alpha(Y)\,], \end{aligned} \tag{20}$$

where VaR is formulated as

$$\mathrm{VaR}_\alpha(Y) := F_Y^{-1}(\alpha), \tag{21}$$

where F_Y is the cumulative distribution function of Y. VaR is not suitable for an event with a low probability and high risk. However, in this case, CVaR can accurately evaluate the risk by focusing on the lower end of the distribution, F_Y. In QR-DQN, CVaR was used as follows:

$$a_t = \arg\max_{a \in \mathcal{A}} \frac{1}{\lceil \alpha N_{\mathrm{q}} \rceil} \sum_{j=1}^{\lceil \alpha N_{\mathrm{q}} \rceil} \theta_j(s_t, a), \tag{22}$$

where $0 < \alpha \leq 1$ is a parameter of CVaR that represents the importance of the lower end of the value distribution. By using α near 0, the lower end of the distribution was emphasized to avoid the risk of incurring a reception failure. When $\alpha = 1$, the CVaR-based policy becomes equivalent to the greedy policy that was based on the expected value of the distribution. The overall procedure, consisting of the learning and application phases, is shown in Algorithm 1. Here, $N_{\mathrm{ep,l}}$ and $N_{\mathrm{step,l}}$ are the number of episodes and steps in learning phase, respectively, and $N_{\mathrm{ep,a}}$ and $N_{\mathrm{step,a}}$ are those in application phase, respectively.

Algorithm 1 Learning and Application Phases

```
# Learning phase
for e = 1, 2, ..., N_ep,l do
    Generate locations of all APs and STAs.
    for t = 1, 2, ..., N_step,l do
        Observe a state s_t from the environment and feed
        it to the DQN/QR-DQN.
        Select an action a_t using (6).
        Observe a reward r_t from the environment.
        Update weight parameters of the DQN/QR-DQN.
    end for
end for
# Application phase
for e = 1, 2, ..., N_ep,a do
    Generate locations of all APs and STAs.
    for t = 1, 2, ..., N_step,a do
        Observe a state s_t from the environment and feed
        it to the DQN/QR-DQN.
        Select an action a_t using (7) or (22).
    end for
end for
```

TABLE 3. Comparison of rate adaptation methods.

Method	Overhear uplink frames?	Learn statistics of reward?
Rule-based	Yes	No
DQN	Yes	Yes (expected value)
Proposed: QR-DQN with risk-measure-based policy	Yes	**Yes (distribution)**

2) COMPARED RULE-BASED METHOD

We introduce a rule-based rate adaptation method that does not use the learned statistics. A comparison of all the methods is shown in Table 3 Specifically, the eBCS AP selected the data rate a_t based on the following equation:

$$a_t = \max_a \{ a \mid \Gamma_{\mathrm{req}}(a) \leq \hat{\Gamma} \}, \tag{23}$$

where $\hat{\Gamma}$ is the estimated received SNR at the eBCS STA that transmits the uplink frames and is furthest from the eBCS AP. The estimated received SNR $\hat{\Gamma}$ was calculated from the path loss $L^{(j)}$, assuming symmetry between the eBCS AP and each eBCS STA, as follows:

$$\hat{\Gamma} := \frac{P_{\mathrm{eBCS}}}{\max_j L^{(j)} \cdot P_{\mathrm{n}}} \cdot \frac{1}{\beta}, \tag{24}$$

$$L^{(j)} := \frac{P_{\mathrm{STA}}}{p_t^{(j)}}, \tag{25}$$

where P_{n} is the noise power and P_{STA}, P_{eBCS} are the transmit powers of the STAs and the eBCS AP, respectively. The parameter $\beta > 1$ underestimates the received SNR to select a lower rate and avoid transmission failures. The eBCS AP was assumed to know P_{STA} by referring to the standard settings of the transmit power for IEEE 802.11 WLANs.

C. RESULTS

In this section, we illustrate some evaluation results and discuss the performance of the proposed methods. First, we evaluated the accuracy of the learned statistics of the reward, given three levels of RSS: high, middle, and low, p_{high}, p_{mid}, and p_{low}. The ground truth of the statistics was generated by Monte Carlo simulation. Since RSS is continuous, it required substantial computational costs to obtain a sufficient number of samples for each level of RSS. Therefore, for ease of computation, we loosened the scope of each level, i.e., we used the samples within $[p_{\mathrm{high}} - \Delta p/2, p_{\mathrm{high}} + \Delta p/2]$, $[p_{\mathrm{mid}} - \Delta p/2, p_{\mathrm{mid}} + \Delta p/2]$, and $[p_{\mathrm{low}} - \Delta p/2, p_{\mathrm{low}} + \Delta p/2]$, where Δp is the loosened width of the RSS levels. As an example, we set the parameters of this simulation as shown in Table 4. Second, for demonstration purposes, we simulated the rate adaptation of the eBCS AP, based on the learned statistics. Therein, we used the learned parameters of the DNN in DQN and QR-DQN, following the policy discussed in Section IV-B1.

1) LEARNED REWARD STATISTICS

Table 5 shows the comparison between the expected value learned by DQN and that computed by the Monte Carlo simulation. The results indicate that DQN was able to approximate the expected values. In particular, DQN output was able to accurately approximate the actions with the highest reward in each RSS level, which are the ones having the most importance in rate adaptation using learned statistics. Thus, the learned results are considered to be applicable to the ACK-less rate adaptation. Likewise, Fig. 3 confirms that the approximation method based on QR-DQN was able to learn the distribution of the reward. For ease of viewing, the distributions are only depicted for the actions that were frequently selected for each input RSS. The ground truth distributions generated by Monte Carlo simulations were empirical cumulative distributions, different from the output of QR-DQN, which directly appeared in the output layer of the DNN. Using these learned statistics, the eBCS AP was able to implement ACK-less rate adaptation in the considered eBCS system.

TABLE 4. Simulation parameters for generating RSS.

Number of Samples	10000
RSS level $(p_{\mathrm{high}}, p_{\mathrm{mid}}, p_{\mathrm{low}})$	$(-81.5, -86.5, -94.5)$ dBm
Loosened width of RSS level Δp	1.0 dB

TABLE 5. Comparison of expected values of reward.

(RSS level, data rate (Mbit/s))	Ground truth	DQN
$(p_{\mathrm{high}}, 8.6)$	0.060	0.060
$(p_{\mathrm{high}}, 51.6)$	0.32	0.33
$(p_{\mathrm{high}}, 103.2)$	**0.36**	**0.37**
$(p_{\mathrm{high}}, 143.4)$	−0.71	−0.54
$(p_{\mathrm{mid}}, 8.6)$	0.060	0.060
$(p_{\mathrm{mid}}, 51.6)$	**0.30**	**0.30**
$(p_{\mathrm{mid}}, 103.2)$	−0.41	−0.22
$(p_{\mathrm{mid}}, 143.4)$	−0.91	−0.78
$(p_{\mathrm{low}}, 8.6)$	**0.060**	**0.060**
$(p_{\mathrm{low}}, 51.6)$	−0.14	−0.073
$(p_{\mathrm{low}}, 103.2)$	−0.65	−0.46
$(p_{\mathrm{low}}, 143.4)$	−0.96	−0.83

2) ACK-LESS RATE ADAPTATION RESULTS

Using the ACK-less rate adaptation framework, we first performed the simulation using various distances B and radii σ. Fig. 4 exhibits the tradeoff between the data rate and success rate. The parameter α of CVaR and β of the rule-based method were set to various values. Fig. 4 confirms that the proposed DQN and QR-DQN with risk-measure-based policy were able to improve the success rate and data rate in terms of the Pareto efficiency, compared with the rule-based method. This indicates that ACK-less rate adaptation using learned statistics of the reward can improve the broadcast performance in the considered system.

Fig. 4 also demonstrates that as the CVaR parameter α becomes larger, the success rate of the QR-DQN with risk-measure-based policy increases while implementing a rate adaptation in the same way as the method that learns only expected broadcast performance. As mentioned in Section III, setting a smaller value of α would imply emphasizing the

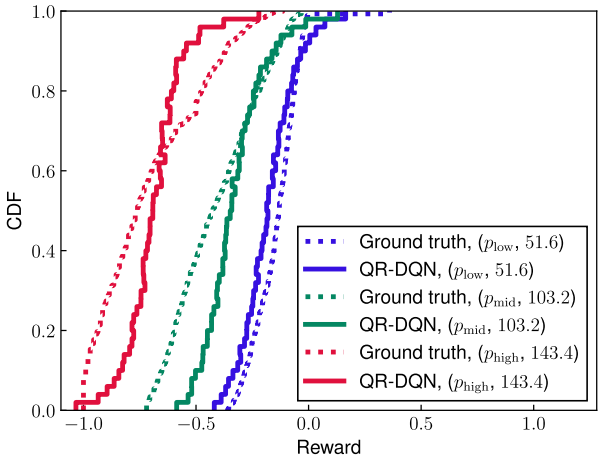


FIGURE 3. Comparison of the reward distributions between the ground truth and QR-DQN output. The ground truth is generated by Monte Carlo simulations. The vertical axis of the DQN output displays the probabilities of the quantiles $\hat{\tau}_i$, and the horizontal axis displays the approximated returns of the quantiles θ_i. For ease of viewing, the distributions are depicted only for the actions with wider distributions.

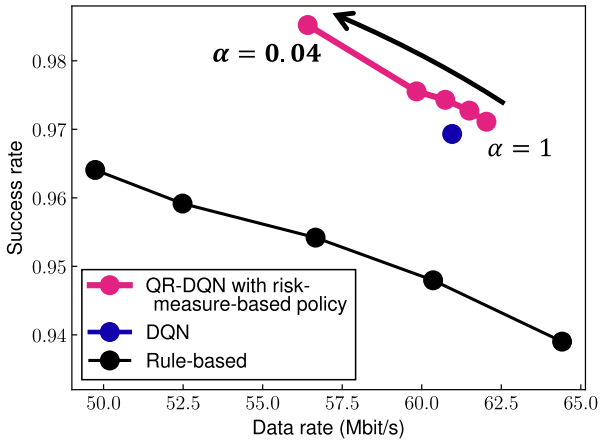


FIGURE 4. Data rate of the eBCS AP vs. reception success rate for the eBCS STAs. The data rate and success rate are averaged over a variety of distances B and radii σ, and the value of m is fixed at 10.

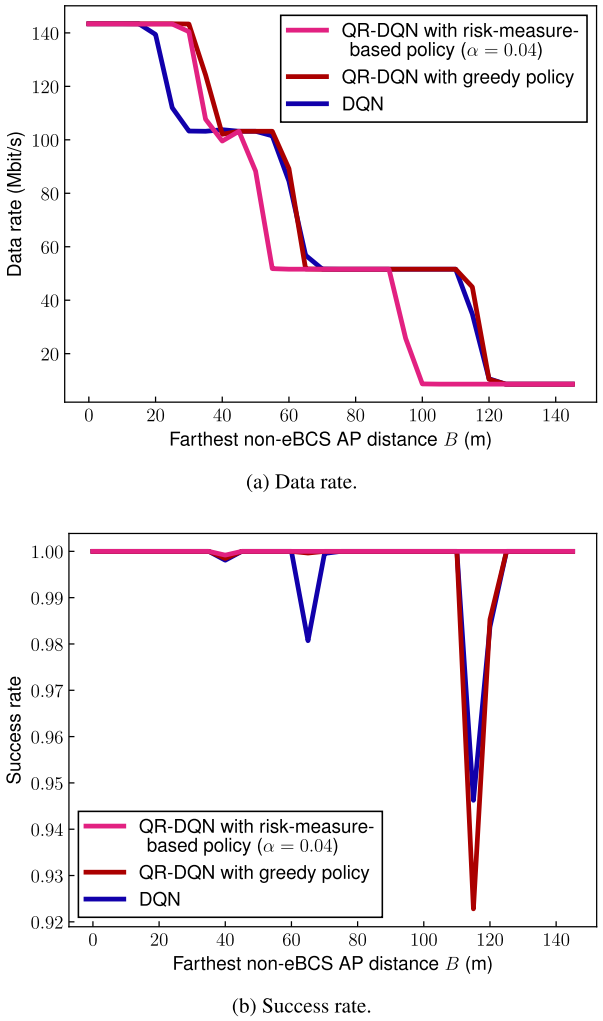


FIGURE 5. Data rate and success rate against distance B. The radius σ is fixed to 10 m, and the value of m is fixed at 10.

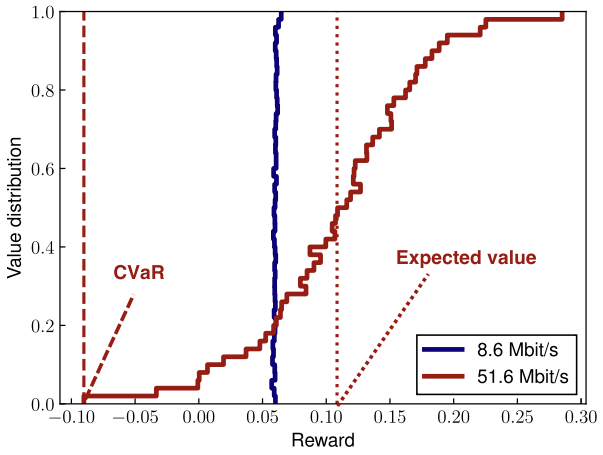
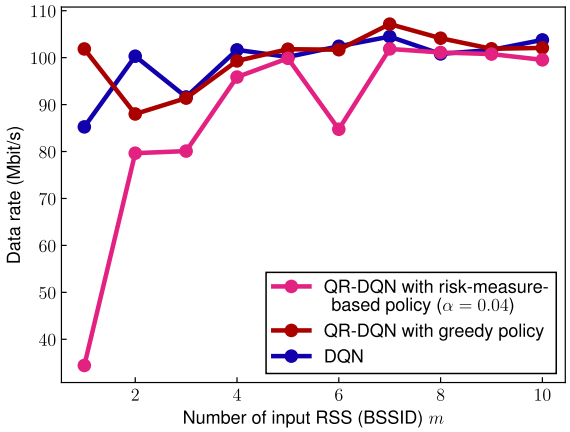


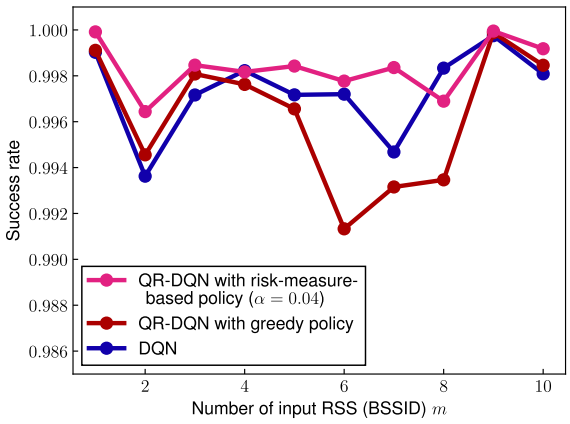
FIGURE 6. Learned value distribution at distance $B = 100$ m. The CVaR parameter is $\alpha = 0.04$, the radius is $\sigma = 10$, and the value of m is fixed at 10.

lower end of the reward distribution that would enable the eBCS AP to avoid the risk of incurring reception failures. Hence, by making α smaller, we could achieve a higher success rate while implementing the rate adaptation in the same way as the method that learns only expected broadcast performance. This confirms the feasibility of the method for improving the success rate by utilizing the learned reward distribution with the risk measure instead of the expected value.

Fig. 5(a) shows that the data rate of the eBCS AP is switched when the distance B is approximately 30, 60, and 110 m. Note that the QR-DQN with greedy policy is realized by $\alpha = 1$. In the simulation, m was kept fixed, and the data rate and success rate were averaged over a variety of distances B and radii σ. The result indicates that the eBCS AP adapted the data rate to a variety of distances between the eBCS AP and STA cluster. More specifically, when the distance was small, the eBCS AP selected a higher data rate, and when the distance was large, it selected a lower data rate.



(a) Data rate.



(b) Success rate.

FIGURE 7. Data rate and success rate against *m*. The distance *B* is fixed to 40 m and the radius σ is fixed to 10 m.

However, Fig. 5(b) shows that reception failures occur when the data rate is switched. In particular, the success rates of the QR-DQN with greedy policy and DQN are seen to drop at distances 60, 110 m, where the data rate is switched. This arises from the fact that even when the observed states are the same, the action to maximize the reward is different since it is possible that the same states represent different deployments. This is due to the existence of the eBCS STAs whose RSS information was not contained in the uplink frames, which rendered the reward distribution wider at such distances. The QR-DQN with greedy policy and the DQN use the greedy policy in the application phase; low reward with low probability that arises from the wider reward distribution cannot be addressed by these methods properly.

On the other hand, the QR-DQN with risk-measure-based policy achieved a higher success rate at distances around 60, 110 m, compared to other methods. This is due to the fact that the QR-DQN with risk-measure-based policy selects a data rate that is one level lower even at slightly shorter distances, i.e., at 55, 100 m, as shown in Fig. 5(a). It can be considered that the QR-DQN with risk-measure-based policy method with CVaR-based policy addresses the risk of reception failures appearing at the lower end of the distribution.

Fig. 6 shows the learned value distribution obtained using the QR-DQN algorithm, where the vertical axis represents the probabilities of the quantiles $\hat{\tau}_i$, and the horizontal axis represents the approximated rewards of the quantiles θ_i. It must be noted that the approximated rewards, θ_i directly appear in the output of the DNN used in QR-DQN. According to the figure, the CVaR-based policy selects a low data rate by emphasizing the lower end of the distribution while the greedy policy selects a high data rate because of the high expected value. This is caused by the wider value distribution of the data rate of 51.6 Mbit/s than that of the data rate of 8.6 Mbit/s, which implies that selecting 51.6 Mbit/s contains the potential risk of a low reward, i.e., a low success rate. In this case, the CVaR-based policy selected 8.6 Mbit/s as $\mathrm{CVaR}_{0.04}(Z(s_t, 8.6)) > \mathrm{CVaR}_{0.04}(Z(s_t, 51.6)$ while the greedy policy selected 51.6 Mbit/s as $\mathbb{E}[Z(s_t, 51.6)] > \mathbb{E}[Z(s_t, 8.6)]$. Roughly speaking, the CVaR-based policy focused on this risk while the eBCS AP in the QR-DQN with risk-measure-based policy avoided it by selecting the appropriate data rates. Additionally, Fig. 7 shows the changes in the data rate and the success rate, with respect to the number of input RSS, m. This confirms the high success rate of the proposed method, regardless of the value of m.

V. CONCLUSION

This paper addressed the challenge of the previous ACK-less rate adaptation framework that it incurs the reception failures at a part of the eBCS STAs. To address this challenge, this paper adopted distributional RL, which approximates the entire distribution of the broadcast performance. The impact of the difference in the targets to learn, i.e., the statistics or the distribution, on the broadcast performance of the ACK-less rate adaptation is studied to use high data rate while avoiding reception failures. The simulations confirmed that: 1) Using learned statistics improved the success and data rates as measured in terms of the Pareto efficiency, as compared with the rate adaptation method. 2) Learning the entire distribution of the indicator enabled the eBCS AP to determine the tail of the distribution using risk measure, and to alleviate reception failures while implementing rate adaptation in the same way, as the method that learns only expected broadcast performance.

In this paper, we assumed that the reception success/failure information was available while the agent was learning the statistics. However, this is not applicable to the real environment because the eBCS AP cannot obtain this information since ACK mechanisms would not be implemented. A feasible solution to this problem would be to design a sim-to-real framework for this ACK-less rate adaptation. Therein, learning the statistics/distribution of the broadcast performance is conducted via a simulation where the success/failure information in all eBCS STAs would be available, and the learned statistics could be applied to real-world deployments. Hence, implementing this sim-to-real framework based on

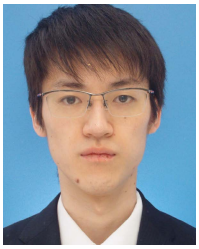
network simulators and demonstrating the applicability of learning the statistics/distribution to real-world environments is a possible future direction of this study.

REFERENCES

[1] C. Ansley, H. Morioka, M. Emmelmann, S. McCann, X. Wang, and A. Patil, *802.11bc Functional Requirements Document*, IEEE document 802.11-19/0151r5, Sep. 2019.
[2] X. Wang, H. Morioka, B. Sadeghi, C. Kain, Y. Inoue, J. Boyer, S. McCann, H. Mano, and A. Patil, *TGbc Use Case Document*, IEEE document 802.11-19/0268r5, Jul. 2019.
[3] H. Morioka and H. Mano, *Broadcasting on WLAN*, IEEE document 802.11-17/0999r1, Jul. 2017.
[4] R. Yavatkar and L. Manoj, "Optimistic strategies for large-scale dissemination of multimedia information," in *Proc. 1st ACM Int. Conf. Multimedia*, 1993, pp. 13–20.
[5] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Tech. J.*, vol. 2, no. 3, pp. 118–133, May 1997.
[6] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. dissertation, MIT, Cambridge, MA, USA, 2005.
[7] *Minstrel Rate Adaptation Algorithm Documentation*. Accessed: Oct. 20, 2021. [Online]. Available: http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt
[8] T. Kanda, Y. Koda, K. Yamamoto, and T. Nishio, "ACK-less rate adaptation for IEEE 802.11bc enhanced broadcast services using sim-to-real deep reinforcement learning," in *Proc. IEEE CCNC*, Jan. 2022, pp. 139–143.
[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[10] 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, 2015.
[11] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. ICML*, Sydney, NSW, Australia, Aug. 2017, pp. 449–458.
[12] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proc. AAAI*, New Orleans, LA, USA, Feb. 2018, pp. 2892–2901.
[13] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Math. Finance*, vol. 9, no. 3, pp. 203–228, Jul. 1999.
[14] J. Bernhard, S. Pollok, and A. Knoll, "Addressing inherent uncertainty: Risk-sensitive behavior generation for automated driving using distributional reinforcement learning," in *Proc. IEEE IV*, Paris, France, Jun. 2019, pp. 2148–2155.
[15] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Psychol., Univ. Cambridge, Cambridge, U.K., 1989.
[16] *Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE Standard 802.11ax-2021, Dec. 2021.
[17] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.
[18] J. Liu *et al.*, *IEEE 802.11ax Channel Model Document*, IEEE document 802.11-14/0882r4, Sep. 2014.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[20] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 293–321, May 1992.

TAKAMOCHI KANDA (Graduate Student Member, IEEE) received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2021, where he is currently pursuing the M.I. degree with the Graduate School of Informatics.

YUSUKE KODA (Member, IEEE) received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2016, and the M.E. and Ph.D. degrees in informatics from the Graduate School of Informatics, Kyoto University, in 2018 and 2021, respectively. He is currently a Postdoctoral Researcher with the Centre for Wireless Communications, University of Oulu, Finland. In 2019, he visited the Centre for Wireless Communications, University of Oulu, to conduct collaborative research. He received the VTS Japan Young Researcher's Encouragement Award, in 2017, and the TELECOM System Technology Award, in 2020. He was a recipient of the Nokia Foundation Centennial Scholarship, in 2019.



YUTO KIHIRA (Student Member, IEEE) received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2020, where he is currently pursuing the M.I. degree with the Graduate School of Informatics.

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. 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, Royal Institute of Technology (KTH), Sweden. His research interests include radio resource management, game theory, and machine learning. From 2004 to 2005, he was a Research Fellow of the Japan Society for the Promotion of Science (JSPS). He is a Senior Member of IEICE and a member of the Operations Research Society of Japan. He was a Tutorial Lecturer in ICC 2019. He has received the PIMRC 2004 Best Student Paper Award, in 2004, and the Ericsson Young Scientist Award, in 2006. He has also received the Young Researcher's Award, the Paper Award, the SUEMATSU-Yasuharu Award, and the Educational Service Award from 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 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. From 2013 to 2020, he was an Assistant Professor with the Graduate School of Informatics, Kyoto University. From 2016 to 2017, he was a Visiting Researcher with the Wireless Information Network Laboratory (WINLAB), Rutgers University, USA. Since 2020, he has been an Associate Professor with the School of Engineering, Tokyo Institute of Technology, Japan. His current research interests include machine learning-based network control, machine learning in wireless networks, and heterogeneous resource management.