Model-Free Ultra Reliable Low Latency Communication (URLLC): a Deep Reinforcement Learning Framework

Ali Taleb Zadeh Kasgari and Walid Saad
Electrical and Computer Engineering Department, Blacksburg, Virginia Tech, VA, USA, Emails:{alitk,walids}@vt.edu

***Abstract*—In this paper, a novel deep reinforcement learning (deep-RL) framework is proposed to provide model-free ultra reliable low latency communication (URLLC) in the downlink of an orthogonal frequency division multiple access (OFDMA) system. The proposed deep-RL framework can guarantee high end-to-end reliability and low end-to-end latency, under data rate constraints, for each user in the cellular system without any models of or assumptions on the users' traffic. Using the proposed model-free approach, the users' traffic is predicted by the deep-RL framework and subsequently used in the resource allocation, irrespective of the actual underlying model. The problem is posed as a power minimization problem under reliability, latency, and rate constraints. To solve this problem using deep-RL, first, the rate of each user is determined. Then, these rates are mapped to the resource block and power allocation vectors of the studied OFDMA system. Finally, the end-to-end reliability and latency of each user are used as a feedback to the deep-RL framework. It is shown that at the fixed-point of the deep-RL algorithm, the reliability and latency of the users are guaranteed. Simulation results show how the proposed approach can achieve any feasible point in the rate-reliability-latency region, depending on the network and service requirements. For example, for a 7 Mbps rate guarantee, the results show that the proposed algorithm can provide ultra-reliable low latency communication with a delay of 8 milliseconds and a reliability of 98%.**

I. Introduction

Ultra reliable low latency communication (URLLC) is expected to be one of the most important features in 5G cellular networks since it will be used for mission critical applications such as Internet of Things (IoT) sensing and control as well as remote control of autonomous vehicles and drones [1]. Thus far, the URLLC research has been mostly focused on the applications with short packet length and low data rates such as uplink transmissions of IoT sensors [1]. However, new wireless applications such as drone communications [2]–[4], autonomous driving [5], [6], and virtual reality [7], have emerged that require URLLC, not only in the uplink, but also in the downlink for control and tracking purposes. Moreover, in order to operate effectively, such applications require both URLLC and reasonably high data rates. For example, autonomous vehicles will need to receive reliable control data from infrastructure nodes, along with high data rate information such as HD maps.

Providing URLLC with rate guarantees for such applications poses many network challenges. First, considering the limited radio resources in a communication system, low latency, high reliability, and high rate become three incompatible design parameters. This incompatibility means that improving one of them could potentially be detrimental to the other two, thus requiring *new designs that can balance the rate-reliability-latency tradeoff*. Second, providing reasonably high data rates while maintaining reliability and latency needs timely and efficient resource allocation. Hence, URLLC resource allocation with rate considerations should allocate the exact amount of resources required by the users. In other words, to balance the rate-reliability-latency tradeoff, the resource allocation scheme must know each user's exact performance needs so that it can satisfy these requirements without wasting any resources or reducing the user's reliability.

This research was supported by the U.S. National Science Foundation under Grant CNS-1836802.

A. Related Works

Recently, there has been a surge in literature that studies problems of URLLC and resource allocation, such as in [1], [8]–[12]. In [8], an algorithm for joint scheduling of URLLC and broadband traffic in 5G cellular systems is proposed. In [9], the authors use extreme value theory to study URLLC in a vehicular network and characterize the queue statistics. The work in [10] considers a model-based and a data-driven approach for designing a burstiness-aware scheduling framework which reserves bandwidth for the users with bursty traffic. The authors in [11] propose a packet prediction mechanism to predict the behavior of future incoming packets based on the packets in the current queue. However, all of these existing works assume that explicit traffic and queue models are available to the resource allocation system [8]–[11]. Since the assumed models are often simplified, they either underestimate or overestimate the users' traffic and queue lengths. This can cause the resource allocation algorithm to either allocate more resources or less resources than the actual requirement of the users which, in turn, can render the system inefficient or degrade the reliability of the users' connections. Also, the previous works on URLLC completely ignore any rate requirements of the users. Moreover, some of these works rely on completely historical data such as [10] and [11] which might also lead to inefficient resource allocation. This due to the fact that a user's traffic often changes based on spatial or temporal factors, and historical data is often not a good predictor of the traffic of wireless users.

To overcome some of these challenges, there has been recent interest in using deep reinforcement learning (deep-RL) for solving wireless networking problems with incomplete information such as in [5], [13]–[16]. In [5], a decentralized resource allocation framework for vehicle-to-vehicle communication is proposed based on deep-RL. The authors in [13] proposed a deep-RL resource management approach for virtualized ad-hoc network. In [14] a deep-RL resource management system is proposed for cloud radio access networks. A deep-RL based resource allocation scheme for mobile edge computing is studied in [16] However, these works do not

investigate URLLC problems. Moreover, since these deep-RL works [14], [16] limited their problem's action space, used discretization to limit the size of the action space [5], or did not address the limitation of deep-RL when dealing with large action spaces, they are not suitable to solve the problem of URLLC resource allocation with rate constraints. This is due to the fact that these works cannot handle the large action space involved in URLLC and they are slower than the requirements of URLLC.

B. Contributions

The main contribution of this paper is a novel, model-free resource management framework that can balance the tradeoff between reliability, latency, and data rate, without explicit prior assumptions on the users' traffic arrival model. We formulate the problem as a power minimization problem under reliability and latency constraints. To solve this problem, we propose a deep-RL framework that dynamically predicts the traffic model of the users and, then, uses those predictions to jointly allocate resource blocks (RBs) and power to downlink users, under URLLC and rate constraints. This framework is shown to effectively find a feasible resource allocation solution such that the low latency, high reliability, and high rate requirements of the wireless users are satisfied. The proposed framework dynamically measures the end-to-end reliability and the delay of each user. Then, it uses this measurement as an online feedback to modify its decisions. In particular, the deep neural network (DNN) weights used in deep-RL are updated using this feedback only. Therefore, our framework does not need any collected dataset for training. Also, the proposed resource allocation system is able to predict the consequences of its actions in the future and use this information to make better resource allocation decisions. This helps the algorithm provide long term reliability and latency guarantees for the users.

Unlike the deep RL approaches that were previously used for wireless networks [5], [13], [14], [16], our approach addresses the large action space problem. In particular, we propose the novel concept of an action space reducer which reduces the size of the action space without limiting it. Using this action space reducer, our deep-RL framework is able to make decisions in real-time as opposed to discretization approach used in [5]. We show that when the proposed algorithm converges, the reliability, latency, and rate of each user are guaranteed. Simulation results show that, without any knowledge or assumption for the traffic model, our algorithm is able to reach any feasible combination of rate, reliability, and latency, given the system's bandwidth and power constraints. For example, to enable URLLC downlink communication at rate of 7 Mbps, latency of 8 milliseconds and reliability of 98% are achievable.

The rest of the paper is organized as follows. Section II introduces the system model. Sections III present the proposed deep-RL framework. Section IV presents the simulation results and conclusions are drawn in Section V.

II. System Model

Consider the downlink of an OFDMA cellular network with a single base station (BS) serving a set $\mathcal{N}$ of N users and having a set $\mathcal{K}$ of K available RBs. Each user has its own, individual rate, reliability, and latency requirements. We do not make any assumptions on the packet arrival or the packet length of each user. The downlink transmission rate from the BS to user i is:

$$r_i(t) = \sum_{j=1}^{K} \rho_{ij}(t) B \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{\sigma^2}\right), \quad (1)$$

where B is the RB bandwidth and $h_{ij}(t)$ is the time-varying Rayleigh fading channel gain of the transmission from the BS to user i on RB j at time slot t. $p_{ij}(t)$ is the downlink transmission power of the BS on RB j to user i at slot t. $\rho_{ij}(t)$ is the RB allocation indicator with $\rho_{ij}(t) = 1$ when RB j is allocated to user i at time slot t, otherwise $\rho_{ij}(t) = 0$.

We define *reliability* $\gamma_i(t)$ as the probability of the end-to-end packet delay exceeding a predefined target end-to-end latency threshold $D_i^{\max}$ for user i. This delay comprises the queuing delay and the transmission delay. To satisfy the reliability and latency condition, the system needs to maintain its rate according to the packet arrival rate, i.e,

$$r_i(t) > \phi(\lambda_i(t), \beta(t), \gamma_i, D_i^{\max}) > \lambda_i(t)\beta_i(t), \quad (2)$$

where $\beta_i(t)$ is the average packet size and $\lambda_i(t)$ is the average packet arrival rate of user i at time-slot t. ϕ is an unknown function that we will implicitly approximate using our proposed method. The goal for the system is to allocate resources so as to minimize the average downlink power while maintaining reliability, latency, and rate for the users. We formally pose this resource allocation problem as follows:

$$\min_{p_{ij},\rho_{ij}} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{i=1}^{N} \sum_{j=1}^{K} p_{ij}(\tau), \quad (3a)$$

$$\text{s.t} \quad \Pr\{D_i > D_i^{\max}\} < 1 - \gamma_i^*, \quad \forall i \in \mathcal{N}, \quad (3b)$$

$$p_{ij}(t) \geq 0, \quad \rho_{ij}(t) \in \{0,1\}, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{K}, \quad \forall t, \quad (3c)$$

$$\sum_i \rho_{ij}(t) = 1, \quad \forall j \in \mathcal{K}, \quad \forall t. \quad (3d)$$

In (3b), D_i is the packet delay for user i. Constraint (3b) takes into account each user's reliability and latency explicitly. We do not consider the rate constraint explicitly in our problem formulation, but it is considered implicitly using (2).

The objective function (3a) is the average power spent by the BS. Constraint (3b) is a reliability condition that captures the fact that the end-to-end delay should be less than $D_i^{\max}$ with a reliability of at least $1 - \gamma_i^*$. Constraints (3c) and (3d) are feasibility conditions. At each time slot t, the resource allocation system has two functions: *Phase 1*: Determining the rate that each user i should obtain in order to ensure target reliability γ_i^* for this user, and *Phase 2*: Allocating power and RBs to each user so that the power is minimized. We

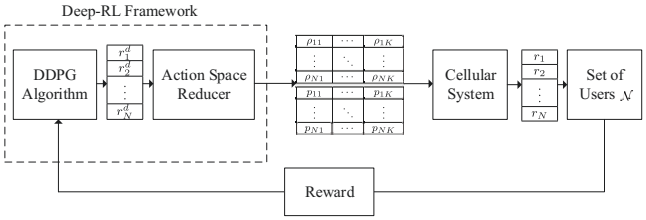

Figure 1: Block diagram for the proposed framework.

should note that the minimum power in Phase 2 is a function of the data rates determined in Phase 1. To determine the reliability of the system in (3b), it is customary to use a specific queuing model, as done in all of the prior art [1], [8], and [10]. In contrast, here, we propose to obtain the reliability in (3b) using an empirical measurement of the number of packets transmitted to user i whose delay exceeds $D_i^{\max}$ over the total number of packets transmitted (to user i) in time slot t, i.e.,

$$\gamma_i(t) = 1 - \Pr\{D_i > D_i^{\max}\} \approx 1 - \frac{\mu_i'(t)}{\mu_i(t)}, \tag{4}$$

where $\mu_i'(t)$ is the number of packets transmitted to user i in time slot t, whose end-to-end delay exceeds $D_i^{\max}$. $\mu_i(t)$ is the total number of packets transmitted to user i in time slot t. By doing so, we do not need to make any a priori assumptions on the delay model of the users. Moreover, counting the number of packets is an easy and practical feedback, because each user can convey this number to the BS via a control channel. As $\mu_i(t)$ grows, the approximation converges to the reliability in (3b). As will be evident from Section IV, despite having no model for the traffic, the proposed approach will still be able to ensure target reliability and delay for each user.

Since the OFDMA resource allocation problem involves a large state space and we have no prior knowledge of the traffic models, we propose a deep-RL framework [17] to allocate resources to the users so that their rate requirement and their stringent reliability constraints are satisfied. Beyond being able to operate without any model, the key advantage of deep-RL over classical reinforcement learning (RL) is that it can solve control problems with a large state space [17]. Deep-RL uses a deep neural network (DNN also known as deep Q-network) for approximating the action-value function (also known as Q-function) in RL.

The proposed deep-RL framework will use two feedback inputs to evaluate its performance and update its DNN in each time slot: the BS downlink power in each time slot $P(t) = \sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij}(t)$, and the measured reliability of each user at each time using (4). Using these two feedbacks, the deep-RL framework can determine $\rho_{ij}(t)$ and $p_{ij}(t)$ for all i and j. After iteratively assigning $\rho_{ij}(t)$ and $p_{ij}(t)$ and receiving feedbacks in a few time slots, the system is able to maintain reliability, latency, and rate for each user. We will discuss Phase 1 and Phase 2 in Subsection III-C and III-B, respectively.

III. Deep-RL for model free URLLC

The block diagram for the proposed deep-RL framework is shown in Fig.1. As we can see from Fig 1, at each time slot, the deep-RL algorithm determines a desired rate $r_i^d(t)$ for each user i. Next, the action space reducer maps $r_i^d(t)$ to the OFDMA resources ρ_{ij} and p_{ij} for all i and j while minimizing the power (Section III-B). Then, each user attains the rate $r_i(t)$ (which is now close to $r_i^d(t)$) and finds a reward function (defined later) and sends it as a feedback to the deep-RL framework. Finally, the deep-RL system uses this feedback and updates the $r_i^d(t)$ for each user accordingly (Section III-C).

A. Deep-RL scheduling

In this section, we define our deep-RL framework for resource allocation. In general, any deep-RL framework is defined by its action-space $\mathcal{A}$, state-space $\mathcal{S}$, and reward R. At each state $\boldsymbol{s}(t) \in \mathcal{S}$, a deep-RL algorithm takes action $\boldsymbol{a}(t) \in \mathcal{A}$ and receives the reward $R(\boldsymbol{a}(t), \boldsymbol{s}(t))$. For our wireless resource allocation problem, we consider the channel gains, the number of packets $\mu_i(t)$ transmitted to each user, and the average packet length $\beta_i(t)$ for each user i as *the state* for the proposed deep-RL framework.Then, we determine the reward for deep-RL so as to guarantee URLLC without the delay model. Deep-RL will use this reward for training the DNN and approximating the action-value function. We define the reward for the proposed deep-RL framework as a function of power and reliability. However, it is implicitly a function of state and action of the deep-RL algorithm. The reward is defined as follows:

$$R(\boldsymbol{a}(t), \boldsymbol{s}(t)) = -\sum_{i\in\mathcal{N}} w_i(t)(1-\gamma_i(t)) - \alpha P(t), \tag{5}$$

where α is a weighting factor for power and $w_i(t)$ is given by:

$$w_i(t+1) = \max\{w_i(t) + \gamma_i^* - \gamma_i(t), 0\}. \tag{6}$$

We can see that $w_i(t)$ is a time-varying weight that increases if $\gamma_i(t) < \gamma_i^*$. Hence, it ensures that the system meets the target reliability of the users. Next, we show that, when the deep RL algorithm maximizes the reward defined in (5), the reliability and delay in (3b) are guaranteed for each user.

Theorem 1. If the BS maximizes the reward in (5), then after the convergence of the deep-RL algorithm, the reliability of each user is guaranteed such that $\gamma_i(t) \geq \gamma^* \, \forall i \in \mathcal{N}$.

Proof: First, assume that the value that $w_i(t)$ converges to is w_i^*. Then, we have to show that

$$\|w_i(t+1) - w_i^*\|^2 = \|\max\{w_i(t) + \gamma_i^* - \gamma_i(t), 0\} - w_i^*\|^2 =$$
$$\|w_i(t) + \gamma_i^* - \gamma_i(t)\|^2 + \|w_i^*\|^2 - 2w_i^{*T}(w_i(t) + \gamma_i^* - \gamma_i(t)) =$$
$$\|w_i(t) - w_i^*\|^2 + \|\gamma_i^* - \gamma_i(t)\|^2 - 2(w_i^* - w_i(t))^T(\gamma_i^* - \gamma_i(t)),$$

Hence,

$$\|w(t+1) - w^*\|^2 - \|w(t) - w^*\|^2 =$$
$$\|\gamma_i^* - \gamma_i(t)\|^2 - 2(w^* - w(t))^T(\gamma_i^* - \gamma_i(t)). \tag{7}$$

Also, from (6) we know that $\gamma_i^* - \gamma_i(t) \leq w(t+1) - w(t)$. We can see that, in the case of one dimensional $w(t)$, the stability condition reduces to $(\gamma_i^* - \gamma_i(t))\,(\gamma_i^* - \gamma_i(t) - 2(w^* - w(t))) \leq 0$.

At the fixed point of the algorithm, we know that $w(t+1) = w(t)$, therefore

$$w(t) + \gamma_i^* - \gamma_i(t) \leq \max\{w(t) + \gamma_i^* - \gamma_i(t), 0\} = w(t), \quad (8)$$

Thus, we can see that $w(t) + \gamma_i^* - \gamma_i(t) \leq w(t)$ and hence, $\gamma_i(t) \geq \gamma_i^*$. ∎

Theorem 1 ensures that the reliability of each user is guaranteed at the fixed point of the algorithm, i.e., when $w(t) = w(t+1)$. Also, the latency for each user is implicitly guaranteed by Theorem 1. The original action space for the deep-RL resource allocation problem is the possible set for ρ_{ij} and p_{ij} for all i and j which has the size of $\mathcal{O}(K^N) \times \mathbb{R}^K$. Therefore, in our URLLC problem, we have a large state space and a large action space. Using deep Q-networks helps us to address the large state space problem. However, we have to address the large action space problem as well. Next, we propose a mechanism, called *action space reducer*, using which we reduce the size of the action space.

B. Reducing Action Space by Optimal Power Allocation

The action space for the studied wireless resource allocation problem consists of the $N \times K$ RB allocation matrix and $N \times K$ power allocation matrix as:

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1K} \\ \vdots & \ddots & \vdots \\ \rho_{N1} & \cdots & \rho_{NK} \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} p_{11} & \cdots & p_{1K} \\ \vdots & \ddots & \vdots \\ p_{N1} & \cdots & p_{NK} \end{bmatrix}.$$

Our mixed integer action space size is $\mathcal{O}(K^N) \times \mathbb{R}^K$ and it is infeasible to use this action space in a deep-RL algorithm. To address this problem, we first use deep deterministic policy gradient (DDPG) [18] algorithm to take actions in a small action space. Then, we map the actions taken by the DDPG algorithm to the original action space using an optimization framework. This approach is appropriate for our problem because our reduced action space is continous in $\mathbb{R}^N$ and DDPG can control such problems. Moreover, the choice of the DDPG is motivated by the fact that it has been successfully deployed in continuous action spaces in robotic problems [18]. DDPG uses a DNN for mapping the state space to an action space and hence is fast in decision making.

Since we are able to control the reliability and latency with the rate, we choose the reduced action space to be the rate of each user. However, given the set of rates for each user, there are many feasible power and RB allocation solutions. We choose the allocation solution with minimum power usage. Hence, we pose a new optimization problem, called *action space reducer*, whose goal is to map the reduced action space which is the rate for each user to the original action space, i.e., RB and power allocation matrices as output so that the power is minimized. This optimization problem maps the action space of our deep-RL algorithm to the optimization variables in (3). To find this RB and power allocation solution, we formally define the action space reducer problem:

$$\min_{\boldsymbol{P},\boldsymbol{\rho}} \quad \sum_{i,j} p_{ij}(t) \quad (9a)$$

$$\text{s.t.} \quad r_i(t) = r_i^d(t), \quad \forall i \in \mathcal{N} \quad (9b)$$

$$p_{ij}(t) \geq 0, \quad \rho_{ij}(t) \in \{0,1\}, \quad \forall i \in \mathcal{N}, \quad j \in \mathcal{K}, \quad \forall t, \quad (9c)$$

$$\sum_i \rho_{ij} = 1, \quad \forall j \in \mathcal{K}, \quad (9d)$$

where constraint (9b) guarantees that the rate of each user $r_i(t)$ is set to the desired rate for each user $r_i^d(t)$ while minimizing the total BS power. We can solve (9) with constraint (9b) in the form of an inequality, i.e., $r_i(t) \geq r_i^d(t)$ using an iterative dual decomposition algorithm. As the number of RBs increases, the primal solution converges to the dual solution and the inequality constraint $r_i(t) \geq r_i^d(t)$ will be satisfied in the form of equality [19]. As we will show in Section IV, as the number of RBs increases, the error for action space reducer decreases.

The Lagrangian for problem (9) with inequality constraint $r_i(t) \geq r_i^d(t)$ can be written as:

$$L(\boldsymbol{P}, \boldsymbol{\rho}, \boldsymbol{\lambda}) = \sum_{i,j} p_{ij}(t) - \sum_i \lambda_i (r_i(t) - r_i^d(t)), \quad (10)$$

where $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_N \end{bmatrix}^T$, and:

$$r_i(t) = B \sum_{j=1}^{K} \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{\sigma^2}\right). \quad (11)$$

The dual problem for (9) is:

$$\min_{\boldsymbol{P},\boldsymbol{\rho}} L(\boldsymbol{P}, \boldsymbol{\rho}, \boldsymbol{\lambda}). \quad (12)$$

We can see that the dual problem is decomposable for each RB, i.e., we can write (12) as:

$$\min_{p_{ij}(t)} \sum_i p_{ij}(t) - B \sum_i \lambda_i \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{\sigma^2}\right), \quad \forall j \in \mathcal{K}. \quad (13)$$

Each subproblem (13) is convex and has a closed-form solution. By taking the derivative with respect to $p_{ij}(t)$, we have:

$$1 - \lambda_i B \frac{h_{ij}(t)}{(\sigma^2 + p_{ij}^* h_{ij}(t)) \log 2} = 0, \quad \forall i \in \mathcal{N} \quad (14)$$

Hence, for each j we have:

$$p_{ij}^* = \left[\frac{\lambda_i B}{\log 2} - \frac{\sigma^2}{h_{ij}(t)}\right]^+, \quad \forall i \in \mathcal{N}, \quad (15)$$

where $[.]^+$ is equivalent to $\max\{., 0\}$. Since each RB can be allocated only to one user, we choose to allocate RB j to user i_j where:

$$i_j = \underset{i}{\operatorname{argmin}}\, p_{ij}^* - B\lambda_i \log_2 \left(1 + \frac{p_{ij}^* h_{ij}(t)}{\sigma^2}\right), \quad \forall j \in \mathcal{K}. \quad (16)$$

Therefore, we find the RB allocated to each user using (16) and the per-RB power using (15). The only parameter that remains to be determined is $\boldsymbol{\lambda}$, which can be derived using the ellipsoid method [20].

After reducing the size of action space, the action space becomes $\boldsymbol{a} = \begin{bmatrix} r_1^d & \cdots & r_N^d \end{bmatrix} \in \mathbb{R}^N$, which in the case of discretization to m levels, has the size $\theta(m^N)$, and is still not a very scalable solution. However, since the reduced action space is in $\mathbb{R}^N$, we can find a solution to our deep-RL problem using DDPG, as shown next.

C. Optimal Rate Allocation with Policy Gradient

A policy gradient algorithm such as DDPG can determine $r_i^d(t)$. This is due to the fact that the reduced action space is continuous and using that, we can estimate the gradient of the expected reward of the deterministic policy [18]. A deterministic policy is a function that maps each state $\boldsymbol{s}$ of the system to a specific action $\boldsymbol{a}$, i.e., $\boldsymbol{a} = \mu_{\boldsymbol{\theta}}(\boldsymbol{s})$, where $\boldsymbol{\theta}$ is a parameter vector in the policy function μ. The expected reward of the deterministic policy is:

$$J(\boldsymbol{\theta}) = \mathbb{E}_s \left\{ \sum_{t=0}^{T} \gamma^{(t)} R(\mu_{\boldsymbol{\theta}}(\boldsymbol{s}(t)), \boldsymbol{s}(t)) \right\}. \qquad (17)$$

We use the gradient of $J(\boldsymbol{\theta})$ to update our deterministic policy $\mu_{\boldsymbol{\theta}}(\boldsymbol{s})$. The goal of deterministic policy gradient algorithms is to find a policy that maximizes the expected return of the algorithm in (17), i.e., find a $\boldsymbol{\theta}$ that maximizes $J(\boldsymbol{\theta})$. We use a DNN (with weights $\boldsymbol{\theta}$) to model the deterministic policy function $\mu_{\boldsymbol{\theta}}(\boldsymbol{s})$. Then, we find $\boldsymbol{\theta}$ using the DDPG algorithm [18]. Ultimately, we obtain a deterministic policy $\mu_{\boldsymbol{\theta}}(\boldsymbol{s})$ which at any given state s gives us the optimal action $\boldsymbol{a}$, i.e., $r_i^d(t)$ for each user $i \in \mathcal{N}$. This action $\boldsymbol{a}$ is mapped to the RB and power allocation matrices using the action space reducer, and hence, this solves our problem.

IV. Simulation Results and Analysis

For our simulations, we consider a square area of size 2 km $\times$ 2 km in which 20 users are served by an OFDMA cellular system. We set the total bandwidth to $B = 45$ MHz and the bandwidth of each RB to 180 kHz. We also set the noise variance to $\sigma^2 = -173.9 \frac{\text{dBm}}{\text{Hz}}$, unless otherwise mentioned. We set the path loss exponent to 3 (urban area), and the carrier frequency to 2 GHz. We set the packet length for our system to 10 kbits, and the maximum BS power to 4 W, and each user's latency $D_i^{\max}$ to 10 ms, unless otherwise mentioned. For evaluation purposes, we assume that the packets arrive according to a Poisson arrival process with the same rate and with the same average packet size for all the users. However, the proposed deep-RL framework does not have any information about this traffic model.

Fig. 2 shows the relation between the rate, delay, and reliability in our system. As we mentioned, the rate, reliability, and latency are incompatible design parameters. However, since our system can attain any feasible combination of the rate, reliability, and latency, we can enable URLLC with reliability of 99% and latency of 4.2 ms with the rate of 1 Mbps, and a reliable high-rate communication with 99% reliability and rate of 10 Mbps with latency of 24.5 ms at the same time. Also, the system can balance between rate, reliability, and latency. As an example, we can see from Fig. 2 that, our system is able to provide ultra-reliable low latency communication with a delay of 8 milliseconds and a reliability of 98% with a rate of 7 Mbps. However, if we need higher rate for this system, without decreasing reliability or increasing latency, then we have to allocate a higher bandwidth to the system. We can increase the rate of each one of the 20 user to 46 Mbps if we increase the system bandwidth from 45 MHz to 200 MHz and the power from 5 W to 20 W. These results provide insightful guidelines for controlling the rate-reliability-delay tradeoff. For example, we see that with a reliability of 98%, delay of 8 ms, and rate of 7 Mbps, a gain of 1% reliability can be done with 47% less delay but at the expense of a seven-fold decrease in the rate.

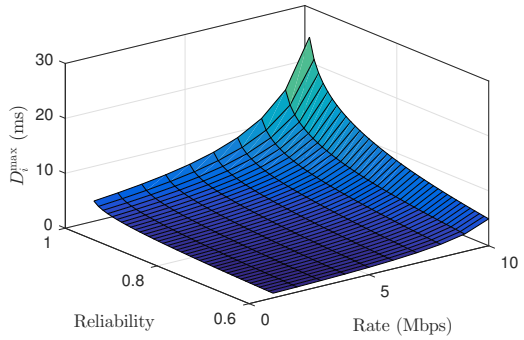

Figure 2: 3D plot of the achievable rate, delay, and reliability for the system.

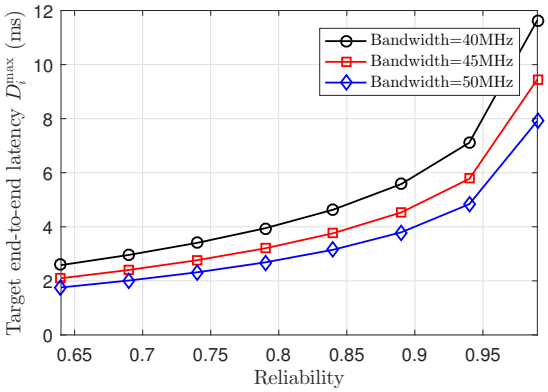

Figure 3: Effect of bandwidth on delay-reliability tradeoff for the designed system.

Fig. 3 shows the effect of the maximum bandwidth on the delay-reliability tradeoffs. We can see that, as we allocate more bandwidth to the system, we are able to achieve higher reliability and lower latency with the same rate. For instance, by increasing the bandwidth from 45 MHz to 50 MHz, we are able to decrease the latency of each user by 16%. Also we can see that increasing the reliability increases the latency in the system and this is because of the reliability latency tradeoff.

The effect of packet size on the reliability of the system with $D_i^{\max} = 10$ ms is shown in Fig. 4. We can see that, for higher rates, the effect of packet size becomes more dominant. The system can only provide more reliability to the traffic with shorter packet sizes. This is due to the fact that applications

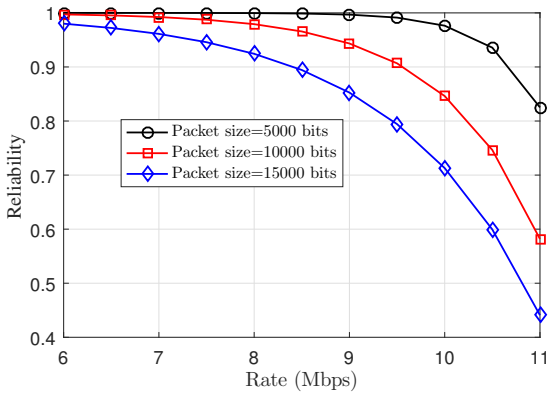

Figure 4: Effect of packet size on rate-reliability tradeoff for the designed system.

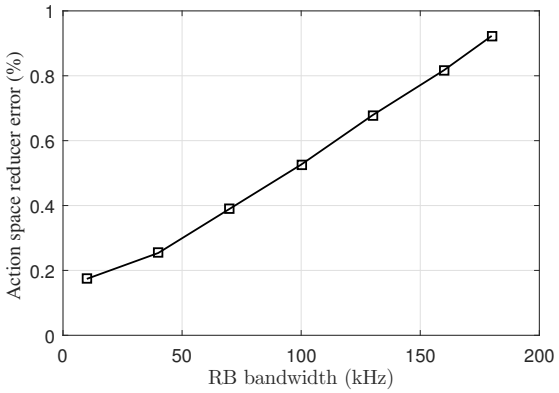

Figure 5: Effect of resource block bandwidth on the per user error of the action reducer.

with shorter packets naturally have a smaller end-to-end delay. Hence, it is less challenging for our system to provide ultra high reliability to such applications. Fig. 4 shows that our system is able to reach URLLC reliability and latency as well as higher data rates with moderate latency and reliability for high data rate and large packet size applications. We can see that at higher data rates the reliability decreases, and this is because the limited bandwidth and power in the system can guarantee reliability up to a certain rate. We can increase this reliable rate by either decreasing packet size, increasing bandwidth, increasing power, and/or increasing target end-to-end latency.

Fig. 5 shows the per user error of the action reducer defined as $E = \frac{\|\boldsymbol{r}-\boldsymbol{r}^d\|}{N\|\boldsymbol{r}\|}$, where $\boldsymbol{r}$ is the vector of wireless downling downlink rate and $\boldsymbol{r}^d$ is the vector of desired rate. This error measures the distance between the input and output rate of action space reducer. We can see that, as the bandwidth of each RB decreases, the number of RBs increases in the system, and hence, the error of our action space reducer will decrease. We can see that, for an RB bandwidth of 180 kHz (typical for LTE), the error is less than 1% for each user.

V. Conclusion

In this paper, we have proposed a a novel deep-RL framework for model-free URLLC in the downlink of an OFDMA system. In particular, we have designed a framework that allocates power and RB to downlink wireless users while guaranteeing their end-to-end reliability and end-to-end latency without any modeling assumptions on traffic or packet sizes. We have used this reliability and latency as a feedback to our deep-RL framework. Then, we have designed an action space reducer to adapt our deep-RL framework to the policy gradient methods and use DDPG algorithm to find the optimal policy. Our results have shown that our system can achieve high reliability, low latency, and high rate without any prior knowledge of users' traffic model.

References

[1] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct 2018.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, Aug. 2016.

[3] A. Rahmati, X. He, I. Guvenc, and H. Dai, "Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks," *arXiv preprint arXiv:1901.02613*, 2019.

[4] M. Mozaffari, A. Taleb Zadeh Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with uavs: Foundations of a 3D wireless cellular network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, Jan 2019.

[5] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *Proc. of IEEE International Conference on Communications (ICC), Kansas City, MO, USA*, July 2018, pp. 1–6.

[6] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," *arXiv:1805.00983*, 2018.

[7] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5621–5635, Nov 2018.

[8] A. Anand, G. D. Veciana, and S. Shakkottai, "Joint scheduling of URLLC and embb traffic in 5G wireless networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM), Honolulu, HI, USA*, April 2018, pp. 1970–1978.

[9] C. Liu and M. Bennis, "Ultra-reliable and low-latency vehicular transmission: An extreme value theory approach," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1292–1295, June 2018.

[10] Z. Hou, C. She, Y. Li, T. Q. S. Quek, and B. Vucetic, "Burstiness-aware bandwidth reservation for ultra-reliable and low-latency communications in tactile internet," vol. 36, no. 11, pp. 2401–2410, Nov 2018.

[11] W. K. Lai and C.-L. Tang, "QoS-aware downlink packet scheduling for lte networks," *Computer Networks*, vol. 57, no. 7, pp. 1689–1698, 2013.

[12] M. Yousefvand and N. B. Mandayam, "Joint user association and resource allocation optimization for ultra reliable low latency hetnets," *arXiv preprint arXiv:1809.06550*, 2018.

[13] Y. He, F. R. Yu, N. Zhao, H. Yin, and A. Boukerche, "Deep reinforcement learning (DRL)-based resource management in software-defined and virtualized vehicular Ad Hoc networks," in *Proc. of ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, Miami, FL, USA*, April 2018, pp. 47–54.

[14] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *Proc. of IEEE International Conference on Communications (ICC), Paris, France*, May 2017, pp. 1–6.

[15] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, Feb 2019.

[16] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain*, April 2018, pp. 1–6.

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

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

[19] A. T. Z. Kasgari and W. Saad, "Stochastic optimization and control framework for 5G network slicing with effective isolation," in *Proc. of 52nd Annual Conference on Information Sciences and Systems (CISS)*, March 2018, pp. 1–6.

[20] A. T. Z. Kasgari, W. Saad, and M. Debbah, "Human-in-the-loop wireless communications: Machine learning and brain-aware resource management," *arXiv preprint arXiv:1804.00209*, 2018.