Delay-aware Cellular Traffic Scheduling with Deep Reinforcement Learning

†Ticao Zhang, ‡Shuyi Shen, †Shiwen Mao, and ‡Gee-Kung Chang
†Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201
‡School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250
Email: tzz0031@tigermail.auburn.edu, ssyzoe@gatech.edu, smao@ieee.org, gkchang@ece.gatech.edu

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322560

***Abstract*—Radio access network (RAN) in 5G is expected to satisfy the stringent delay requirements of a variety of applications. The packet scheduler plays an important role by allocating spectrum resources to user equipments (UEs) at each transmit time interval (TTI). In this paper, we show that optimal scheduling is a challenging combinatorial optimization problem, which is hard to solve within the channel coherence time with conventional optimization methods. Rule-based scheduling methods, on the other hand, are hard to adapt to the time-varying wireless channel conditions and various data request patterns of UEs. Recently, integrating artificial intelligence (AI) into wireless networks has drawn great interest from both academia and industry. In this paper, we incorporate deep reinforcement learning (DRL) into the design of cellular packet scheduling. A delay-aware cell traffic scheduling algorithm is developed to map the observed system state to scheduling decision. Due to the huge state space, a recurrent neural network (RNN) is utilized to approximate the optimal action-policy function. Different from conventional rule-based scheduling methods, the proposed scheme can learn from the interactions with the environment and adaptively choosing the best scheduling decision at each TTI. Simulation results show that the DRL-based packet scheduling can achieve the lowest average delay compared with several conventional approaches. Meanwhile, the UEs' average queue lengths can also be significantly reduced. The developed method also exhibits great potential in real-time scheduling in delay-sensitive scenarios.**

***Index Terms*—Delay; Deep reinforcement learning (DRL); Packet scheduling; Recurrent neural network (RNN).**

I. Introduction

The envisioned applications of the 5G communication technologies, such as video streaming, virtual reality, and 360 degree videos, are all imposing stringent delay requirements [1]. Radio access network (RAN) has become a key enabling technology to address this issue. In RAN, radio resource management (RRM) [2], [3] plays a vital role in allocating the resources to user equipments (UEs) by jointly exploiting the advanced MAC layer functions, such as resource sharing, link adaptation, hybrid automatic retransmission request (HARQ), and channel quality indicator (CQI) reporting.

In conventional RAN, a packet scheduler is deployed at the base station (BS) and it is responsible for allocating the wireless spectrum resources to UEs based on their quality of service (QoS) requirements as well as their reported channel conditions. In each Transmission Time Interval (TTI), the packet scheduler needs to solve a decision-making problem to decide how the resources are allocated to the UEs. Due to the large number of UEs and the dynamic wireless environment, it is difficult to obtain an optimal solution with the existing optimization methods within the channel coherence time. As a result, most of the existing approaches rely on some predefined rules. For example, conventional packet scheduling strategies, such as the max-CQI scheduling approach and proportional fairness (PF) scheduling, are all rule-based scheduling policies which simply assign the resource block (RB) to the UEs' that have the best channel condition or the best relative channel condition. The problem is these methods are designed based on human's understanding of a network. Although simple and useful, it is hard to guarantee they are optimal and accurate.

In todays' highly complex wireless environment, variety of applications have emerged, for which rule-based scheduling methods have their limitations. For example, from a spectrum efficiency point view, max-CQI scheduling, which allocate radio resources to the UEs that is expected to have the best achieved rate, would be optimal. However, this is at a cost of sacrificing other important performance metrics, such as fairness, energy-saving, and complexity. In some delay-sensitive scenarios such as vehicle-to-vehicle (V2V) communications [4], ensuring timely delivery of packets is more important than bringing an additional increase to the system throughput. As UEs' QoS requirements may change, developing an intelligent scheduling method that can adaptively select the best scheduling policy is of vital importance.

Recently, deep reinforcement learning has made breakthroughs the field of networking and communications [5]–[10]. In time-varying and unpredictable networks, DRL has proved to be effective in tackling real-time decision-making problems. For example, the authors in [11] developed a decentralized resource allocation mechanism for V2V communications based on DRL. The vehicles can make decisions to find the optimal sub-channel allocation as well as the power level for transmission without global information. Vehicles can learn to satisfy their QoS requirement while also minimizing the interference to other vehicles. In [12], an online DRL-based algorithm is developed to optimally adapt task offloading decisions and wireless resource allocations to the channel conditions. Instead of solving the large combinatorial problem directly, the proposed agent learns the binary decisions from past experience. This method can achieve near optimal performance while significantly decreasing the computational time. In [6], the joint beam forming, power control, and interference coordination problem in a downlink multiple access OFDM cellular network is formulated as a combinatorial problem. The authors show that closed-form expression does not exist and

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

finding the optimal solution requires an exhaustive search. The developed method leverages the power of DRL to avoid the exhaustive search and achieves a near-optimal performance. In our recent works, we have applied DRL to solve the resource allocation problem at a wireless backhaul [9] and to develop a smart congestion control scheme [10].

In the field of cellular traffic scheduling, Ref. [13] investigates how DRL can help solve scheduling problems in cellular networks. It shows that by exploiting the expert knowledge of existing rule-based scheduling method in the training process of the DRL agent, the DRL's learning ability can get improved. In [14], a DRL-packet scheduler is developed to adapt to the dynamic scheduling conditions. Instead of using a single scheduling rule across the entire transmission, Ref. [14] proposes a framework to dynamically select the best scheduling rule at each TTI based on UEs' QoS requirements. Simulations show that the developed method outperforms a conventional scheduling method in terms of delay and package drop rate requirements.

It is envisioned that the application of AI/ML techniques to the design of 6G systems will be fundamental to improve the system performance [15]. In this paper, we incorporate DRL to address the problem of delay-aware packet scheduling in the downlink of a cellular network. We show that delay-aware packet scheduling is a complex combinatorial problem, which is challenging to solve. By modeling the packet scheduling problem as a Markov decision process (MDP) problem, a deep Q-learning agent that is based on a recurrent neural network (RNN) is designed to learn a delay optimized scheduling solution through the interactions with the environment. Simulation results show that the DRL-based packet scheduler, designed to minimize the queueing delay of all UEs, outperforms several existing scheduling schemes. Meanwhile, the framework in this paper can be easily generalized to other scenarios with various QoS requirements.

The paper is organized as follows. In Section II, the system model is introduced and the delay-aware scheduling problem is formulated. In Section III, a DRL-based packet scheduling algorithm is proposed. Simulation results are presented in Section IV to validate the superiority of the proposed method. Finally, Section V concludes this paper.

II. System Model and Problem Statement

A. System Model

We consider the downlink transmissions of a cellular network, where UEs are served by a base station (BS). As shown in Fig. 1, multiple UEs request packets from the BS and the BS performs traffic scheduling with UE-specific queues. We consider an orthogonal frequency division multiplexing (OFDM) system, where the available wireless resources in time and frequency are divided into resource blocks (RBs). The RB scheduling is performed at each time slot (i.e., TTI).

Let $\mathcal{U} = \{1, 2, ..., U\}$ denote the set of active UEs and $\mathcal{B} = \{1, 2, ..., B\}$ the set of RBs, where U and B are the total number of UEs and RBs, respectively. We aim to develop a packet scheduler to allocate the set of RBs to the UEs so that the average delay of the UEs can be minimized.

At a time slot t, we denote the maximum number of bits that could be sent for an RB $b \in \mathcal{B}$ to a UE $u \in \mathcal{U}$ as $C_{ub}[t]$.

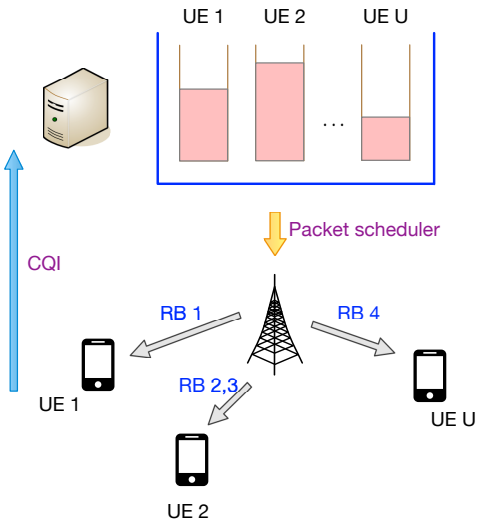

Fig. 1: Delay-aware cellular downlink traffic scheduling system model.

According to [16], the value of $C_{ub}[t]$ depends on the channel quality indicator (CQI) reported by UE u. Based on the CQI, a proper modulation and coding scheme (MCS) is assigned to the allocated RB. In practice, there exists a mapping table between $C_{ub}[t]$ and the MCS.

B. Traffic Model

We consider the case where the UEs are characterized by data request pattern. Suppose that at time slot T, the requested data size of UE u is $A_u[t]$. The BS will assign RBs to UEs and transmit the requested data to the corresponding UEs. However, the wireless resources are limited. Not all the UEs can get allocated RBs immediately. The BS maintains a separate queue for each UE. The requested packets of the UE will be queued in the buffer. At time slot t, the queue length of UE u is denoted as $Q_u[t]$. The buffer state of a UE can be written as

$$Q_u[t] = \max(\min(Q_u[t-1] + A_u[t], Q_{\max}) - D_u[t], 0), \quad (1)$$

where $Q_{\max}$ is the maximum buffer size of all UEs, $D_u[t]$ is the scheduled, transmitted data for UE u in time slot t, which can be computed as

$$D_u[t] = \sum_{b=1}^{B} x_{ub}[t] \cdot C_{ub}[t], \quad (2)$$

where $x_{ub}[t]$ is the RB allocation indicator: $x_{ub}[t] = 1$ if UE u is assigned with RB b at time slot t; and $x_{ub}[t] = 0$ otherwise.

C. Traffic Delay

Packets of UEs are time stamped and queued in the buffers for transmission based on the first-in-first-out principle. For each packet, the difference between the current time and the arrival time, which we call the head of line (HoL) time, is used to measure the packet delay. At time slot t, the HoL packet

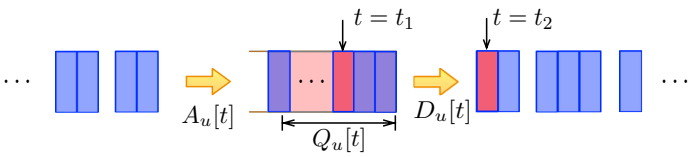

Fig. 2: The traffic model considered in this paper.

delay of UE u is denoted by $d_u[t]$. During a time period T, the average delay of UE u can be computed as

$$W_u = \frac{1}{T}\sum_{t=1}^{T} d_u[t] \tag{3}$$

As shown in Fig. 2, at a time t, the HoL time $d_u[t]$ can be computed by definition $d_u[t] = t_2 - t_1$, where t_1 is the time when the packet first enters the queue and t_2 is the time when the packet leaves the queue.

D. Problem Formulation

Our objective is to jointly optimize the RB allocation in the wireless links so that UE's average delay can be minimized. We formulate the problem as follows.

$$\min_{x_{ub}[t]} \quad \Gamma_{\text{Delay}} = \frac{1}{U}\sum_{u=1}^{U} W_u \tag{4}$$

$$\text{s.t.} \quad x_{ub}[t] = \{0,1\},\ \forall u, b, t \tag{5}$$

$$\sum_{u}^{U} x_{ub}[t] \leq 1,\ \forall b, \tag{6}$$

where constraint (5) means that the RB assignment variables are binary, and constraint (6) ensures that each RB can only be assigned to one UE. To solve Problem (4) is to find the best RB scheduling policy at each time slot for all UEs and RBs. This problem is difficult for the following reasons: (i) Constraints (5) and (6) makes the problem combinatorial; (ii) The objective function does not have a closed-form expression in terms of the scheduling policy $x_{ub}[t]$. A direct optimization may become hard; (iii) the number of RBs and the number of UEs may be very large, which makes the optimization problem more challenging. In some delay-sensitive applications such as vehicle communications, a fast and efficient algorithm is needed [4], [11].

III. DRL-based Packet Scheduling

A direct optimization of problem (4) is hard. In this section, we show that problem (4) exhibits Markov decision process (MDP) property, based on which, a deep reinforcement learning (DRL) based packet scheduling policy is proposed.

A. MDP Problem

As in (1), the queue length of UE u at time slot t depends on the queue length at time slot $t-1$, the requested data $A_u[t]$, and the scheduled traffic $D_u[t]$. The BS observes the queue length of each UE. Based on the requested packet size of different UEs and their corresponding channel conditions, the BS allocates RBs to the UEs so that their averaged delay can be minimized. Mathematically, at time slot t, the decision of

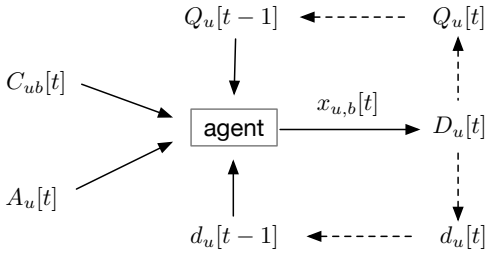

Fig. 3: A Markov decision process (MDP) problem.

computing the current scheduled traffic for UE u is a function of the current channel condition (i.e., the maximum amount of traffic carried in an RB), the previous queue status, the current HoL delay, and the current request data size of all UEs, i.e.,

$$D_u[t] = f(C_{ub}[t], Q_u[t-1], d_u[t], A_u[t]). \tag{7}$$

In addition, as analyzed in Section II, the current HoL packet delay depends on the queue status as well as the history data request rates $A_u[t]$ and data traffic $D_u[t]$, $t = 1, 2, ...$ Therefore, the buffer state (1) can be modeled as an MDP. MDP is a discrete time stochastic control process. It provides a mathematical framework for modeling the decision making where the outcomes are partly random and partly depending on the policy that is made by the controller. As shown in Fig. 3, the packet scheduler, which we call *agent*, observes the current state $\{C_{ub}[t], Q_u[t-1], d_u[t], A_u[t]\}$, and then makes decision on how each RBs are assigned to UEs. Then the traffic is scheduled, which in turn updates the queue length as well as the HoL packet delay for each UE. At the next time slot, the agent makes decision again, and so forth. The ultimate goal is to find a stable "policy" so that the averaged HoL delay can be minimized.

B. Deep Reinforcement Learning (DRL)

Reinforcement learning (RL) is an important tool to solve the MDP problem. In RL, the agents learns from the interactions with the environment. Q-learning algorithm is the most widely used method in RL. In Q-learning, the set of possible states is denoted as $\mathcal{S}$, and the set of discrete actions is denoted as $\mathcal{A}$. At each time instant t, the agent takes action $a^t \in \mathcal{A}$ when observing the state $s^t \in \mathcal{S}$ and receives a reward r^t. Then the system enters the next state s^{t+1}. The Q-learning algorithm aims to learn an optimal policy π which maps state s^t to action a^t, so that the reward over time can be maximized.

For example, we can define the reward function as

$$R^t = \sum_{\tau=0}^{\infty} \gamma^\tau \cdot r^{t+\tau}, \tag{8}$$

where $\gamma \in (0, 1]$ is a tradeoff scalar between the immediate and future rewards. Under a policy π, the *Q-function* of the agent is defined as

$$Q_\pi(s, a) = \mathbb{E}_\pi[R^t | s^t = s, a^t = a]. \tag{9}$$

Q-learning aims to maximize the Q-function by maintaining a *Q-table*. However, when the state and action spaces become continuous and large, the problem becomes intractable. To address this problem, DRL uses a deep neural network (DNN)

to approximate the mapping table. DRL inherits the advantages of both RL and deep learning, and is more efficient than RL.

C. DRL-based Delay-aware Packet Scheduler

We next present the DRL-based RB scheduling algorithm. At each time step t, the agent performs a certain action a_t based on the current state s_t. The agent receives a reward and moves to the next state.

The action is defined as the choice of $x_{ub}[t]$. Note that we have U UEs and B RBs in total; so the dimension of the action space at each time slot would be $\mathcal{O}(2^{U\times B})$. We map the choice of $x_{ub}[t]$ to a real integer number that is between $[0, U^B - 1]$. Each integer number in the interval corresponds to a unique RB allocation to the UEs. We use the interval to denote the action space $\mathcal{A}$, i.e.,

$$\mathcal{A} = [0, U^B - 1]. \tag{10}$$

In practical systems, the number of RBs is huge. A direct use of RB would bring an extremely large action space. In OFDM, frequency selective wireless channels are transferred into multiple flat channels over different sub-bands. consecutive sub-bands can be grouped together, which is called resource block group (RBG) in LTE [3]. This way, the action space can be greatly reduced and the algorithm can converge faster.

The state space include UE's packet arrival rate, buffer state, the transmission rate of different RBs, which is denoted by

$$\mathcal{S}[t] = \{S_1[t], S_2[t], ..., S_U[t]\}, \tag{11}$$

where $S_u[t]$ is the observed state of UE u, given by

$$\mathcal{S}_u[t] = \{A_u[t], Q_u[t-1], d_u[t-1], C_{ub}[t]\}. \tag{12}$$

As a result, the dimension of the state space is $UB + 3U$.

What makes DRL appealing is we can customize the reward function for a specific problem. In this paper, we directly use the negative value of the objective function (4) as the reward function. Maximizing the reward function will be equivalent to minimizing the average delay of all UEs. Our developed method can be quite general and flexible, as the reward function can be extended by jointly consider the effect of throughput, QoS requirements, latency, and priority. We leave this topic for our future investigation.

We propose Algorithm 1 to solve Problem (4), which is a DRL-based approach. The algorithm perform RB scheduling so that the UEs' average delay can be minimized. An RNN is incorporated, which takes state as input and outputs a Q-value for each of the candidate actions. The main steps include

- Select an action at time slot t
- Reward the action based on the computed time delay
- Train the RNN based on the outcome

We call the period of time in which an interaction between the agent and the environment takes place as an *episode*. In the beginning of each episode, the environment initializes the state and then the agent interacts with the environment for several training steps (or TTIs) during this episode. We perform several episodes until the accumulated reward converges.

Algorithm 1 Delay-optimal cellular traffic scheduling algorithm with DRL

if training **then**
  Start environment simulator, generating arrival traffic and the status of the RBs ;
  Initialize the time, states, action and replay buffer $\mathcal{D}$;
  for each episode **do**
    for each time slot t **do**
      Observe state $S[t]$ as in (11) ;
      $\epsilon = \max(\epsilon \cdot d, \epsilon_{\min})$;
      Sample $r \sim \mathcal{N}(0, 1)$;
      if $r \leq \epsilon$ **then**
        Select an action $A[t] \in \mathcal{A}$ randomly ;
      else
        Select an action such that $A[t] = \arg\max_{a'} Q_\pi(S[t], a'; \theta[t])$;
      end if
      Compute reward as $r[t] = -\frac{1}{U}\sum_{u=1}^{U} W_u$;
      Observe the next state S' ;
      Store the experience $(S[t], A[t], S', r[t])$ in $\mathcal{D}$;
      Minibatch sample from $\mathcal{D}$, $e_j \triangleq (S_j, A_j, r_j, S'_j)$;
      Set $y_j := r_j + \gamma \cdot \max_{a'} Q_\pi(s_{j+1}, a'; \theta[t])$;
      Perform gradient optimization method on $(y_j - Q_\pi(s_{j+1}, a'; \theta[t]))$ and obtain the optimal θ^* ;
      $\theta_t = \theta^*$;
      $t = t + 1$;
      $S[t] = S'$;
    end for
  end for
  Save θ_t and the agent ;
else
  Load the agent ;
  Based on the observed state, output the action ;
end if

IV. Simulation Results and Discussions

A. Parameter Setting

We consider the case where two UEs request data from the BS. The data request pattern of UE 1 follows the Poisson process with parameter $\lambda = 8$, i.e.,

$$P(A_1[t] = k) = \frac{e^{-\lambda}\lambda^k}{k!},\ k = 0, 1, 2, ... \tag{13}$$

The data request pattern of UE 2 follows the uniform distribution, where

$$P(A_2[t] = k) = \frac{1}{5},\ \forall k \in \{6, 7, 8, 9, 10\}. \tag{14}$$

The maximum queue length $Q_{\max}$ is set as 100. We assume that the quality of each RB can be measured at each TTI, based on which the maximum number of bits that can be transmitted on the RB can be computed. In this simulation, we assume that the maximum number of bits that can be transmitted on the RB for a specific UE can only take discrete values from a set $\{2, 3, 4, 5\}$ uniformly. Although we use discrete values here, it is worth noting that the proposed algorithm also applies to continuous value settings and various distributions of the UE data requests.

We create a DQN agent with RNN. The structure of the critic network is shown in Fig. 4. There are three layers

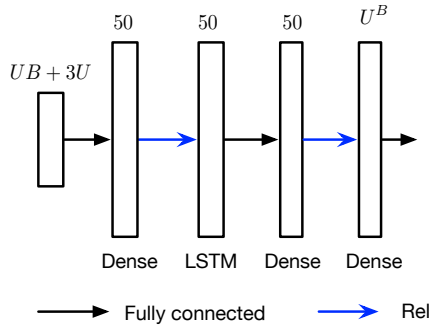

Fig. 4: The recurrent neural network used in the agent.

between the input and the output: two dense layers and one long-short-term memory (LSTM) layer. Each layer has 50 neurons. The training parameters are listed in Table I. The number of episodes is 200. In each episode, the number of steps is 300. In the offline training phase, the value of the requested data size $A_u[t]$ and the channel capacity $C_{ub}[t]$ are generated according to a known distribution in each step. In the testing phase, we generate 1000 simulations and average the results.

B. Benchmark Algorithms

1) Round-robin: The round robin allocation strategy simply allocates all RBs to each user in turn. It is one of the simplest algorithms. Regardless of the channel condition and the traffic requirement, Round robin assumes all the UEs have equal priority, while different UEs may have different channel quality on the same radio resource. No optimization is performed. It is evident that round-robin scheduling will incur a poor performance.

2) Max-CQI: The Max-CQI scheduler only considers the channel condition while allocating RBs to the UEs. The Max CQI aims to maximize the system's capacity by allocating RBs to the UEs that have the best channel condition, i.e., at each time slot t, $x_{u^*,b}[t] = 1$ where

$$u^* = \arg\max_u \{C_{ub}[t]\}. \tag{15}$$

The max-CQI packet scheduler can maximize the transmitted traffic since the RBs are allocated greedily to the UEs that can achieve the highest transmission rate. However, the UEs with low channel quality would have little chance to get the transmission resources. As a result, the average delay would be bad. Extremely, a UE may never get RB allocated.

3) Proportional Fairness (PF): Proportional fairness aims to maximize network capacity while also ensure fairness. The idea is to balance the average past throughput and the expected rate. In our simulations, we choose $x_{u^*,b}[t] = 1$ where

$$u^* = \arg\max_u \left\{ \frac{C_{ub}[t]}{Y}_u[t-1] \right\}, \tag{16}$$

where $Y_u[t]$ is the average scheduled traffic of UE u in period $[0, t-1]$ and $Y_u[t] = \frac{1}{t}\sum_{i=1}^{t} D_u[i]$. It is easy that if a UE does not get RBs for a long time, the value of $Y_u[t]$ will be low. Therefore, there will be a high chance that the UE gets RB allocated in the next time slot. Some prior works use a time window to average the allocated traffic.

TABLE I: The DRL Hyperparameters

Parameter	*Value*
Number of episodes	200
Number of steps per episode	300
Discount factor	0.99
Experience replay length	1000,000
ϵ decay d	0.9999
Initial exploration rate ϵ	1
Maximum batch-training trajectory (RNN)	20

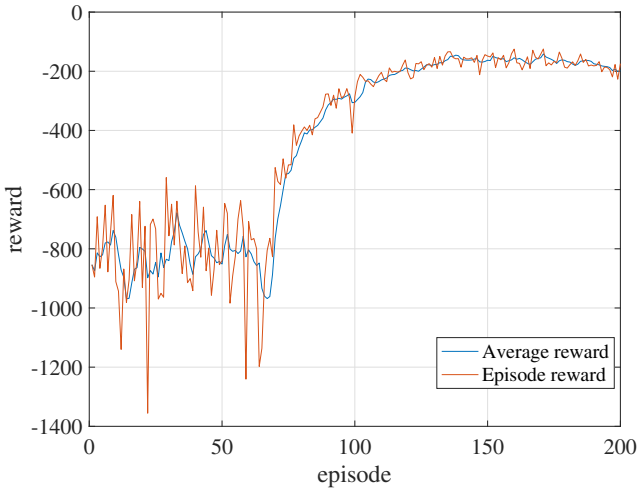

Fig. 5: Training reward vs. episode for the proposed algorithm.

C. Performance Metrics

The aim of this paper is to minimize the packet delay for all UEs. Therefore, the performance metric will be the averaged delay, which is calculated as in (4). Meanwhile, we also compare the scheduled traffic (i.e., throughput) of the UEs, which is computed as

$$\Gamma_{\text{throughput}} = \frac{1}{UT}\sum_{u=1}^{U}\sum_{t=1}^{T} D_u[t], \tag{17}$$

and the average queue length, which is defined as

$$\Gamma_{\text{QueueLength}} = \frac{1}{UT}\sum_{u=1}^{U}\sum_{t=1}^{T} Q_u[t]. \tag{18}$$

D. Results and Discussions

1) Convergence: The convergence plot of the training process is presented in Fig. 5. Both the average result of all the episode as well as the instant per episode reward are plotted. It can be seen that after around 100-episode training, the reward value begins to converge. Furthermore, the maximum reward is attained at around the training episode 150. The optimal number of training episodes can be set to be around 150 to achieve the best performance.

2) Delay: The delay performance comparison is shown in Fig. 6(a). The proposed DRL-based scheduling algorithm achieves the lowest delay, which is 0.4049s. Max-CQI, which aims to maximize the traffic throughput, has the largest delay, which is nearly 3.5 times larger than that of the proposed algorithm. By taking fairness into consideration, PF avoids

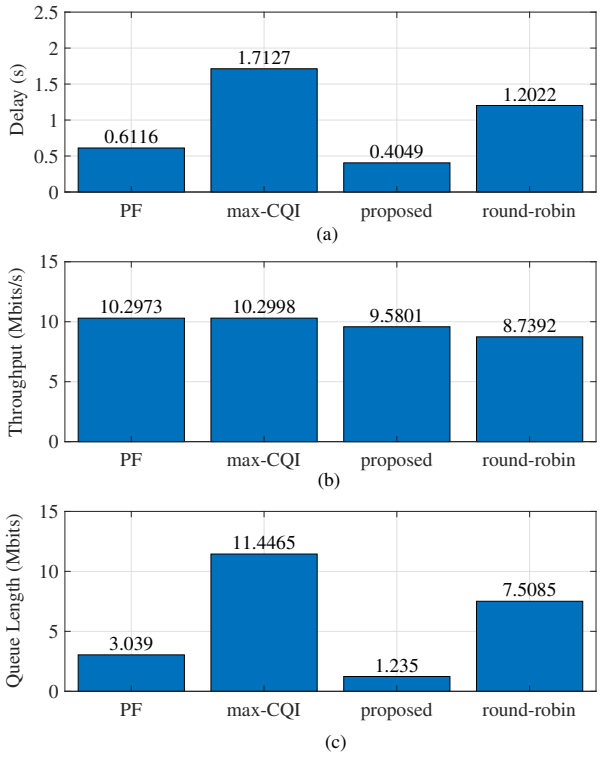

Fig. 6: Performance comparison of four scheduling algorithms.

the situation where UEs with bad channel condition do not get RB allocated. Therefore, the average queue delay can be reduced. Despite that, the proposed DRL scheduling algorithm achieves the best performance in terms of packet delay.

3) Throughput: The achieved throughput comparison is shown in Fig. 6(b). This performance metric is defined as the sum of the transmitted packets over a time period and reflects the system throughput. As analyzed before, max-CQI achieves the highest system throughput. The performance of PF is quite close to that of max-CQI. The proposed method also achieves a quite high throughput. Round-robin has the poorest throughput performance.

4) Queue Length: The average queue length is also an essential metric when we compare the delay performance. In addition, when the queue length is large, there is a high chance that the packet may be dropped, which will bring a high packet drop rate (PDR). As a result, the UE will request the packet again and the delay performance would be even worse. The average queue length is depicted in Fig. 6(c). Our proposed method has the lowest average queue length. As a comparison, the max-CQI method has the longest queue length, which is almost 9 times longer than our proposed method. Round-robin, which serves the UEs in turn at different time slots, can ensure that each UE get served "equally." However, it does not consider the channel condition diversity for different UEs and does not fully exploit the resources. As a result, the queue length performance of round-robin is also poor.

V. Conclusions

In this paper, we aimed to minimize the average packet delay of a downlink multi-access OFDM cellular network, where the UEs have different packet request patterns and channel conditions. We developed a DRL-based packet scheduling method. We showed that our proposed method can achieve the lowest delay and the shortest average queue length. The proposed method has a great potential for delay-sensitive applications, such as vehicle-to-vehicle (V2V) communication in the 5G era. Moreover, the reward function can be customized by jointly considering the dynamic traffic load, QoS parameters, and application requirements.

Acknowledgments

This work is supported in part by the NSF under Grant CNS-1822055 and CNS-1821819.

References

[1] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46–53, June 2013.

[2] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management toward 5G radio access networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1656–1686, Third Quarter 2016.

[3] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 678–700, Second Quater 2012.

[4] X. Wang, S. Mao, and M. Gong, "An overview of 3GPP cellular vehicle-to-everything standards," *ACM GetMobile: Mobile Computing and Communications Review*, vol. 21, no. 3, pp. 19–25, Sept. 2017.

[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2016.

[6] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, Mar. 2019.

[7] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, June 2019.

[8] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key technologies and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, Fourth Quarter 2019.

[9] M. Feng and S. Mao, "Dealing with limited backhaul capacity in millimeter wave systems: A deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 50–55, Mar. 2019.

[10] K. Xiao, S. Mao, and J. Tugnait, "TCP-Drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access J.*, vol. 7, no. 1, pp. 11 892–11 904, Jan. 2019.

[11] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[12] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, in press.

[13] J. Wang, C. Xu, Y. Huangfu, R. Li, Y. Ge, and J. Wang, "Deep reinforcement learning for scheduling in cellular networks," in *Proc. WCSP'19*, Xi'an, China, Oct. 2019, pp. 1–6.

[14] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, Y. Lu, R. Trestian, and G. Ghinea, "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1661–1675, Dec. 2018.

[15] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access J.*, vol. 8, no. 1, pp. 57 063–57 074, Mar. 2020.

[16] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2010.