DRL-Based Channel and Latency Aware Radio Resource Allocation for 5G Service-Oriented RoF-mmWave RAN

Shuyi Shen, Ticao Zhang, Shiwen Mao, *Fellow, IEEE*, and Gee-Kung Chang, *Fellow, IEEE*, *Fellow, OSA*

***Abstract*—A channel and latency aware radio resource allocation algorithm based on deep reinforcement learning (DRL) is proposed and evaluated. The proposed scheme aims to optimize the uplink scheduling for service-oriented multi-user millimeter wave (mmWave) radio access networks (RAN) in the 5G era. In the DRL system, multiple application flows are implemented with various statistical models and the key function modules of the system are designed to reflect the operation and requirements of service-oriented RANs. In particular, the mmWave channel characteristics utilized in the system are collected experimentally and verified via a radio-over-fiber (RoF)-mmWave testbed with dynamic channel variations. Results show that the proposed DRL algorithm can operate adaptively to channel variations and achieve at least 12% average reward improvement compared to conventional single-rule schemes, providing joint improvement of bit error rate and latency performance.**

***Index Terms*—Deep reinforcement learning, radio resource allocation, scheduling, millimeterwave.**

I. Introduction

RADIO access networks (RAN) in the 5G New Radio and beyond are envisioned to be service-oriented, supporting multiple users and various applications with different quality-of-service (QoS) requirements [1]. In addition to capacity and speed requirements, latency becomes an important performance benchmark, especially for time-sensitive data traffic. Applications such as video streaming, low-latency gaming, and real-time services including robotic control, intelligent factories, telehealth will have different delay and reliability requirements [2]. As a result, for simple pre-scheduling or fixed radio resource allocation schemes used in legacy wireless communication networks, it will be challenging to manage the increased QoS complexity while providing operational flexibility and efficiency.

Currently, millimeter wave (mmWave) links are implemented for 5G RANs, which can result in dynamic channel conditions that add to the complexity of radio resource management (RRM) [3]. Although mmWave can provide wide bandwidth and high capacity, it is subject to less diffraction in beam propagation, high Friss path propagation loss and atmospheric absorption loss. For example, mmWave operating in the frequency range of 24.25 to 52.6 GHz is standardized as Frequency Range 2 (FR2) by 3GPP Release 15 [4]. In outdoor environments, such mmWave links can experience abrupt signal strength variations due to raindrops, moving pedestrians, or vehicles [5]. Whereas inside a smart factory, mmWave links are susceptible to line-of-sight (LoS) blockages caused by moving robots or stock boxes. Considering both complex QoS objectives and dynamic channel conditions, the needs of agile and adaptive radio resource scheduling and allocation are urgently anticipated in 5G and beyond RANs.

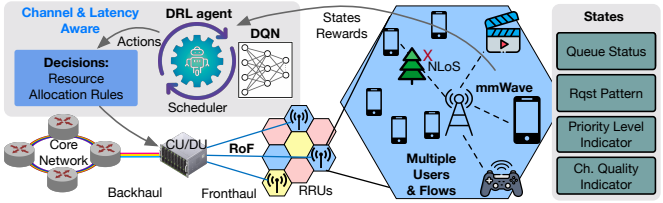

Fig. 1: System architecture design in 5G environment. (Rqst: Request; Ch: Channel.)

To tackle the challenges, research works have been reported to develop intelligent radio resource allocation and scheduling. In [6], deep reinforcement learning (DRL) is utilized to optimize resource block (RB) allocation in a mmWave mobile backhaul. In the work, capacity is the optimization objective and the DRL action is the direct RB allocation and user mapping, which can be extremely complicated if the RB space scales up. In [7], Markov decision process (MDP) is used to model the operations of a mobile edge computing (MEC) system. Considering random task arrivals and channel state variations, the method can optimize power consumption while meeting the latency requirements. However, only a single user is considered in the work, which is not sufficient because multi-user contention and management are required to solve the scheduling problem. In [8], the authors implement deep deterministic policy gradient (DDPG) for radio resource scheduling in a 5G RAN, taking multiple users, varied channel conditions, and random traffic arrivals into account. Bit error rate (BER) and delay are jointly considered. The limitation of the work is that only *Poisson* distribution is used to model the arrival patterns of user equipment (UE), despite the diverse application arrival patterns in reality.

In this paper, we utilize DRL to achieve both delay and

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. This work is supported in part by the NSF under Grant CNS-1822055 and CNS-1821819. (Corresponding author: Shuyi Shen.)

Shuyi Shen and Gee-Kung Chang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308 USA (e-mail: ssyzoe@gatech.edu; geekung.chang@ece.gatech.edu).

Ticao Zhang and Shiwen Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: tzz0031@auburn.edu; smao@ieee.org).

channel condition aware packet scheduling and radio resource allocation in the uplinks of a service-oriented mmWave RAN. In DRL, an agent interacts with the environment and aims to optimize the decision-making process. This fits well with the requirements of a scheduling process, which makes prioritization and resource allocation decisions based on the request patterns and channel conditions. The schematic diagram of the system is depicted in Fig. 1. The system will consider multi-user multi-service scenarios with different QoS requirements, to jointly optimize BER and latency. Furthermore, the system takes channel variations into account by varying mmWave link conditions including LoS and none-LoS (NLoS) blockages. Channel characteristics in this work are experimentally collected via a radio-over-fiber (RoF)-mmWave testbed and then implemented in the DRL system. Based on the provided state information which includes service queue status, application request patterns and priority levels, as well as channel quality indicators, the DRL-based scheduler will take the decision action to choose the optimal scheduling and resource allocation rule.

The main contributions of the paper are summarized as follows:

1) We establish a DRL framework for joint BER and latency optimization for time-sensitive traffic in a service-oriented 5G system subject to mmWave channel variations. Different statistical models are implemented for the arrival intervals and packet sizes of diverse applications. Conventional request-grant cycles of the uplink scheduling process are implemented, taking into account possible congestion and queuing delay under heavy traffic load. In addition, we design and formulate the state and reward of the DRL scheduler such that it will reflect queue status, channel variations, and service-customized latency performance based on QoS requirements.

2) In our previous work [9], direct RB allocation mapping to UE is implemented as DRL actions. Through our investigation, we find that such straightforward action design can cause extreme complexity and require huge computational resources if used in a multi-user wide-band mmWave RAN. Therefore, re-design of the action is required to improve the convergence efficiency. The action of the proposed DRL-based scheduler is to select the optimal resource allocation rule regarding the current transmission time interval (TTI). A similar scheme is also adopted in [8].

3) In contrast to most of the previous DRL-related works with only simulation results, the mmWave channel characteristics utilized in the proposed system are experimentally collected and verified via a mmWave testbed with RoF-enabled mobile fronthaul. In this work, photonic-assisted mmWave generation is implemented to achieve wide-bandwidth transmission and experimentally verified channel variations. To realize the channel conditions of mmWave links such as reflection, blockage, and reduced transmission power, channel variation is introduced in the scheduling process.

The paper is an extension of our recent work published in [10], with expanded research results validated by comprehensive system design, theoretical analysis, and experimental demonstration. The remainder of this paper is organized as follows. Section II introduces the framework and design of the DRL algorithm with the illustration of the scheduling process. The system architecture is illustrated in Section III, with implementation details of the component modules. In particular, the mmWave channel demonstration is presented. The evaluation of the DRL system and results are analyzed and discussed in Section IV, which covers the DRL training process and the performance comparison with conventional schemes. Finally, the conclusions are summarized in Section V.

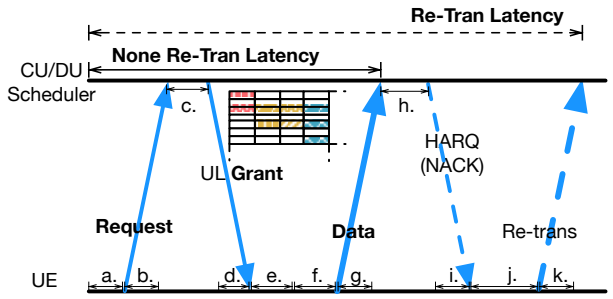

Fig. 2: Uplink scheduling: the request-grant cycle.

II. Scheduling Process and DRL System Design

We consider the uplink transmission of a mmWave remote radio unit (RRU) supported by RoF mobile fronthaul as shown in Fig. 1. The system is flow-oriented and involves multiple UEs that are using applications with different QoS requirements and experiencing different channel conditions. One UE can have multiple active services/flows simultaneously.

The scheduling process follows the request-grant cycles that are widely implemented in mobile communication networks, which is depicted in Fig. 2. During the scheduling process, at each TTI, UEs will firstly request transmission opportunities before actual data transmission. The scheduler located in the central unit (CU) or distributed unit (DU) will process the requests and then distribute the uplink (UL) grants. Not all requests can be satisfied especially when the traffic load is heavy, which may cost additional queuing delay, as indicated by Fig. 2 at point f. The UEs will then prepare and send the data packets using the allocated RBs. In our system, upon receiving the uplink data the scheduler will check the pre-forward-error-correction (pre-FEC) BER of the received data which determines whether re-transmission (Re-Tran) is required as illustrated in Fig. 2. In a real system, the scheduler may send hybrid automatic repeat request (HARQ) or NACK accordingly. For simplicity, the queuing delay is considered only for the original data transmission, while not for the re-transmission, i.e., guaranteed resources for re-transmissions are assumed in the system.

Let $\mathcal{U} = \{1, 2, \ldots, U\}$ denote the set of UEs and $\mathcal{F} = \{1, 2, \ldots, F\}$ denote the set of flows. F is the total number of flows and U is the total number of UEs. One UE can have multiple active flows. If a flow $f \in \mathcal{F}$ belongs to an UE $u \in \mathcal{U}$, then $v(f) = u$, indicating the corresponding UE u of flow f. $\mathcal{B} = \{1, 2, \ldots, B\}$ is the set of resource groups (RG) for allocation. RG is grouped RBs sharing the same modulation order, the design of which will be explained in Section IV. The total number of RGs is B. At TTI t, the capacity of the RG $b \in$

$\mathcal{B}$ corresponding to flow f is $C_{f,b}(t)$, as different UEs can have different channel conditions. Actually, $C_{f,b}(t)$ is determined by $C_{u,b}(t)$ given the flow to UE mapping. Similarly, $E_{f,b}(t)$ denotes the BER of the RG b corresponding to f, which will be calculated from the experimentally measured error vector magnitude (EVM).

In the flow-oriented system, different flows will have different packet sizes and arrival intervals. At TTI t, the requested data size of flow f is $Y_f(t)$. The requested packets will be stored in the corresponding queue. At TTI t, the queue length of flow f is $Q_f(t)$, which is determined by the queue length of the last TTI $(t-1)$, the new arrival of requests $Y_f(t)$, and the granted data size $G_f(t)$ at this TTI:

$$Q_f(t) = Q_f(t-1) + Y_f(t) - G_f(t) \tag{1}$$

in which $G_f(t) \leq (Q_f(t-1) + Y_f(t))$. The granted data size of each flow is determined by the resource allocation scheme, which can be calculated by:

$$G_f(t) = \sum_{b=1}^{B} x_{f,b}(t) \cdot C_{f,b}(t) \tag{2}$$

where $x_{f,b}(t)$ is the allocation indicator. $x_{f,b}(t) = 1$ if flow f is assigned with RG b at TTI t, otherwise $x_{f,b}(t) = 0$. The unit of $C_{f,b}(t)$, $Q_f(t)$, $Y_f(t)$, and $G_f(t)$ is the unit RG capacity.

Let $N_f(t)$ denote the number of packets of flow f that have been requested from $t = 1$ to t. N_f denotes the total number of requested packets of flow f. $E_f(j)$ denotes the received pre-FEC BER of packet j from flow f. The pre-FEC BER threshold of flow f is ET_f. Packets with $E_f(j) > ET_f$ will be re-transmitted, which will cause extra delay. $m_f(t)$ is the number of latency-satisfied packets from flow f at TTI t, which are scheduled packets with the overall latency satisfying the delay budget requirement D_f. Therefore, the total number of latency-satisfied packets of flow f will be $M_f = \sum_{t=1}^{T} m_f(t)$.

A. Problem formulation

The objective of the system is to optimize the mmWave resource allocation and scheduling so that the average ratio of latency-satisfied packets ($\frac{M_f}{N_f}$) will be maximized. To facilitate the DRL reward design which will be discussed in Section II-C, here the harmonic mean $\mathcal{H}(\frac{M_f}{N_f})$ is considered. Different from the arithmetic mean widely used, the harmonic mean tends to emphasize the impact of small outliers [11], which is desired in a scheduling problem as we want to avoid flows with a very low ratio of latency-satisfied packets. The ratios of latency-satisfied packets are utilized so that the flow-specific latency thresholds are used as benchmarks only within one flow, other than shared across all the flows, to avoid unfairness in resource allocation.

We formulate the problem as follows:

$$\max_{x_{f,b}(t)} \mathcal{H}(\frac{M_f}{N_f}) = (\frac{1}{F}\sum_{f=1}^{F}(\frac{M_f}{N_f})^{-1})^{-1} \tag{3}$$

$$s.t. \quad x_{f,b}(t) \in \{0,1\}, \forall f, b, t \tag{4}$$

$$\sum_{f=1}^{F} x_{f,b}(t) \leq 1, \forall b, t \tag{5}$$

where (4) shows that RG assignment variables are binary, and (5) suggests that each RG can only be assigned to one flow. The solution of (3) aims to find the best resource allocation at each TTI for all flows and RGs. This problem is difficult for the following reasons: i) constraints (4) and (5) makes the problem combinatorial; ii) the number of RGs and the number of flows can be very large, which makes the optimization problem more challenging; iii) the objective does not have closed form expressions in terms of $x_{f,b}(t)$. A direct optimization is difficult. To solve ii) and iii), instead of directly deciding $x_{f,b}(t)$, the action of the proposed DRL-based scheduler is modified to select the optimal scheduling and resource allocation rule for each TTI, which will determine $x_{f,b}(t)$ following different scheduling objectives. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_y\}$ denote the set of candidate rules. At TTI t, the selected rule is $P(t) \in \mathcal{P}$. The problem becomes:

$$\max_{P(t)} (\frac{1}{F}\sum_{f=1}^{F}(\frac{M_f}{N_f})^{-1})^{-1} \tag{6}$$

with $P(t)$ satisfying constraints (4) and (5).

B. DRL framework

In (1), the queue length of flow f at TTI t is determined by the queue length of the last TTI $Q_f(t-1)$, the requested data $Y_f(t)$, and the granted data size $G_f(t)$. The scheduler will make the decision based on the observation of channel conditions, queue status, and request patterns. The decision-making is partly random due to the arrival request patterns and partly dependent on the available resource allocation rules in the scheduler. Therefore, the queue state (1) can be modeled as a Markov decision process (MDP)[9]. We use Q-learning algorithm, the most widely used reinforcement learning method, to solve the MDP problem. Considering a large number of RGs and flows, dynamic optimization environment and targets, a deep neural network (DNN) is used in the proposed system instead of a conventional Q-table. The deep Q network (DQN) will be trained to reflect the mapping between the state and action spaces during the DRL process.

In the proposed DRL-based scheduling algorithm, we define the period of time in which the interaction between the agent and the environment takes place as an episode, and each TTI t corresponds to a step of an episode. The state space of flow f at TTI t includes the head-of-line (HoL) latency of the top packet in the queue, denoted by $s_{f,1}(t)$; the requested data size $s_{f,2}(t) = Y_f(t)$; the flow priority indicator $s_{f,3}$; and the capacity (spectral efficiency) of all RGs $\mathbf{s}_{f,4}(t) = \{C_{f,b}(t), \forall b\}$. The state at t can be expressed as:

$$\mathbf{s}(t) = \{\mathbf{s}_1(t), \mathbf{s}_2(t), \ldots, \mathbf{s}_F(t)\} \tag{7}$$

where $\mathbf{s}_f(t)$ is the observed state of flow f:

$$\mathbf{s}_f(t) = \{s_{f,1}(t), s_{f,2}(t), s_{f,3}, \mathbf{s}_{f,4}(t)\} \tag{8}$$

From (7) and (8), the size of $\mathbf{s}(t)$ will be $3F+BF$, or $3F+BU$ if we consider one UE may have multiple flows. The state size

is dependent on the number of RGs and UEs, from which it can be seen the computational complexity increases with available bandwidth resources and the number of users.

In our previous work of DRL scheduler which aims to optimize delay with a small RB space [9], the action is defined as the choice of $x_{f,b}(t)$. In this case, the size of action space will be $|\mathcal{A}| = F^B$ which will be extremely large in a multi-flow wideband system. As analyzed in Section II-A, to cope with the large RG space and the service-oriented QoS requirements, the action space $\mathcal{A}$ in the proposed scheduler consists of resource allocation rules that have been widely investigated and implemented by network operators with different scheduling targets, i.e., $\mathcal{A} = \mathcal{P}$. The size of the action space reduces to $|\mathcal{A}| = |\mathcal{P}|$, which are independent of B and F, therefore improving the convergence efficiency.

In the learning process, the DQN agent will maintain a critic $Q(s, a)$, which takes observation of state $\mathbf{s}(t)$ and action $a(t)$ as inputs and returns the expectation of the long-term reward:

$$Q(\mathbf{s}(t), a(t) \mid \theta_Q) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(t+i) \mid \mathbf{s}(t), a(t)\right] \quad (9)$$

Algorithm 1 BER and delay aware scheduling algorithm based on DRL

if training **then**
  Initialize environment and generate traffic patterns
  Initialize the time, states, action and replay buffer $\mathcal{K}$
  for each episode **do**
    for each TTI t **do**
      Load the status of the RGs
      Observe state $\mathbf{s}(t)$ as shown in (7)
      $\epsilon = \max(\epsilon \cdot d, \epsilon_{\min})$
      Sample $r \sim \mathcal{U}(0, 1)$
      if $r \leq \epsilon$ **then**
        select an action $a(t) \in \mathcal{A}$ randomly
      else
        Select an action $a(t)$ using (11)
      end if
      Compute the reward $r(t)$
      Observe the next state $\mathbf{s}'$
      Store the experience $(\mathbf{s}(t), a(t), r(t), \mathbf{s}')$ in $\mathcal{K}$
      From $\mathcal{K}$, sample a random minibatch of K experiences: $\{e_k \equiv (\mathbf{s}_k, a_k, r_k, \mathbf{s}'_k)\}$.
      Set $y_k = r_k + \gamma \max_{a'} Q(\mathbf{s}'_k, a' \mid \theta_Q), \forall k$
      Perform the gradient optimization on loss $L = \frac{1}{K}\sum_{k=1}^{K}(y_k - Q(\mathbf{s}_k, a_k \mid \theta_Q))^2$ to get the optimal θ_Q^*
      Update θ_Q from θ_Q^*, with the target method
      $t = t + 1$
      $\mathbf{s}(t) = \mathbf{s}'$
    end for
  end for
  save θ_Q and the agent
else
  Load the agent
  Observe the state and output the action using (11)
end if

where $r(t+i)$ is the instantaneous reward, γ is the discount factor, and θ_Q represents the parameter values of the DQN. The long-term reward is:

$$R(t) = \sum_{i=0}^{\infty} \gamma^i \cdot r(t+i) \quad (10)$$

The training and testing algorithm is illustrated in algorithm 1. $Q(s, a)$ is initialized with random parameters θ_Q and will periodically update over the training process as it interacts with the environment. At each step or TTI t, with probability ϵ which updates with decay rate d, the system will randomly generate an action, otherwise, it will observe the current state and select the action with the greatest Q-value:

$$a(t) = \underset{a(t)\in\mathcal{A}}{\arg\max}\ Q\left(\mathbf{s}(t), a(t) \mid \theta_Q\right) \quad (11)$$

After taking a certain action, the system calculates the instantaneous reward $r(t)$ and observes the next state $\mathbf{s}'$. The transition experience $(\mathbf{s}(t), a(t), r(t), \mathbf{s}')$ is stored in a replay memory $\mathcal{K}$. After that, a random minibatch of K experiences will be selected for the optimization and update of θ_Q.

C. Reward design

One key step of DRL design is to customize the reward function for the desired target. In this work, the problem is to optimize (6) with restrictions (4) and (5). The reward of the DRL system is designed as follows. At each TTI t for each flow f, all packets that have been requested will be categorized into four types as depicted in Fig. 3. For flow f at TTI t, the total number of requested packets is $N_f(t)$; for those packets that have been scheduled, the total number of scheduled packets whose latency satisfy the service latency requirement D_f is $M_f(t)$, the total number of packets whose latency exceed D_f is $L_f(t)$; for the packets in the queue waiting to be scheduled, the total number of packets whose queuing time already exceed delay budget D_f is $W_f(t)$. The reward of flow f at TTI t is defined as:

$$r_f(t) = 1 - \frac{L_f(t)}{M_f(t)} - \frac{2W_f(t)}{M_f(t)} \quad (12)$$

In (12), the second term $-\frac{L_f(t)}{M_f(t)}$ reflects negative feedback if the current scheduling method has resulted in too much latency, whereas the third term $-\frac{2W_f(t)}{M_f(t)}$ with the weight factor 2 indicates more significant negative feedback to prevent latency-failure packets from queuing up and leading to large

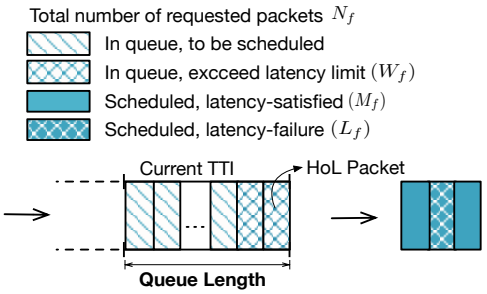

Fig. 3: Illustration of packet and queue status.

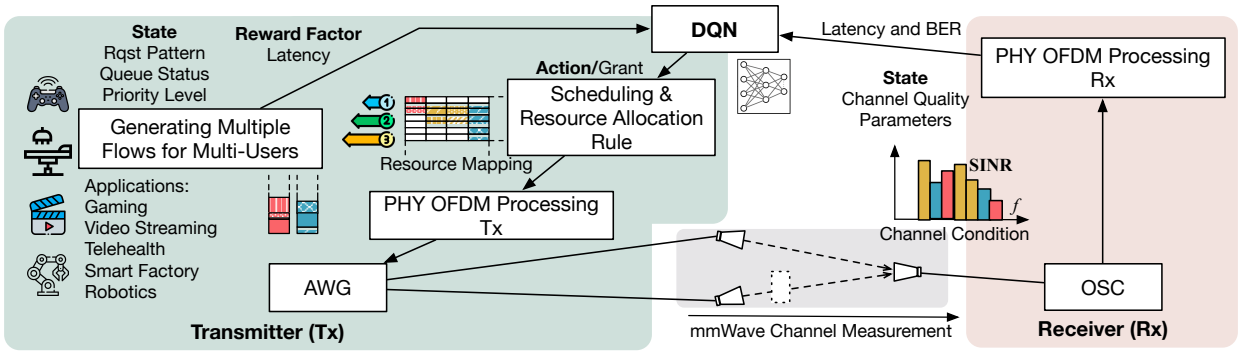

Fig. 4: The architecture and modules of the mmWave-RoF testbed with DRL-based scheduler.

queuing delay. In (12), $M_f(t)$ rather than $N_f(t)$ is used for the denominator to reduce the influence of random packet arrival, therefore the reward can better reflect the scheduling efficiency. Channel conditions can influence the reward as poor BER will lead to re-transmissions which cause extra latency. The overall reward of TTI t is the weighted sum of r_f:

$$r(t) = \frac{1}{F}\sum_{f=1}^{F} r_f(t) \tag{13}$$

If there are no new requests, at the end of the scheduling process, the third term in (12) will vanish as all packets have been processed ($W_f = 0$):

$$r = \frac{1}{F}\sum_{f=1}^{F} r_f = \frac{1}{F}\sum_{f=1}^{F}(1 - \frac{L_f}{M_f}) \tag{14}$$

With all packets scheduled, we have $L_f = N_f - M_f$, then (14) is equal to:

$$\begin{aligned} r &= \frac{1}{F}\sum_{f=1}^{F}(1 - \frac{N_f - M_f}{M_f}) \\ &= \frac{1}{F}\sum_{f=1}^{F}(2 - \frac{N_f}{M_f}) \\ &= 2 - \frac{1}{F}\sum_{f=1}^{F}(\frac{M_f}{N_f})^{-1} \end{aligned} \tag{15}$$

When (15) is maximized, (6) is maximized accordingly, which complies with the optimization objective.

III. Operation Implementation

The mmWave radio access testbed with DRL-based scheduler consists of several key function modules as shown in Fig. 4. The directions of arrows in Fig. 4 indicate the processing flow in an episode that follows the request-grant cycle. The delays of different stages in the scheduling process indicated in Fig. 2 are summarized in Table I. The delay parameters are based on [12], in which $2km$ standard single-mode fiber (SMF) and $50m$ wireless distance is assumed.

In the system, the flow generation module will generate packets based on different application types. The DRL agent will take the decision action provided with the state information from the flow generation module and the mmWave channel module. In this work, mmWave channel information is obtained through experimental measurement of multi-user RoF-mmWave testbed instead of channel simulation. The mmWave channel module consists of physical layer (PHY) orthogonal-frequency-division-multiplexing (OFDM) processing module in the transmitter and receiver side (Tx and Rx). An arbitrary wave generator (AWG) and a digital oscilloscope (OSC) are used in the experiment to generate analog OFDM waveforms and to capture the received waveforms for channel information extraction. The detailed implementation of each module will be illustrated in the following subsections.

TABLE I: Delay Components

Propagation Delay	$70.86\mu s$	b, d, g, i, k
UE Processing	$0.32ms$	a, e, j
Scheduler Processing	$62.98\mu s$ (14 symbols)	c
Re-transmission Processing	$0.21ms$	h
Queuing Delay	Traffic-based	f

A. Flow generation module

The DRL system involves multiple users that are using applications with different QoS requirements. One UE can have multiple active flows simultaneously. For the flows implemented in the system, the packet arrival pattern, QoS priority, delay budget, and other key flow-specific parameters are summarized in Table II. There are four types of flows in the system. The priority indicators listed in Table II are based on 3GPP QoS specification [13]. Service with a smaller priority value has higher priority in the scheduling process. The priority value is a component ($s_{f,3}$) of the DRL state input. Among the applications, the flow for robotics control ($f1$) has critical latency requirement (1ms) and high data rate [2]; the flow for conventional video streaming or Web file transfer protocol (FTP) transmission ($f2$) can tolerate more latency; the flow for serious gaming or smart factory application ($f3$) also has critical latency requirement but can be supported with moderate data rate; the flow for telehealth ($f4$) such as telediagnosis and surgery may require latency on the order of 1-10ms with data rate around 100Mbps [2].

Different statistical models are employed for packet sizes and arrival intervals to mimic flow behaviors in reality. For time-sensitive traffic such as robotic control, the packet arrival

TABLE II: Flow Parameters

Service	Robotics	Video Streaming	Gaming/Factory	Telehealth
Priority	30	56	30	56
Speed (Mbps)	300-350	10	3	300
Delay Bdgt.	1ms	5ms	1ms	2ms
Pkt. Size	Rand	Log Norm.	*Gaussian*	*Poisson*
Pkt. Interval	*Bernoulli*	*Poisson*	Fixed	Cont.
UE	1 (f1)	1 (f2)	2 (f3)	2 (f4)

processes follow *Bernoulli* processes [14]. In the DRL system, the probability of packet arrival is 0.8 for each TTI at the data rate of randomly generated 300-350Mbps to simulate control signaling. The video streaming is abstracted from FTP models with the file size using Log-normal variable ($\mu = 11, \sigma = 0.1$, leading to an average file size of $0.1Mb$) [15]. The FTP file arrival interval follows *Poisson* distribution with $\lambda = 100$ and packets are generated from each file accordingly. Flows following live streaming video model can cause a significant queuing delay in upstream transmissions due to the influx of FTP file packets. For real-time gaming flows or smart factory signaling, normally distributed packet arrival intervals ($\mu = 320\mu s, \sigma = 65\mu s$) and normally distributed sizes ($\mu = 110b, \sigma = 40b$) are implemented [15]. The packets of real-time gaming usually have small packet sizes and sparse arrival intervals. To model telehealth traffic, the packet size follows *Poisson* distribution at the rate of 300Mpbs and packets will occur at every TTI.

B. Scheduling and resource allocation rules

The action of the DRL-based scheduler is to select the optimal resource allocation rule for the current TTI. The candidate rules are summarized in Table III. Different rules have different scheduling objectives [16], [17]. In Table III, the first rule targets to maximize the signal to interference and noise ratio (SINR) based on UE channel conditions. The proportional fair (PF) rule considers the trade-off between fairness and spectral efficiency, and it is aware of the channel condition and transmission history of UEs. The exponential (EXP) rule uses an exponential function to take into account channel condition, spectral efficiency, HoL latency, and QoS requirements. Similarly, the LOG rule utilizes a logarithmic function to evaluate these factors. In both EXP and LOG rules, flows are prioritized when their HoL delays are approaching the delay deadline. The implementation details can be found in [17]. In the proposed DRL algorithm, the action at each TTI is optimized with respect to different traffic and channel conditions. For example, max-SINR rule may be favored over LOG rule when the channel condition suddenly deteriorates.

TABLE III: Resource Allocation Rules (Action Space)

Rule	Feature	Objective
Max-SINR	Channel	Best BER
PF	Channel & Speed Aware	Fairness & Throughput
EXP	Channel-Speed-Delay Aware	Fairness & Bounded Delay
LOG	Channel-Speed-Delay Aware	Fairness & Bounded Delay

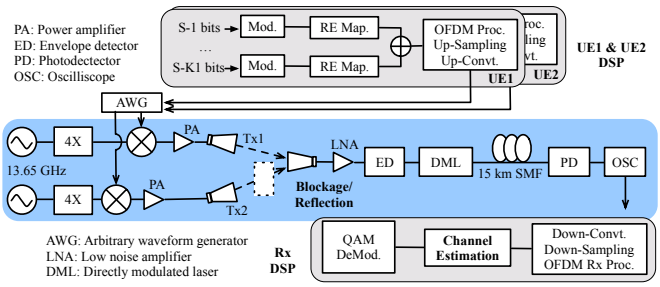

Fig. 5: Experimental testbed for mmWave channel measurement.

TABLE IV: OFDM and RG Numerologies

Numerology, μ	4
Subcarrier spacing	240kHz
Effective subcarrier number	840/2048
Effective bandwidth	201.6MHz
Number of symbols per TTI	8
TTI duration	$35.4\mu s$
RB size	12 subcarriers
RG size in frequency	5RB/60 subcarries
RG size in time	2 symbol duration
Number of RGs per TTI	56
Modulation	QPSK/16QAM

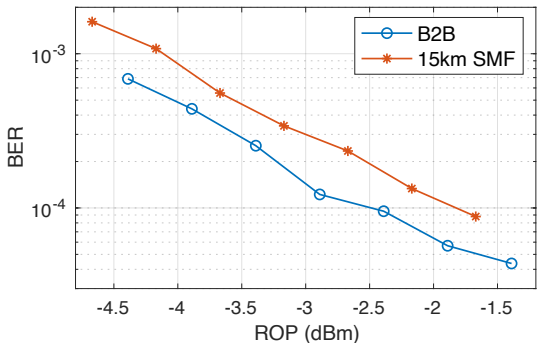

Fig. 6: BER performance versus ROP in back-to-back (B2B) and fiber transmission scenarios.

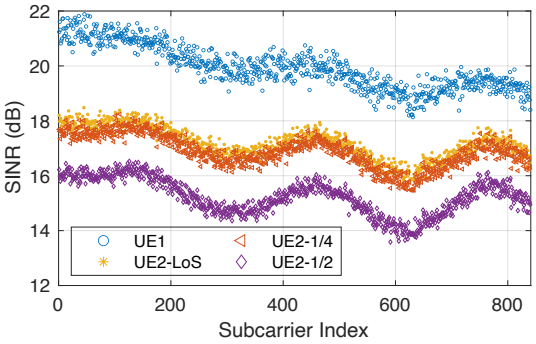

Fig. 7: SINR per subcarrier of both UEs using 16QAM in different scenarios.

C. Experimentally collected mmWave channels

The experimental testbed setup to obtain the mmWave channel information is depicted in Fig. 5, in which two UEs are

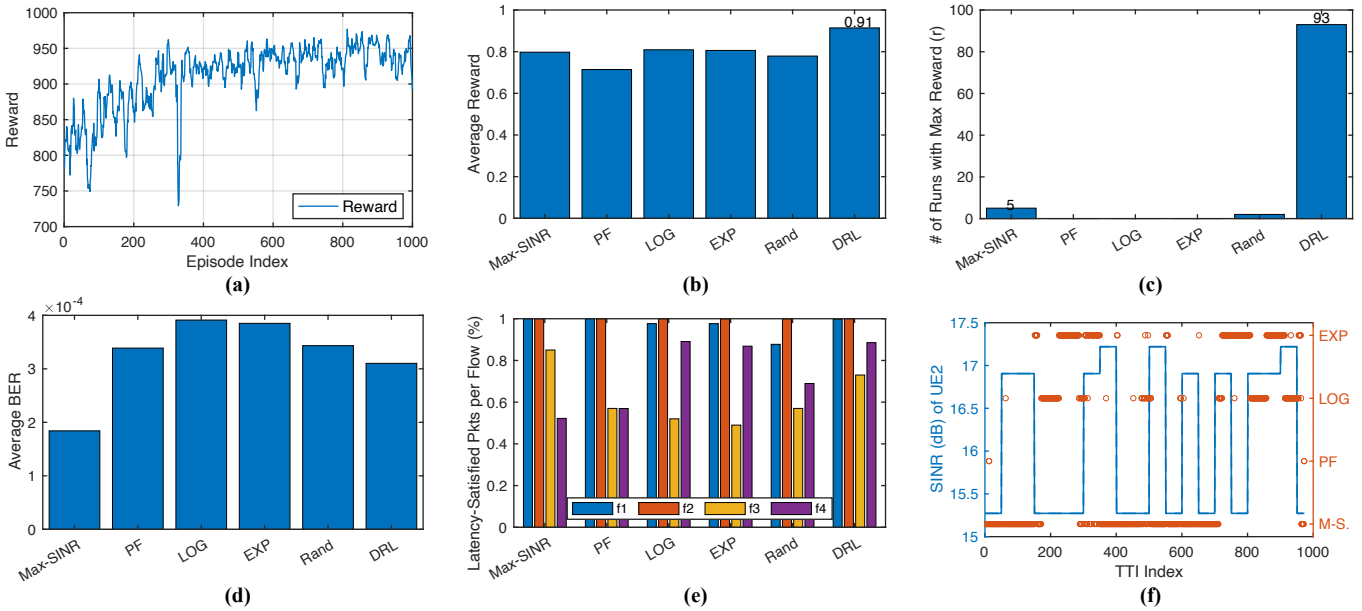

Fig. 8: (a) Reward convergence of the training process. (b) The average reward of different rules for 100 test runs. (c) The number of runs with maximum reward achieved per rule. (d) The overall BER per rule. (e) $\frac{M_f}{N_f}$ per flow for different rules. (f) UE2 SINR variation and the corresponding rule selection per TTI.

accessing one RRU through RoF-mmWave uplinks consisting of $1m$ wireless link and $15km$ SMF. Due to the devices available in the lab, there are two UEs and four flows tested in the system without loss of generality. In reality, more UEs can be implemented when needed. The UE-flow mapping is indicated in Table II. For each UE, the EVM of each RB will be measured and converted to SINR as a channel quality parameter for the scheduling processing. For more efficient processing, RBs are grouped to a RG when being allocated. Subcarriers and symbols in one RG have the same QAM modulation. The OFDM numerology and frame design are based on 3GPP 5G specification [18]. The OFDM and RG numerology are summarized in Table IV.

In the experiment, the operating mmWave frequency is $54GHz$, generated by quadrupled $13.65GHz$ RF sources. OFDM waveforms synthesized by the AWG will be modulated to the mmWave carriers through electrical mixers. The mmWave signals are then amplified and radiated by horn antennas. At the receiver side, a horn antenna will capture the signals, followed by an envelope detector that down-converts the signal to the baseband. For RoF mobile fronthaul transmission, the signal is converted to the optical domain through a directly modulated laser (DML) and converted back to the electrical domain by a photodetector (PD) after fiber transmission. Finally, the received signal will be collected by an OSC for digital signal processing.

The BER versus received optical power (ROP) performance of the testbed is shown in Fig. 6. For the channel measurement, the testbed is set at the optimal operating condition (ROP = -1.5dBm). To realize the dynamic channel conditions of mmWave links such as reflection, blockage, and reduced transmission power, channel variation is introduced for UE2. The channel of UE2 is measured with three conditions: i) LoS link; ii) the link is 1/4 blocked (slightly blocked); iii) the link is 1/2 blocked (severely blocked), while UE1 always has an LoS link. The channel SINR is shown in Fig. 7, which is calculated from experimentally obtained EVM [19]. In the scheduling process, each channel condition will last for 50 TTIs and randomly switch to the next condition. Different channel conditions and rule selection will lead to different BER performance. Upon decoding the received signals, the scheduler will check the packet BER per flow. If the BER exceeds the pre-set threshold ET_f (we use $ET_f = 6.9 \times 10^{-4}, \forall f$, considering forward error correction [20]), re-transmission will be triggered and the overall packet latency will hence become longer.

IV. Evaluation and Discussion

We create a DQN agent with recurrent neural network (RNN). There are three hidden layers between the input layer and the output layer: two dense layers and one long short-term memory (LSTM) layer, which have 30, 20, 16 neurons, respectively. The hyper-parameters of the DQN are summarized in Table V. The convergence plot of the training process is presented in Fig. 8(a). It can be seen that after around 600-episode training, the reward starts to converge. Considering the episode duration, the convergence time will be approximately 21s if the computational resources are sufficient such that the processing time is less than the designed value. In a real-world application, the convergence time will depend on the hardware capability. The episode duration is dependent on the design of the system and can be modified as required by the traffic and channels, and the computational power at the CU/DU processing units. Also note that once the agent is trained, the computation time for inference is negligible. The model is also adaptive to moderate channel or traffic variations, which will be introduced later in the section. With

such a level of variations, the agent does not require additional time for re-training. The fluctuation of the converged reward is caused by the randomness of the traffic patterns as indicated in Table II. Generally, the maximum average reward (1000) per episode can be achieved if the traffic load is light. However, in that case, the DRL agent can randomly choose any action to fulfill the latency requirements. Therefore, the traffic load in the paper is set to a heavier case to exploit the advantages of DRL.

TABLE V: DRL Hyper-parameters

Number of episodes	1000	Experience replay length	10^6
Number of steps per episode	1000	Discount factor	0.99
Batch size	64	ϵ decay d	10^{-4}
Sequence length	20	Learning rate	10^{-4}

We define a test run as the scheduling over 1000 TTIs with randomly generated request patterns based on Table II. The DRL agent is tested for 100 test runs, and the performance is evaluated and compared to the four single-target resource allocation rules listed in Table III. The case of randomly selecting rules TTI-by-TTI is also presented as a reference ('Rand'). A higher reward value indicates lower percentages of latency-failure packets, as indicated in (14). Fig. 8(b) presents the average reward of 100 test runs using different scheduling and resource allocation schemes. It can be seen that the proposed DRL algorithm can achieve an average reward of $r = 0.91$. If we assume all the packets as from an effective 'flow', the effective number of packets $\bar{M}$ and $\bar{L}$ can be used to calculate reward as $r \equiv 1-\frac{\bar{L}}{\bar{M}}$, and the effective ratio of latency-satisfied packets will be $\frac{\bar{M}}{N} = \frac{\bar{M}}{\bar{L}+\bar{M}} = \frac{1}{\frac{\bar{L}}{\bar{M}}+1} = \frac{1}{(1-0.91)+1} = 92\%$. However, among single-rule cases, LOG rule can achieve the best reward of 0.81. The proposed DRL algorithm can achieve 12% average reward improvement in comparison. Fig. 8(c) shows the number of times for each scheme to achieve the highest reward. Compared to other single-target schemes, the proposed DRL algorithm predominantly achieves the highest reward for 93 times out of 100 test runs.

We also investigate the BER and latency performance of the DRL-based scheduling. We select one test from the 100 test runs for result visualization. Fig. 8(d) shows the average BER of all packets for each resource allocation scheme, the proposed DRL scheme can achieve the second-best BER performance, only worse than the max-SINR scheme whose target is to minimize BER. Fig. 8(e) shows the ratio of QoS latency-satisfied packets per flow ($\frac{M_f}{N_f}$) for all schemes. Compared to latency-aware LOG and EXP rules, the DRL algorithm is able to improve $\frac{M_f}{N_f}$ of $f3$ and $f4$ (from UE2 with channel variation, for UE-flow mapping, see Table II), without sacrificing the performance of $f1$ and $f2$ (from UE1 with stable LoS links). Regarding the issue of allocation fairness, it can be seen that there are small ratio value differences within one UE, (between $f1$ and $f2$, between $f3$ and $f4$), indicating flows are assigned with similar portions of RGs based on the requested amount using the proposed algorithm. The differences in ratios are ultimately influenced by the channel quality but not the inter-flow latency threshold differences. Overall, the proposed DRL algorithm can jointly optimize BER and latency performance.

Fig. 8(f) presents the rule selection per TTI with respect to the channel variation of UE2. The blue curve indicates the SINR fluctuation of UE2, from which it is shown that each channel state lasts for 50 TTIs. As the rule selection can be jointly affected by the channel variations and flow request patterns, it can be seen that the pattern of rule selection synchronizes well with the channel SINR variation. The results show that the DRL system can react adaptively to channel condition variations.

V. Conclusions

A DRL-based scheduler operating with both latency and channel condition awareness is proposed and verified for service-oriented multi-user mmWave RANs. The operation of the DRL scheduler is verified with experimental validation of RoF-mmWave channel conditions and variations, as well as various service flows with different QoS requirements. Among all the test runs, the DRL algorithm predominantly achieves the highest reward, providing at least 12% average reward improvement compared to other single-target schemes. Results also show that the proposed DRL system can operate adaptively with channel variations and jointly optimize BER and latency performance simultaneously. The proposed DRL system has been demonstrated as a promising AI/ML-based technique that is applicable to the post-5G RAN systems.

References

[1] Y. Alfadhli, M. Xu, S. Liu, F. Lu, P.-C. Peng, and G.-K. Chang, "Real-time demonstration of adaptive functional split in 5G flexible mobile fronthaul networks," in Proc. Opt. Fiber Commun. Conf. and Exposition (OFC), San Diego, CA, USA, 2018, pp. 1–3.

[2] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat and H. Dai, "A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3098-3130, Fourthquarter 2018, doi: 10.1109/COMST.2018.2841349.

[3] Q. Hu, Y. Liu, Y. Yan, and D. M. Blough, "End-to-end Simulation of mmWave Out-of-band Backhaul Networks in ns-3." in WNGW 2019 with ns-3. Association for Computing Machinery, pp. 1–4. doi:https://doi.org/10.1145/3337941.3337943.

[4] 3rd Generation Partnership Project (3GPP), "User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone.," 3GPP, Tech. Spec. 38.101-2, 2020, V16.5.0.

[5] R. Zhang et al., "An Ultra-Reliable MMW/FSO A-RoF System Based on Coordinated Mapping and Combining Technique for 5G and Beyond Mobile Fronthaul," in Journal of Lightwave Technology, vol. 36, no. 20, pp. 4952-4959, 15 Oct.15, 2018, doi: 10.1109/JLT.2018.2866767.

[6] M. Feng and S. Mao, "Dealing with Limited Backhaul Capacity in Millimeter-Wave Systems: A Deep Reinforcement Learning Approach," in IEEE Communications Magazine, vol. 57, no. 3, pp. 50-55, March 2019, doi: 10.1109/MCOM.2019.1800565.

[7] D. Han, W. Chen and Y. Fang, "Joint Channel and Queue Aware Scheduling for Latency Sensitive Mobile Edge Computing With Power Constraints," in IEEE Transactions on Wireless Communications, vol. 19, no. 6, pp. 3938-3951, June 2020, doi: 10.1109/TWC.2020.2979136.

[8] S. Tseng, Z. Liu, Y. Chou and C. Huang, "Radio Resource Scheduling for 5G NR via Deep Deterministic Policy Gradient," 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICCW.2019.8757174.

[9] T. Zhang, S. Shen, S. Mao and G. -K. Chang, "Delay-aware Cellular Traffic Scheduling with Deep Reinforcement Learning," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322560.

[10] S. Shen, T. Zhang, S. Mao, and G.-K. Chang, "DRL-Based Channel and Latency Aware Scheduling and Resource Allocation for Multi-User Millimeter-Wave RAN," in Proc. Opt. Fiber Commun. Conf. and Exposition (OFC), 2021, pp. 1-3.
[11] Komić, J. "Harmonic Mean," In: Lovric M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg. 2011. pp.622-624.
[12] E. Dahlman, et al., (Ericsson), "Scheduling," in 5G NR: the next generation wireless access technology, 1st ed., Academic Press, 2018, pp. 275-299.
[13] 3GPP, "System Architecture for the 5G System (5GS)," 3GPP, Tech. Spec. 23.501, 2019, V16.2.0.
[14] 3GPP, "Study on scenarios and requirements for next generation access technologies," 3GPP, Tech. Rep. 38.913, 2017, v14.2.0.
[15] G. White, et al., "Low Latency DOCSIS: Technology Overview," DOCSIS Research and Development, CableLabs, Feb. 2019.
[16] 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," in IEEE Communications Surveys & Tutorials, vol. 15, no. 2, pp. 678-700, Second Quarter 2013, doi: 10.1109/SURV.2012.060912.00100.
[17] A. Ahmad, M. T. Beg and S. N. Ahmad, "Resource allocation algorithms in LTE: A comparative analysis," 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 2015, pp. 1-6, doi: 10.1109/INDICON.2015.7443115.
[18] 3GPP, "NR Physical channels and modulation.," 3GPP, Tech. Spec. 38.211, 2020, V.16.2.0.
[19] R. A. Shafik, M. S. Rahman and A. R. Islam, "On the Extended Relationships Among EVM, BER and SNR as Performance Metrics," 2006 International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, 2006, pp. 408-411, doi: 10.1109/ICECE.2006.355657.
[20] Juniper, "Forward Error Correction (FEC) and Bit Error Rate (BER)," Juniper, Dec. 2020.