Radio Resource Management in Multi-numerology 5G New Radio featuring Network Slicing

Karim Boutiba
EURECOM
Sophia Antipolis, France
karim.boutiba@eurecom.fr

Miloud Bagaa
Aalto University.
CSC-IT Center for Science Ltd.
Espoo, Finland
miloud.bagaa@aalto.fi, miloud.bagaa@csc.fi

Adlen Ksentini
EURECOM
Sophia Antipolis, France
adlen.ksentini@eurecom.fr

***Abstract*—5G New Radio (NR) introduces several key features to support the new emerging vertical industry use-cases, mainly: (1) Different numerology that gives more flexibility in managing time slot duration, and hence satisfying different delay requirements; (2) Bandwidth part that permits dedicating parts of the bandwidth to ensure different data rate requirements. However, although 5G NR introduces several enhancements, it makes radio resource management, more precisely resource scheduling, more complex and challenging. In this paper, we address the challenge of radio resource management in 5G NR featuring network slicing. We introduce a novel scheduling solution based on Deep Reinforcement Learning (DRL) to allocate resources and numerology for UEs to satisfy their different requirements. We evaluated the solution for different network configurations and compared its performance with the maximum achievable throughput. Simulation results demonstrated the efficiency of the proposed algorithm to allocate resources and the ability to scale for larger bandwidths covering both Frequency Range 1 (FR1) and FR2, as well as serving a higher number of User Equipment (UE).**

I. Introduction

The 5^{th} generation (5G) of wireless networks is designed to support variant network services with different requirements. 5G services are organized into three principal categories, which are ultra-reliable and low-latency communication (uRLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) [1]. These use-cases have conflicting requirements that require a radio design with high customization and flexibility to meet the different conditions. Indeed, while the eMBB services need more bandwidth to satisfy the high throughput requirements, the uRLLC services require a shorter time slot duration to ensure the lower latency requirement. Finally, mMTC services need better frequency management to sustain a massive number of connected devices (mainly sensors/actuators).

To guarantee the heterogeneous requirements of 5G services and support flexibility, 5G NR (New Radio) introduces the BandWidth Part (BWP) concept. A BWP is a subset of contiguous physical resource blocks (PRBs) that share the same characteristics, mainly the subcarrier spacing (SCS) [2]. 5G NR offers 5 configurations with different SCS (Table I). Each configuration is indexed by a scalar called numerology μ. The PRB shape, in time and frequency, is different for each numerology μ. The amount of frequency that a PRB occupies and its time slot duration can be expressed by $12 \times 15 \times 2^{\mu}$ KHz and $1/2^{\mu}$ ms, respectively. Therefore, by adapting the numerology concept, 5G NR reduces the slot duration down to 125 microseconds, considerably reducing the RAN latency, which is very beneficial for supporting uRLLC services.

On the other hand, the available bandwidth at a 5G base station can be divided into BWPs with the same or different SCS. A User Equipment (UE) can be configured with up to four BWPs and can use up to one BWP at a given time. Although the procedures of creating and configuring BWPs are standardized, deciding the size and the numerology for each BWP is still a challenging and open point. The problem is further difficult when a UE belongs to more than one slice, knowing that each slice has different requirements. Indeed, to schedule radio resources for a UE, different dimensions need to be considered: the numerology to use, the bandwidth part size, the network slice, and the type of 5G service.

In this paper, we tackle the challenges of radio resource allocation in 5G NR by proposing a novel Deep Reinforcement Learning (DRL) based scheduler that allocates resources for a list of UEs to satisfy their different slice' SLA requirements. The scheduler aims to select the numerology to be used and the number of resources allocated per UE at each time slot during a time window while taking into account the channel quality of the UE. Furthermore, we designed the scheduler to be independent of the number of users in the system, and we have modeled the state to make the solution scalable for larger bandwidths up to 400 Mhz, which correspond to the usage of the mmWave band.

The rest of the paper is organized as follows: Section II describes the resource allocation problem in 5G NR, the related work, and introduces DRL algorithms. Our proposed solution is presented in Section III and evaluated in Section IV. We conclude the paper in Section V.

II. Background

A. Problem description

Radio resource scheduling issue is one of the most interesting problems in 5G networks. Whilst in 4G, only the frequency domain is considered, in 5G, both frequency and time domains need to be considered due to the multi-numerology settings. The radio resource scheduler observes the radio channel as

a 2D grid, where the x-axis is the time window ΔT and the y-axis is the frequency bandwidth ΔF. The scheduler's objective is to divide the grid among different active users at the gNB relaying on a specific policy. The latter depends on the radio resource scheduler implementation. For example, the proportional fair scheduler equally divides the grid to satisfy the users, while the early-deadline scheduler privileges the users requesting low-latency communications. In this paper, we assume a network that consists of a set of UEs that compete to access the radio resources. The UEs can use different network slices, each characterized by different objectives, characteristics, and service level agreement (SLA). According to 3gpp specifications [3], a UE can belong to more than one network slice and can have different SLAs. The radio scheduler has to respect these SLAs by dividing the grid resources efficiently. The UE should use one numerology at a given time slot, but it can use more than one PRB. Last but not least, it is important to note that authors of [4] have proven that the resource allocation problem is NP-hard.

Table I: Numerology and slots duration

μ	SCS	Slot duration (ms)
0	15khz	1
1	30khz	0.5
2	60khz	0.25
3	120khz	0.125
4	240khz	0.0625

B. Related work

In [5], the authors proposed a heuristic-based solution to perform numerology multiplexing as well as resource allocation taking into account the Quality of Service [6] and channel quality. However, they did not address the optimality of their solution. The authors in [7] proposed a mixed-integer linear program to distribute the available bandwidth among the users considering different channel conditions and inter-band interference as a consequence of mixed numerologies. However, the numerology was fixed per user, and all the users had the same requirements in terms of QoS. The work in [8] designed a random forest-based decision algorithm to accomplish the numerology selection for each service. But, the proposed solution considered neither the frequency efficiency nor its optimality. In [9], the authors introduced an optimization method for resource and numerology allocation in multi-user scenarios. They have modeled the problem as a multi-scenario max-min Knapsack problem, which was solved by an integer programming solution.

In [10], the optimization problem was formulated as an integer linear program, and a linear relaxation of the problem was proposed. However, the authors ignored the latency, which is the main criteria in uRLLC services, and did not consider the coexistence of the different types of 5G services. In [11], the authors introduced an efficient heuristic approach to meet diverse QoS requirements of Machine-to-Machine (M2M) applications while achieving spectral efficiency. However, they ignored the time domain in the problem definition. Authors in [12] designed a deep reinforcement learning (DRL) model able to solve the radio resource scheduling problem under different numerology settings. But, the proposed solution did not consider using different numerologies in the same bandwidth. In [13], the authors leveraged deep reinforcement learning to design a model-free solution. Nevertheless, the presented problem assumed that all users belong to one and only one slice and have the same numerology, making the system less flexible.

Finally, it should be noted that all the cited solutions do not consider that a UE can belong to more than one slice with different requirements and consider only small instances of the problem.

C. Deep Reinforcement Learning Background

DRL is a Machine Learning (ML) technique that can be beneficial for 5G networks and beyond [14] to derive configuration or management decisions in a timely and efficient manner. In addition, the DRL technique has a tremendous benefit for Radio Access Networks (RAN), whereby the decisions should be taken in real-time (around $1ms$). Indeed, the DRL-based framework can learn with time and make fast decisions in a stochastic environment, providing self-configured and self-optimized network functions, such as radio resource allocation. A DRL framework has two actors: an agent and an environment. The agent observes a state S_t from the environment, applies an action a_t, gets a reward r_{t+1}, and hence the environment moves to a next state S_{t+1}. The agent can be in two modes: i) exploration mode, where the agent explores and builds the knowledge about the environment, and ii) exploitation mode, where the agent exploits the acquired knowledge by following the optimal policy π_* that gives for each state S_t the optimal action a_t^*.

The policy π_* maximizes future cumulative discounted reward G_t defined as follows:

$$G_t \doteq \sum_{k=0}^{T} \gamma^k r_{t+k+1} = r_{t+1} + \gamma G_{t+1} \tag{1}$$

With $\gamma \in [0, 1]$ defined as the discount rate penalizes the future rewards, and T equals to the time window which is finite for episodic problems (i.e., problems that ends when the environment is a final state) and infinite for continuing problems.

Accordingly, the ability of DRL to derive good decisions quickly, deal with unseen environments and be scalable make it suitable for solving the resource allocation problem in 5G NR, which is known as a NP-hard problem [4].

III. Deep Reinforcement Learning Radio Scheduler (DRL-RS)

A. DRL-RS general overview

The system's environment is described as a 2D matrix, where the x-axis represents the time slots over a window ΔT and the y-axis corresponds to the PRBs over a frequency bandwidth ΔF. Let δ_T and δ_F denote the minimum time slot and the frequency bandwidth that can be assigned to a

PRB, respectively. Formally, $\delta_T = 2^{\mu_{max}}$, where μ_{max} is the maximum numerology in the system, and $\delta_F = 2^{\mu_{min}}$, where μ_{min} is the minimum numerology in the system.

We consider a dynamic number of UEs in the system. Each UE can belong to more than one slice. For each slice, a UE has different throughput and latency requirements (SLA: Service Level Agreement). The throughput SLA is the minimum throughput that needs to be achieved by a UE for that slice, while the latency SLA is the maximum latency that a UE can not exceed when serving that slice. In operation, the UE needs to satisfy all its slice' SLAs. To capture that a UE can have more than one slice, we introduce the concept of virtual UEs. Formally, each UE consists of a set of virtual UEs, whereby each of which belongs to only one slice. The number of virtual UEs of a UE equals the number of slices where that UE is involved. The slices belonging to the same UE are grouped in groups. The time slots of members of the same group should neither overlap in time nor use different numerologies. For example, if we have a UE that belongs to 2 slices, we will consider 2 virtual UEs in our environment with the two extra constraints between these 2 virtual UEs. For the sake of simplicity, we called a virtual UE a simple UE.

Our scheduler will loop over the active UEs (i.e., UEs having data in their transmission queues) until either the resource grid is filled or all the UE's SLAs are met. For each UE, the scheduler chooses a time slot t, a numerology μ, and a number of resources N. This can be represented by a rectangle of shape $2^{\mu_{max}-\mu} * (N * 2^{\mu})$ stacked in the resource grid at time slot t.

Figure 1 illustrates a simple example that shows the functionality of DRL-RS. We consider 4 UEs with a latency SLA of 0.125ms, 1ms, 0.25ms, 0.5ms, and a throughput SLA that can be met with 1, 3, 1, 2 PRBs, respectively. The 2D matrix consists of a grid, whereby the size of each module of the grid is a rectangle $(\delta_T \times \delta_F)$. For simplicity and without losing the generality, the number of resources needed is computed according to the Modulation and Coding Scheme (MCS) assigned to a UE, which depends on the UE's Channel Quality Indicator (CQI) value. We consider $\mu_{max} = 3$, $\mu_{min} = 0$, $\Delta T = 1ms$ and $\delta_F = 8$. Let the red line represents the frequency boundary on which the future allocated resources are stacked. It is worth noting that DRL-RS needs to use different numerologies to satisfy different latencies. For instance, numerology 3 needs to be used to satisfy the latency SLA of the first UE. In the first iteration, for the first UE, the scheduler allocates 1 PRBs with numerology 3 at t = 0. Then, it allocates 2 PRBs for the 2^{nd} UE at t = 4 with numerology 1. The third UE takes 1 PRB using numerology 3 at $t = 1$, while the 4^{th} UE takes 1 PRB at $t = 2$ using numerology 2. The stop condition is not met yet; hence, the scheduler continues looping over the active UEs that did not meet their SLAs, i.e., the 2^{nd} and 4^{th} UEs. The 2^{nd} UE takes one more PRBs at $t = 4$ with numerology 1. Finally, the 4^{th} UE takes one PRB at $t = 2$ using numerology 2. At this step, all UEs' SLAs are met, and thus the scheduler finished the resource allocation process.

B. DRL-RS design

In the balance of this section, we describe our solution DRL-RS.

1) State: The state S^i observed by the model when serving UE i is composed of 4 vectors, which are F, T_i, O_i and E_i, respectively. Vectors F and T_i contain the frequency boundary (the red line in Figure 1) and the numerology used by UE i for each time slot t, respectively. If a UE is not using any numerology at time t, $U_{i,t}$ is set to -1. Meanwhile, vectors O_i and E_i contain the information about SLA of the current UE and the other UEs, respectively. O_i contains two values, O_i^{thg} measuring the throughput SLA and O_i^{lat} measuring the latency SLA. O_i^{thg} indicates the rate of the achieved throughput over the throughput SLA. The bigger O_i^{thg} is, the better performances of UE i becomes. Since a UE can achieve a throughput higher than its throughput SLA, O_i^{thg} can have a value bigger than 1. For instance, to decrease the state space size, we can limit O_i^{thg} with a maximum value O_{max}^{thg}. On the other hand, O_i^{lat} indicates the rate of the PRBs used before the latency SLA. The bigger O_i^{lat} is, the better performance of the UE i becomes. Note that a UE will respect the latency requirements if all the allocated PRBs are scheduled before the latency SLA, which is equivalent to $O_i^{lat} = 1$. Formally,

$$O_i^{thg} = \min\{\frac{achieved\ throughput}{throughput\ SLA}, O_{max}^{thg}\} \text{ with } O_{max}^{thg} > 1$$
$$O_i^{lat} = \frac{Number\ of\ allocated\ PRBs\ before\ latency\ SLA}{Total\ number\ of\ allocated\ PRBs}$$

E_i contains three values: N_i^{thg}, N_i^{lat} and Min_i^{thg}. N_i^{thg} and N_i^{lat} count the number of UEs excluding UE i that have met their throughput and latency SLAs, respectively. Min_i^{thg} is the smallest throughput SLA achieved by other UEs.

The state's design considers the scalability of the solution regarding the number of UEs, the SLA requirements, and the bandwidth size.

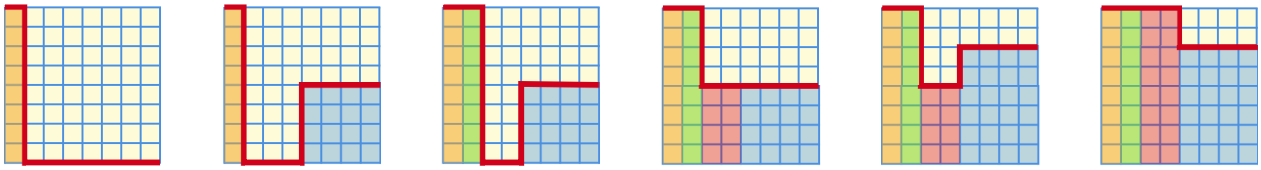
2) Actions: The agent takes three actions (t, μ, N): t is a time slot, μ is the numerology to use at the given time slot t, and N is the number of resources to allocate at t. The agent will allocate a rectangular shape of ($2^{\mu_{max}-\mu}$,($N*2^{\mu}$)) at time slot t and put it on the top of the frequency boundary line. Since we have a lot of not feasible actions, we added a pre-processing step in order to compute an action space A that contains only the possible actions at the current state S_t^i.

3) Reward: We have adopted an episodic approach; i.e., an episode is over when max T steps are reached, the resource grid is filled, or all UEs SLAs are met.

The agent gets the reward r defined as follow:

$$r = \begin{cases} \alpha * (O_{i,t}^{thg} - O_{i,t-1}^{thg}) + (1-\alpha) * \mathcal{S} & \text{if not done} \\ \mathcal{K} & \text{if done and SLAs are met} \\ \mathcal{M} & \text{otherwise} \end{cases}$$

Indeed, while the episode is still in progress and for each step, the agent takes a reward equivalent to the improvement made by the current action at step t since the previous step $t-1$. We formulate this improvement by the term $(O_{i,t}^{thg} - O_{i,t-1}^{thg})$.



(a) 1st UE ($t = 0$, $\mu = 3$, $N = 1$) (b) 2nd UE ($t = 4$, $\mu = 1$, $N = 2$) (c) 3rd UE ($t = 1$, $\mu = 3$, $N = 1$) (d) 4th UE ($t = 2$, $\mu = 2$, $N = 1$) (e) 2nd UE ($t = 4$, $\mu = 1$, $N = 1$) (f) 4th UE ($t = 2$, $\mu = 2$, $N = 1$), all SLAs are met

Figure 1: Example of resource allocation using DRL-RS

Table II: 5G NR Configurations

Index	Bandwidth (Mhz)	ΔT (ms)	Number of UEs	Number of slice per UE	Slices SLA list (Mbps, ms)	MCS
1	20	3	3	2	(9,10),(0.2,1)	16
2	20	3	10	2	(5,10),(0.2,1)	26
3	40	1	10	2	(5.5,10),(0.2,0.5)	16
4	40	1	10	2	(13,10),(0.7,0.25)	26
5	100	3	3	3	(40,10),(1,2),(0.2,1)	16
6	100	3	6	3	(40,10),(2,2),(0.7,1)	26
7	400	1	5	2	(110,10),(1,0.125)	16
8	400	1	20	1	(50,10)	26

Moreover, to minimize the number of steps needed to finish an episode, we added a penalty $\mathcal{S}$ for each step. Also, we added weights α and $(1-\alpha)$ to the two previous terms in order to control their contribution in the reward term. Once the episode is done, the agent takes a high positive reward $\mathcal{K}$ if the SLAs are met for all UEs, or a penalty $\mathcal{M}$ else.

C. DRL-RS detailed description

DRL-RS leverages the Deep Q-Network (DQN) algorithm [15] which is one of the most efficient DRL algorithms for continuous state space and discrete actions. DRL-RS executes two steps: decision making and updating the Q-Networks. In DQN, two networks are used: a local Q-Network and a target Q-Network. The latter is the same as the local network except that its parameters are updated every τ steps. They are combined to help the convergence and stabilization of the learning.

1) Decision making: The DRL-RS agent observes a state S_t^i for UE i and feeds it to the local Q-Network to get a discrete action distribution of one action a. Since we need three output values (t, μ, N), we partitioned the integer value a, using the method "the partitioning of an integer into different parts" introduced in [16], as follow: $t = a \div (\mu_{max} * N_{max})$ where N_{max} is the maximum number of resources that can be allocated to one UE and μ_{max} is the maximum numerology in the system. Note that $\div$ is the division of integer division and $\mod$ is the mod of integer division.

$\mu = a \mod (\mu_{max} * N_{max}) \div N_{max}$

$N = (a \mod (\mu_{max} * N_{max}) \mod N_{max}) + 1.$

Then, the agent removes the illegal actions (the actions that are not possible, for example, allocating resources that overlap with other existing resources) by setting their probabilities to a negative value. Then, we apply an ϵ-greedy approach to choose an action. This means that the agent will choose a random action over the possible actions with ϵ probability and the best action over the action distribution with a 1- ϵ probability. The value of ϵ decays overtime during the learning, pushing the agent to explore the environment at the beginning of the training and pushing it to exploitation over time.

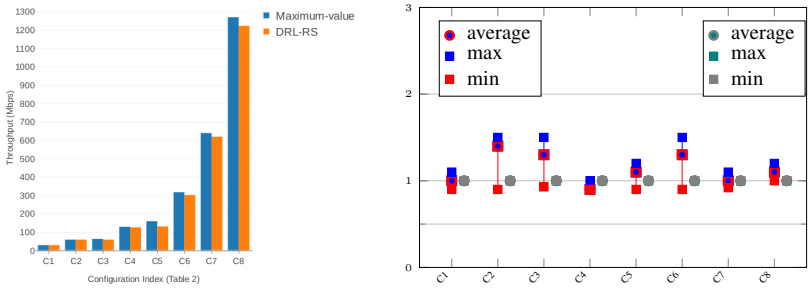
2) Updating the Q-Networks: At each step, the current state, the action, the next state, and the reward are stored in a buffer known as the replay buffer. The local Q-Network is updated using a random sample from the replay buffer, which reduces the correlation between the agent's experiences and increases the stability of the learning. Using mean square error (MSE) and ADAM optimizer [17], the parameters of the local Q-Network are optimized at every step by considering the local and target values, while the parameters of the target Q-Network are updated every τ^{-1} steps to stabilize the convergence of the algorithm.

IV. Performance Evaluation

In this section, we will introduce the DQN parameters used for training the DRL-RS agent. Then, we will evaluate the trained agent in a 5G simulated environment.

A. Training phase

We have trained the DRL-RS agent using 500 independent episodes. We have fixed the maximum number of steps at each episode T by 100. We have varied the reward parameters and chosen the values that stabilize the convergence of the model. We have set the penalty reward $\mathcal{M}$ to -0.02, $\mathcal{S}$ to -0.01, $\mathcal{K}$ to 5000 and α to 0.5. We have set O_{max}^{thg} to 2. We have employed two fully connected hidden layers of 64 nodes, both for the QNetworks. We have used a discount factor γ of 0.99, batch size of 128, and a learning rate of $5 * 10^{-4}$. The replay buffer size was set to 10^9. We have used the soft update with coefficient $\tau = 0.001$. Also, we employed ADAM optimizer [17]. Regarding the ϵ-greedy approach, we set the start value



(a) Maximum throughput vs DRL-RS achieved throughput

(b) DRL-RS throughput and latency SLA

Figure 2: Efficiency and Scalability performance

to 1, the ϵ-decay to 0.99, and the end value to 0.01. The model converges after 200 episodes.

B. Inference phase

To evaluate the model in a 5G environment, we used the multi-numerology 5G Simulator developed in [18] that relies on 6.1.4.2 of TS 38.214 [19] specifications to compute the Transport Block Size (TBS). For each Bandwidth size and ΔT, we trained an instance of the model using MCS 16, 3 UEs. Each UE is attached to one eMBB and one uRLLC slice with a fixed SLA. The sum of throughput SLA of UEs nearly equals the maximum achievable throughput in the corresponding bandwidth.

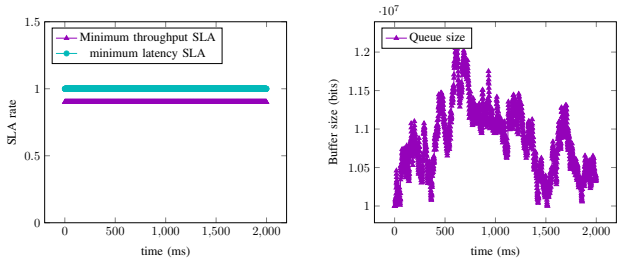
Regarding the scalability test of DRL-RS, we have varied the network parameters for each model instance (Table II). For example, we varied the MCS between 26 (good channel quality) and 16 (medium channel quality). We changed the number of slices per UE and scaled the number of UEs in the cell. We tested the DRL-RS agent on different bandwidth configurations from low bandwidth 20 Mhz to high bandwidth 400 Mhz, hence covering both FR1 ($< 6Ghz$) and FR2 ($> 6Ghz$) bands defined by 3GPP for 5G NR. It should be noted that 400 Mhz is the maximum bandwidth that can be used in 5G NR at the current time.

Finally, to test the efficiency of DRL-RS, we compared the achievable throughput by DRL-RS with the maximum throughput that can be achieved in each configuration.

Figure 2(a) compares the throughput obtained in each configuration (noted by DRL-RS in the figure) and the maximum throughput (i.e., theoretical) achievable in that configuration according to a 3GPP compliant tool [20] (noted by Maximum-value in the figure). We notice that the achieved throughput is close to the maximum achievable throughput, which means that DRL-RS is able to fill the resource grid efficiently while satisfying the latency SLA for all the slices . Figure 2(b) depicts the achieved SLA for each configuration listed in Table II. The y-axis represents the metrics used for measuring the SLA: $\max_i O_i^{thg}$, $\min_i O_i^{thg}$, $\text{average}_i O_i^{thg}$ for the throughput SLA (blue and red in the figure) and $\max_i O_i^{lat}$, $\min_i O_i^{lat}$, $\text{average}_i O_i^{lat}$ for the latency SLA (green and grey in the figure). We remind that these values are described in Section III-B1. The x-axis represents the configuration indexes summarized in Table II. Figure 2(b) reveals that the slice' SLAs are met for each configuration. We notice that the maximum and the minimum achieved SLA for throughput are greater than 1 ($\max_i O_i^{thg} > 1$) and 0.94 ($\min_i O_i^{thg} > 0.94$), respectively. This means that all the UEs have achieved their throughput SLA. We observe that our solution has a gap of 6% compared to the maximum achievable throughput due to the wasted resources that DRL-RS may generate. Indeed, these resources cannot be used by any UE due to shape and numerology constraints. We also notice that the latency SLAs are respected for all the configurations ($\min_i O_i^{lat} = 1$).

Table III: Traffic Simulation parameters

Parameter	Scenario 1	Scenario 2
Duration	2s	2s
eMBB slice packet size	1.1 Mb	0.4 Mb
2nd eMBB slice packet size	-	2 Kb
uRLLC slice packet size	1 Kb	50 b
average eMBB slice arrival rate	80 packets/s	80 packets/s
average 2nd eMBB slice arrival rate	-	800 packets/s
average uRLLC slice arrival rate	800 packets/s	800 packets/s



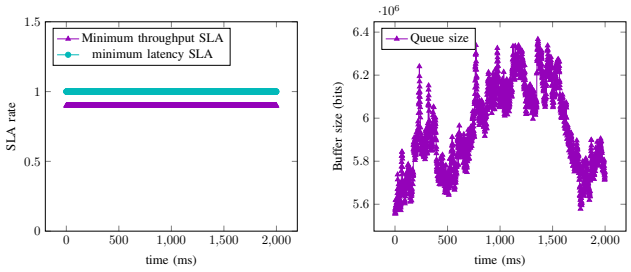
(a) Minumum SLA over time

(b) Average Buffer size over time

Figure 3: Scenario 1: traffic simulation using configuration 7

In Figures 3 and 4, we simulated traffic over 200 radio frames. The x-axis represents time in ms, while the y-axis is the SLA indicators ($\min_i O_i^{thg}$ and $\min_i O_i^{lat}$) for Figures 3(a) and 4(a), and the average transmission buffer size over UEs for Figures 3(b) and 4(b). The simulation parameters are presented in Table III. We leveraged the Poisson distribution

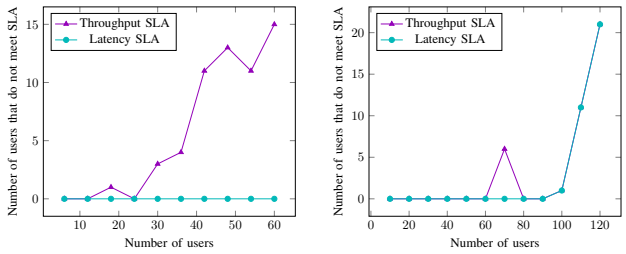
to generate traffic arrivals. The network configurations used in scenarios one and two correspond to configuration 7 and configuration 6 enumerated in Table II, respectively. Our objective behind these figures is to demonstrate the efficiency of DRL-RS over time, given that traffic arrives following a stochastic distribution. Indeed, we observe that the SLAs are met throughout the time (Figures 3(a), 4(a)), and the average buffers' size is not exploding, which validates that the DRL-RS allocation over time is able to satisfy the high traffic loads in larger bandwidths.



(a) Minumum SLA over time (b) Average Buffer size over time

Figure 4: Scenario 2: traffic simulation using configuration 6

In the last experimentation, we varied the number of UEs in configuration 4 and configuration 7. We focused on configuration 7 since it has the largest bandwidth in 5G NR. The results are shown in Figure 5. The x-axis represents the number of UEs in the system, while the y-axis represents the number of UEs that did not meet their SLAs. We notice that DRL-RS can support up to 100 UEs with a latency SLA of 0.125 ms in configuration 7, while in configuration 4, DRL-RS can meet all slices latency and handle up to 26 UEs while respecting their SLAs.



(a) Configuration 4 (b) Configuration 7

Figure 5: Performance over number of UEs

Conclusion

In this paper, we introduced DRL-RS, a novel radio resource scheduler based on DRL featuring Network Slicing in 5G NR. DRL-RS allows sharing radio resources efficiently to satisfy throughput and latency SLAs for a high number of UEs and large bandwidths in 5G NR. Simulation results clearly showed that DRL-RS is efficient and scalable for any configuration in 5G NR. Our future focus will be modeling the problem using optimization theory that will allow us to obtain the optimal assignment and hence be able to compare DRL-RS performances with an optimal configuration.

Acknowledgment

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program under the 5G!Drones project (Grant No. 857031).

References

[1] ITU-R. "Framework and overall objectives of the future development of IMT for 2020 and beyond". In: (2015).

[2] 3rd Generation Partnership Project (3GPP). "5G; Study on New Radio Access Technology; Radio Interface Protocol Aspects". In: *TR 38.804* Release 14 (2017).

[3] 3GPP. "Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS)". In: *TS 23.501* Release 16 (2021).

[4] Lei You et al. "Resource Optimization With Flexible Numerology and Frame Structure for Heterogeneous Services". In: *IEEE Communications Letters* (2018).

[5] Anique Akhtar and Hüseyin Arslan. "Downlink resource allocation and packet scheduling in multi-numerology wireless systems". In: *WCNCW*. 2018.

[6] Kandaraj Piamrat et al. "QoE-based network selection for multimedia users in IEEE 802.11 wireless networks". In: *LCN 2008, The 33rd IEEE Conference on Local Computer Networks, Montreal, Quebec, Canada*.

[7] Ljiljana Marijanović et al. "Optimal Resource Allocation with Flexible Numerology". In: *ICCS*. 2018.

[8] Jingxuan Zhang et al. "Machine Learning Based Flexible Transmission Time Interval Scheduling for eMBB and uRLLC Coexistence Scenario". In: *IEEE Access* (2019).

[9] Ljiljana Marijanovic, Stefan Schwarz, and Markus Rupp. "Multi-User Resource Allocation for Low Latency Communications Based on Mixed Numerology". In: *VTC2019-Fall*. 2019.

[10] Ljiljana Marijanovic, Stefan Schwarz, and Markus Rupp. "A Novel Optimization Method for Resource Allocation Based on Mixed Numerology". In: *ICC*. 2019.

[11] Yalcin Sadi, Serhat Erkucuk, and Erdal Panayirci. "Flexible Physical Layer based Resource Allocation for Machine Type Communications Towards 6G". In: *2nd 6G SUMMIT*. 2020.

[12] Faroq Al-Tam, Noélia Correia, and Jonathan Rodriguez. "Learn to Schedule (LEASCH): A Deep Reinforcement Learning Approach for Radio Resource Scheduling in the 5G MAC Layer". In: *IEEE Access* (2020).

[13] Marco Zambianco and Giacomo Verticale. "Spectrum Allocation for Network Slices with Inter-Numerology Interference using Deep Reinforcement Learning". In: *PIMRC*. 2020.

[14] N. C. Luong et al. "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey". In: *IEEE Communications Surveys Tutorials* (2019).

[15] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: (2013).

[16] Donald E. Knuth. *The Art of Computer Programming: Fascicle 3: Generating All Combinations and Partitions*. Vol. 4. 2005.

[17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[18] Karim Boutiba et al. "NRflex: Enforcing network slicing in 5G New Radio". In: *Computer Communications* (2021).

[19] 3GPP. "5G NR; Physical layer procedures for data". In: *TS 38.214* Release 15 (2018).

[20] 3GPP 5G tools. *5G NR Throughput calculator*. 2021. URL: https://5g-tools.com/5g-nr-throughput-calculator/. (accessed: 26.10.2021).