On using Deep Reinforcement Learning to balance Power Consumption and Latency in 5G NR

Karim Boutiba, Adlen Ksentini
EURECOM
Sophia Antipolis, France
karim.boutiba@eurecom.fr, adlen.ksentini@eurecom.fr

***Abstract*—Future generation cellular networks consider Power Consumption (PC) as a key concern in designing and operating wireless communication systems. In this context, 3GPP has proposed several techniques to reduce User Equipment (UE) PC, such as Connected-mode Discontinuous Reception (C-DRX), with a new set of parameters introduced by 5G New Radio (NR) and BandWidth Part (BWP) adaptation. However, they did not specify how to derive the C-DRX parameters and BWP configuration that reduce the PC while avoiding latency overflow. To address this shortcoming, we propose a novel solution to jointly derive the C-DRX parameters and the BWP configuration to find a trade-off between low PC and low latency. Given the inherent dynamics and uncertainty in wireless network environments, our solution relies on Deep Reinforcement Learning (DRL) to learn from the dynamic traffic pattern and derive the best C-DRX and BWP configuration that minimizes PC while achieving low latency. Simulation results demonstrate the effectiveness of the proposed methodology in reducing the PC (i.e., 50-95% power gain) while avoiding latency overflow for a different number of connected UEs (i.e., 1 to 20 UEs).**

I. Introduction

Next-generation cellular networks should accommodate the explosive growth in mobile data traffic and support different heterogeneous services to empower industry verticals and enable new business models. To achieve this goal, 5G New Radio (NR) has introduced several features to increase throughput and decrease latency, mainly numerology, massive Multiple Input and Multiple Output (mMIMO) and higher bandwidths. These features increase Power Consumption (PC) for both the network infrastructure and the User Equipment (UE). The latter, which are typically powered by a limited battery, can suffer from poor Quality of Experience (QoE) [1] due to the rapid discharge of the battery. The communications industry should therefore develop strategies to optimize the energy efficiency of 5G networks without compromising Quality of Service (QoS) [2]. Leading wireless equipment vendors have begun studying UE power-saving schemes in 5G NR [3]. They have investigated several techniques to reduce PC, such as Connected-mode Discontinuous Reception (C-DRX) with a new set of parameters introduced by 5G NR and BandWidth Part (BWP) adaptation. C-DRX allows UEs to periodically enter a sleep state during which the physical layer functions of UEs become inactive, and thus PC is reduced. However, the latency may increase as UEs may be in the sleep state when the data arrives at the gNodeB. On the other hand, BWP adaptation consists in reducing the BWP size based on the demand of UEs, which decreases the PC when UEs do not ask for high data rates. The 3GPP study, conducted in [3], shows the impact of C-DRX parameters and BWP adaptation on network performance, mainly on latency and power gain. However, the 5G NR standard does not provide solutions to derive the C-DRX parameters and the BWP configuration dynamically. To address this shortcoming, we introduce a Deep Reinforcement Learning (DRL)-based solution, called DRL-based Latency and Power optimizer (DRL-LP), to jointly derive the C-DRX parameters and the BWP configuration. The proposed solution is designed to be scalable in terms of the number of UEs and traffic patterns. Indeed, The DRL agent observes UEs' history of: (i) the experienced latency; (ii) the buffer status; and (iii) the number of scheduled UEs; during a time window. Then, for each UE, the DRL agent sets the C-DRX parameters and the BWP size for the next time window. To the best of our knowledge, no prior work has combined C-DRX and BWP adaptation to freduce the PC further while ensuring low latency.

The main contributions of this work are manifolds:

- We model the problem and the objective function to minimize the PC and the latency considering the C-DRX parameters and BWP adaptation.
- We introduce a DRL-based solution to jointly derive the C-DRX and BWP configuration and find a compromise between low PC and low latency. The C-DRX configuration includes new parameters introduced by 5G NR that help decrease the latency such as the C-DRX slot offset.
- The proposed DRL solution is designed to support multiple UEs, and different traffic patterns, i.e., the DRL agent is trained only once and then deployed regardless of the number of UEs and their traffic patterns.
- We evaluate the solution on periodic and aperiodic traffic for different numbers of UEs. The periodic traffic is characterized by random inter-arrival periods, and the aperiodic traffic follows the Poisson distribution with random parameters.

The rest of the paper is organized as follows: Section II describes the background on C-DRX and BWPs, and summarizes the related works on reducing the PC. Section III introduces the problem formulation. Our proposed solution is presented

in Section IV and evaluated in Section V. Finally, we conclude the paper in Section VI.

II. Background

A. C-DRX

Without C-DRX in 5G NR, a UE stays awake all the time to decode the downlink data. This consumes a large amount of the UE's power. The gNB configures the UE with a set of C-DRX parameters. These C-DRX parameters are selected based on the type of application to minimize PC. However, they may increase the latency because the UE may be in a C-DRX sleep state when the data arrives at the gNB, and the latter should wait for the UE to become active. When C-DRX is activated, the time is divided into cycles, which can be long or short. Each cycle consists of an ON period and an OFF period. The ON period, defined in terms of milliseconds, is the period during which the UE remains awake and decodes the downlink data. We consider the C-DRX parameters in 5G NR, depicted in Figure 1, to be (i) the cycle length; (ii) the duration of the ON period; (iii) the offset between the start of the cycle and the start of the ON period. The latter is a new parameter introduced by the 5G NR standard in order to shift the ON period since the arrival of the data is not always aligned with the cycle length, i.e., the data may arrive in the middle or at the end of the cycle. The ON period can be shifted to the middle or end of the cycle instead of leaving the UE active for the entire cycle, thus reducing latency while saving maximum power.

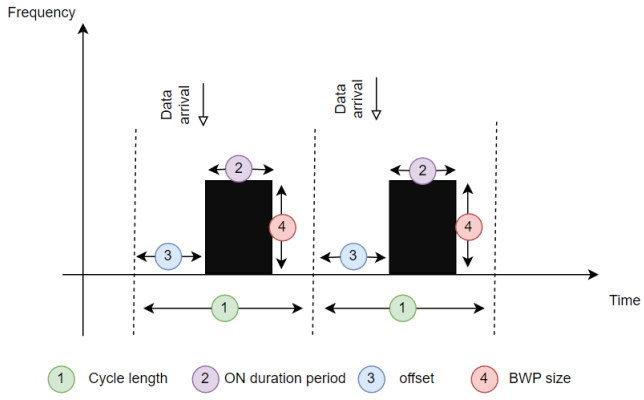

Figure 1: C-DRX parameters and BWP

B. BWP Adaptation

Higher bandwidths have a direct impact on peak data rates and QoE, but UEs do not always demand high data rates. Using high bandwidth can result in higher PC. In this regard, BWP was introduced by 5G NR to allow UEs to operate in bandwidths lower than the configured bandwidth, making NR an energy-efficient solution despite supporting the broadband operation. BWPs are contiguous subsets of PRBs allocated for a UE, meaning that the UE expects to use resources only in a specific portion of the bandwidth. The UE can be configured with up to 4 BWPs and can only use one at a time. Thus, adapting the size of the BWPs to the needs of the UE can help reduce PC.

C. Related work

In [4], a logic controller-based C-DRX system was proposed to adaptively adjust the C-DRX parameters by learning the information of historical delay time and the packet arrival rate. However, the authors only considered the C-DRX cycle length parameter and did not consider multiple UE scenarios that can impact the latency. Authors of [5] proposed using a DRL-based actor-critic algorithm to choose the C-DRX cycles based on traffic statistics and to use symmetric sampling to accelerate online learning. However, they only (i) considered C-DRX cycle length parameter; (ii) evaluated two scenarios (i.e. the traffic follows two Poisson distributions with fixed mean arrival rates). Authors of [6] presented a novel Contextual Bandit-based approach to optimize the C-DRX configuration of UEs in 5G NR. However, they did not consider the latency factor, which is strongly impacted by C-DRX. In [7], the authors leveraged channel capacity predictions to minimize the energy usage of UEs and create longer sleep opportunities while preventing video interruptions. However, authors only considered the C-DRX cycle length parameter without optimization for low-latency services. Authors of [8] designed a control policy to adjust the ON duration period parameter in order to satisfy eXtended Reality (XR) requirements and minimize PC. Authors of [9] developed a model to evaluate the impact of the BWP adaptation on power gains. However, they did not consider multiple UE scenarios since the scheduling opportunities in a BWP are limited per slot.

To the best of our knowledge, no prior work has combined C-DRX and BWP adaptation to reduce PC further. The choice of this combination (i.e., combining C-DRX and BWP) is motivated by the possibility of reducing PC by using the same narrow BWP for more UEs by shifting C-DRX cycles (i.e., making the UEs ON duration period not cross) and thus reduce their PC. In addition, most of the work on C-DRX has considered only one parameter to be optimized to make the problem easy to solve.

III. Network Model and Problem Formulation

We consider a network consisting of a set of UEs denoted $\mathcal{K}$ and sharing bandwidth of size $\mathcal{W}$. Let Δ, Φ, Γ and Ω be the sets of cycle length, ON periods, offsets and BWP sizes, respectively. Each UE $i \in \mathcal{K}$ may use a different C-DRX configuration and a different BWP during a set of $\mathcal{S}$ time slots. Let Δ_i^j, Φ_i^j, γ_i^j, and ω_i^j be Boolean decision variables that indicate whether the UE i uses a cycle length of $f(j)$, an ON period of $g(j)$ an offset of $h(j)$, and a BWP of size $l(j)$, respectively; $f(x)$, $g(x)$, $h(x)$, and $l(x)$ are functions that return the x^{th} element of Δ, Φ, Γ, Ω, respectively. Let $\mathcal{P}_{i,t}$ be a variable that measures the PC of UE i at time slot $t \in \mathcal{S}$. P^{max} is a constant that defines the maximum PC of a given UE in all time slots of $\mathcal{S}$. Let $\mathcal{L}_{i,t}$ a variable that measures the delay of the data arrived at slot t for UE i. If no

data has arrived for UE i at slot t, then $\mathcal{L}_{i,t} = -1$. $\mathcal{L}_i^{max}$ is a constant that defines the maximum latency allowed for UE i. Note that the latency is the waiting time between data arrival and transmission. Hence, $\mathcal{L}_{i,t}$ is influenced by the C-DRX parameters (i.e., data arrives during the sleep period and thus latency increases), the BWP size (i.e., the BWP size does not allow all data to be sent in one transmission and thus latency increases), and the packet arrival slot.

We can formulate the objective function as follows:

$$\min \sum_{i=1}^{|\mathcal{K}|} \sum_{t=1}^{|\mathcal{S}|} \alpha \times \frac{\mathcal{L}_{i,t}}{\mathcal{L}_i^{max}} + (1-\alpha) \times \frac{\beta_i \times \mathcal{P}_{i,t}}{\mathcal{P}^{max}} \quad (1)$$

S.t.

$$\forall i \in \mathcal{K}, \forall t \in \mathcal{S} : \mathcal{L}_{i,t} \geq -1, \mathcal{P}_{i,t} \geq 0 \quad (2)$$

Where α is a given constant ($0 \leq \alpha \leq 1$) that defines the priority between latency and PC. If $\alpha = 1$, then we are only interested in optimizing latency, whereas, if $\alpha = 0$, then we are only interested in optimizing PC. According to the energy model introduced by the 3GPP specification in [3], the relationship between the PC and the BWP size is linear and the scaling factor β_i is given by equation 3.

$$\forall i \in \mathcal{K} : \beta_i = 0.4 + 0.6 \times \frac{\left(\sum_{j=1}^{|\Omega|} \omega_i^j \times l(j) - 20\right)}{80} \quad (3)$$

Equation 4 ensures that each UE i has only one C-DRX configuration and one BWP during $\mathcal{S}$.

$$\forall i \in \mathcal{K}, \sum_{j=1}^{|\Omega|} \omega_i^j \leq 1, \sum_{j=1}^{|\Delta|} \Delta_i^j \leq 1, \sum_{j=1}^{|\Phi|} \Phi_i^j \leq 1, \sum_{j=1}^{|\Gamma|} \gamma_i^j \leq 1 \quad (4)$$

Equation 5 ensures that the sum of the BWP sizes does not exceed the available bandwidth $\mathcal{W}$ when the UEs ON periods cross. Let $\mathcal{B}_{i,t}^j$ be a Boolean system variable that indicates whether UE i is active (i.e., during the ON period) and uses a BWP of size $l(j)$ at time slot t.

$$\forall t \in \mathcal{S}, \sum_{i=1}^{|\mathcal{K}|} \sum_{j=1}^{|\Omega|} \mathcal{B}_{i,t}^j \times l(j) \leq \mathcal{W} \quad (5)$$

Each BWP has a limited amount of scheduling opportunities (i.e. Downlink Control Information (DCI)) per slot. Let $\mathcal{K}_j^{max}$ a constant that defines the maximum number of UEs allowed per slot for the BWP j. Equation 6 ensures that for each BWP, the number of UEs scheduled per slot does not exceed the maximum number of DCIs per slot, in each BWP.

$$\forall t \in \mathcal{S}, \forall j \in \Omega \sum_{i=1}^{|\mathcal{K}|} \mathcal{B}_{i,t}^j \leq \mathcal{K}_j^{max} \quad (6)$$

Equation 7 ensures that $\mathcal{P}_{i,t} = 0$ when UE i is in sleep mode at slot t. Otherwise $\mathcal{P}_{i,t}$ takes a positive value, since we minimize the sum of $\mathcal{P}_{i,t}$. It is sufficient to set $\mathcal{P}_{i,t} = 1$ when UE i is awake at slot t. In addition, the equation 7 allows us to define the system Boolean variable $\mathcal{B}_{i,t}^j$ used in equations 5 and 6. Let $\mathcal{N}$ denotes a subset of integers in range 0 to the maximum number of cycles in the system (i.e. $|\mathcal{S}| \div \min_j f(j)$).

$$\begin{gathered}
\forall i \in \mathcal{K}, \forall t \in \mathcal{S}, \forall k \in \mathcal{N} : \\
if\ k \times \sum_{j=1}^{|\Delta|} \Delta_i^j \times f(j) + \sum_{j=1}^{|\Phi|} \Phi_i^j \times g(j) \leq t \\
And \\
t \leq k \times \sum_{j=1}^{|\Delta|} \Delta_i^j \times f(j) + \sum_{j=1}^{|\Phi|} \Phi_i^j \times g(j) + \sum_{j=1}^{|\Gamma|} \gamma_i^j \times h(j) \\
then \\
\mathcal{P}_{i,t} = 1,\ \mathcal{B}_{i,t}^j = \begin{cases} 1 & \text{if } \gamma_i^j = 1\ \forall j \in (1..|\Omega|) \\ 0 & \text{otherwise} \end{cases} \\
else \\
\mathcal{P}_{i,t} = 0,\ \mathcal{B}_{i,t}^j = 0\ \forall j \in (1..|\Omega|)
\end{gathered} \quad (7)$$

Unfortunately, we cannot use the optimization problem mentioned above, mainly because the arrival of traffic is unknown and it is difficult to predict it and hence the variable $\mathcal{L}_{i,t}$ is difficult, if not impossible, to compute. Here the problem can be transformed into a linear problem, but since we can not solve it, it is not worthy doingso.

IV. Deep Reinforcement Learning-based Latency and Power optimizer (DRL-LP)

As aforementioned, it is hard to solve the optimization problem efficiently and without prior knowledge of the traffic patterns. For this reason, we propose the DRL-LP framework that leverages DRL. The DRL hides the complexity and the stochastic nature of the environment. It also helps the DRL-LP framework to make efficient and quick decisions that adapt according to the traffic patterns. Moreover, DRL-LP gains the ability to learn with time and adapts to different and unseen situations. In the balance of this section, we will present DRL and DRL-LP overview followed by a detailed description of DRL-LP.

A. DRL Overview

Machine Learning (ML) plays an important role in 5G Networks and Beyond. Particularly DRL, a ML technique that can be used without the need for data sets. DRL can be leveraged to derive configurations or management decisions in real-time [10] (i.e., less than $1ms$) in a stochastic environment, which makes it suitable for the Radio Access Networks (RAN) domain. DRL can provide self-configure and self-optimized network functions, such as radio resource allocation [11]. 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 the 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^*. Accordingly, the ability of DRL to derive good decisions quickly, deal with unseen environments, and be scalable make it suitable for solving the joint latency and PC minimization problem in 5G NR.

B. DRL-LP Overview

DRL-LP periodically loops over the UEs in $\mathcal{K}$. For each UE i, DRL-LP captures an observation, then applies a configuration (i.e., C-DRX parameters and BWP size) and finally gets a reward after applying the configuration during $\mathcal{S}$ slots. We have designed the DRL-LP agent to ensure generality and then work in an unseen environment. The DRL-LP agent has been designed to work independently from the number of UEs and the traffic pattern. In what follows, we define the elements of the DRL-LP agent, including the state, the reward, and the action.

i) State: The DRL-LP agent captures an observation, per UE i, composed of four parts: $\{\mathcal{L}_i, \mathcal{B}_i, \mathcal{C}_i, \mathcal{U}\}$. The first three parts are specific for UE i, while the fourth part is common between all UEs. $\mathcal{L}_i$ is the history of latency $\mathcal{L}_{i,t}$ experienced by UE i during the past $|\mathcal{S}|$ slots. For each slot t, if arrived data $\mathcal{L}_{i,t}$ represents the time delay between t and the data transmission, else if no data arrived in slot t then $\mathcal{L}_{i,t} = -1$. $\mathcal{B}_i$ is the history of buffer status of UE i during the past $\mathcal{S}$ slots. $\mathcal{C}_i$ summarizes the action applied to the environment (i.e., both the C-DRX parameters and the BWP size). Each element $\mathcal{C}_{i,t}$ of $\mathcal{C}_i$ equals to the BWP size if UE i is awake at slot t, else it equals to 0. $\mathcal{U}$ is an array of $\mathcal{U}_t$ that represents the number of scheduled UEs for each slot t.

ii) Action: The DRL-LP agent has 4 discrete actions: (δ_i, ϕ_i, γ_i, ω_i). For each UE i, δ_i is the C-DRX cycle length, ϕ_i is the ON period, γ_i is the offset between the start of the ON period and the start of the cycle, and ω_i is the size of the BWP to configure. Accordingly, The C-DRX configuration (δ_i, ϕ_i, γ_i) is applied for UE i and BWP of size ω_i is configured for UE i.

iii) Reward: We have adapted an episodic approach, whereby each episode runs for fixed number of steps. For each episode, a randomly selected traffic pattern is applied. The reward r_t has been defined as follows:

$$r_t = \alpha \times \left(1 - \frac{\max_t \mathcal{L}_{i,t}}{\mathcal{L}^{max}}\right) + (1-\alpha) \times \left(1 - \frac{\left(0.4 \ + \ 0.6 \ \times \frac{(\omega_i \ - \ 20)}{80}\right) \times \mathcal{P}_i}{|\mathcal{S}|}\right)$$

Where $\max_t \mathcal{L}_{i,t}$ is the maximum latency for UE i during the past time window and $\mathcal{P}_i$ is the number of slots in which UE i was awake. $\mathcal{L}^{max}$ is the maximum latency in the system. α is a given constant ($0 \leq \alpha \leq 1$) that defines the priority between latency and PC. The agent gets a higher reward whenever the maximum latency gets smaller or the sleep period is larger while using a smaller BWP.

C. DRL-LP detailed description

DRL-LP leverages the Deep Q-Network (DQN) algorithm with local and target networks, which is one of the most efficient DRL algorithms for continuous state space and discrete actions. We have tried the A2C [12] and the Actor-Critic using Kronecker-Factored Trust Region (ACKTR) [13] algorithms but the exploration phase was not efficient for DRL-LP environment as the agent was not able to converge. DRL-LP executes two steps: decision-making and updating the Q-Networks. We used two networks: a local Q-Network and a target Q-Network. The target network is the same as the local network, except that its parameters are updated every τ^{-1} step. They are combined to help the convergence and stabilization of the learning.

1) Decision making: DRL-LP agent observes a state and feeds it to the local Q-Network to get a discrete action distribution. Then, an ϵ-greedy approach is applied to choose an action from each distribution, which means DRL-LP 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. ϵ will decrease over time during the learning pushing the agent to explore the environment at the beginning of the training and driving 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 [14], the parameters of the local Q-Network are optimized at every step by considering the local and target values. In contrast, the parameters of the target Q-Network are updated every τ^{-1} step to stabilize the algorithm's convergence.

V. Performance Evaluation

In the balance of this section, we will introduce the simulation environment and parameters used for training DRL-LP agent. Then, we will evaluate the trained agent in a 5G simulated environment.

A. Simulation parameters and training phase

We have trained the DRL-LP agent using 3000 independent episodes. In each episode, the traffic pattern is selected randomly among two traffic pattern categories with different parameters: (i) periodic arrival rate with a period $\lambda_p \in \{10, 20, 50, 100, 200\}$ (ii) aperiodic traffic that follows a Poisson distribution with the mean arrival rate $\lambda_a \in \{1/10, 1/20, 1/50, 1/100\}$. For the periodic traffic, we add an initial offset o_{init} selected randomly, such as o_{init}, a positive integer smaller than the period. The goal behind o_{init} is avoiding data arrivals aligned with C-DRX cycles, which

makes the simulation more realistic. The data size distribution is selected randomly among (i) a fixed data size; (ii) a Poisson distribution data size. For both distributions, the mean data size is selected randomly among $\in \{10^3, 10^6\}$ Bytes in each episode. The number of steps in each episode is equal to 100 steps. In each step, the simulation runs for 100 slots. We used numerology 0 wherein 1 slot is 1 ms. We have trained the agent using 12 UEs. The set of cycle lengths is $\{10, 20, 50, 100\}$. The set of ON periods is $\{3, 5, 10\}$. The set of offsets is $\{0, 0.5 \times T_c\}$, such as T_c is the cycle length. The set of BWPs is $\{20, 50\}$. Each BWP has a limited amount of scheduling opportunities (i.e., DCI). We assume an aggregation level of 2, meaning that the maximum number of scheduled UEs per slot is 4 and 10 for the 20 MHz and 50 MHz BWP, respectively.

Table I: DRL-LP parameters

Parameter	Value
α	0.5
$\mathcal{L}^{max}$	3 ms
Number of hidden layers	2
Hidden layer size	128 nodes
Discount factor γ	0.99
Batch size	256
Learning rate	$5 * 10^{-4}$
Replay buffer size	10^9
Soft update coefficient τ	0.001
Optimizer	ADAM [14]
ϵ-start	1
ϵ-decay	0.998
ϵ-end	0.01
Number of training episodes	3000

The considered parameters of the DRL-LP agent are presented in Table I. To evaluate DRL-LP in a 5G environment, we extended the 5G system level simulator developed in [15] to support C-DRX operations. We have implemented our simulation environment using Python and Pytorch library. We have used a machine with 32 CPUs, an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz (2.7 GHz with Turbo Boost technology), and 128 GB of RAM.

Figure 2 depicts the convergence evaluation of the DRL-LP agent during training. The x-axis represents the episodes, while y-axis represents the score (sum of rewards during an episode) averaged every 100 episodes. We observe that the DRL-LP agent converges after 2000 episodes since the curve tangents tend toward 0.

B. Inference phase

We evaluated the DRL-LP framework in terms of: i) Latency; ii) PC; iii) Number of UEs. We ran the simulation for 4000 slots. We compared DRL-LP with static C-DRX configurations and without C-DRX. We used the same traffic pattern selection mechanism as the training phase. Configuration A and configuration B denote the static configuration (cycle length: 10ms, ON period: 5ms, offset: 0ms and BWP: 50 MHz) and (cycle length: 100ms, ON period: 10ms, offset: 0ms and BWP: 20 MHz) respectively.

In Figure 3, the x-axis represents the latency (in ms), which is measured by $t_t^c - t_a^c$, where t_a^c is the arrival time of a c data block and t_t^c is the transmission time of the last part of the c data block. The y-axis represents the cumulative distribution function (CDF) of the latency collected by running the simulations for 4000 ms. We observe that without C-DRX, the latency is less than 2 ms because the UE is always on, and the data is scheduled directly on arrival. In configuration A, the latency is less than 5 ms because the maximum sleep time of a UE is 5 ms, and in the 50 MHz BWP up to 10 UE can be scheduled per slot. While for configuration B, more than 50% of the samples have latency greater than 50 ms, and 10% have latency greater than 150 ms because the sleep time is higher (up to 90 ms) and the BWP does not allow more than 4 UEs to be scheduled per slot. We note that for DRL-LP, 78% of the samples have latency less than 5 ms, which is better than configuration A, and 90% less than 25 ms, which is better than configuration B. We conclude that DRL-LP is able to avoid latency overflow while dynamically changing the configuration of C-DRX and BWP.

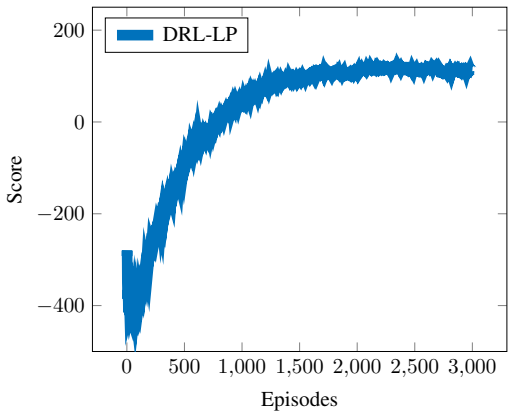

Figure 2: Convergence evaluation of DRL-LP agent during the training mode

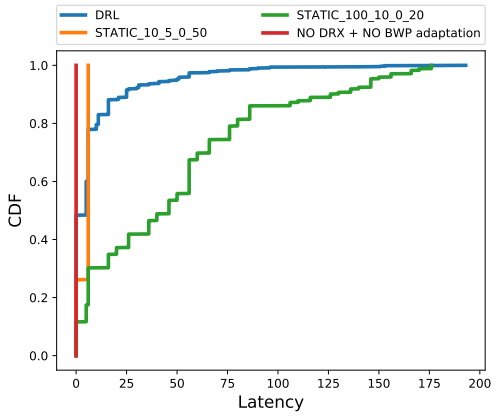

Figure 3: CDF of the Latency during inference mode

In figure 4, the x-axis represents the PC of a UE, calculated using the energy model in [3]. The y-axis represents the CDF of the PC collected at the same time as the latency in Figure 3. We observe that without C-DRX, the PC is the highest because the UEs are awake all the time. While in configuration A, we

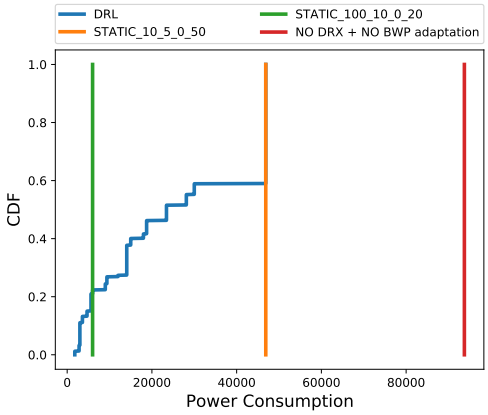

Figure 4: CDF of the PC during inference mode

observe that all UEs achieve a gain of about 50% because all UEs are sleeping for half of the cycle. In Configuration B, the power gain is higher (90%) because the UEs sleep for 90% of the time. We note that 20% of the DRL-LP samples have a lower PC than Configuration B, which means a power gain of over 90%. In addition, all DRL-LP samples achieve a lower PC than configuration A (i.e., a power gain of more than 50%). We conclude that DRL-LP achieves a good balance between PC and latency (i.e., it achieves more than 50% power gain while maintaining less than 5 ms latency).

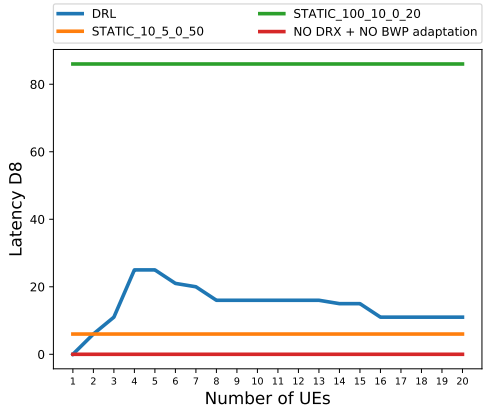

Figure 5: The 8th Decile of Latency during inference mode

In Figure 5, the x-axis represents the number of UEs and the y-axis represents the 8th decile (i.e., 80% of the samples have a value less than this) of the samples collected during the 4000 slots. We observe that DRL-LP is able to keep the latency below 25ms. We note that DRL-LP achieves better latency (i.e., less than 20ms) when run on more than 8 UEs. We justify this by the nature of DRL, which consists of approximating continuous states using neural networks, making it more biased to the observed states, and by the fact that the agent was trained on 12 UEs, which makes it more biased for larger number of UEs.

Conclusion

This paper introduced DRL-LP, a Deep Learning Reinforcement (DRL)-based solution to balance Power Consumption (PC) and latency in 5G NR. DRL-LP will be used by the 5G base station to derive the C-DRX and BWP configuration per UE. The simulation results clearly showed that DRL-LP is able to find a trade-off between latency (i.e., achieve latency less than 5ms) and PC (i.e., achieve power gain more than 50%). Our future goal is to implement DRL-LP on OpenAirInterface (OAI) 5G platform to validate real use cases.

Acknowledgment

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program MonB5G project (Grant No. 871780).

References

[1] Tarik Taleb and Adlen Ksentini. "QoS/QoE predictions-based admission control for femto communications". In: *Proc. of IEEE International Conference on Communications, ICC 2012, Ottawa, ON, Canada, June 10-15*. IEEE, 2012.

[2] Pantelis A. Frangoudis et al. "An architecture for on-demand service deployment over a telco CDN". In: *Proc. of ICC 2016, Kuala Lumpur, Malaysia, May 22-27*. IEEE, 2016.

[3] 3GPP. "Study on User Equipment (UE) power saving in NR". In: *TR 38.840* Release 16 (2019).

[4] Ziyang Zhang and Xin Zhang. "Logic Controller-Based Discontinuous Reception (DRX) System for NR". In: *Journal of Physics: Conference Series* (2021).

[5] JianHong Zhou et al. "Actor-Critic Algorithm Based Discontinuous Reception (DRX) for Machine-Type Communications". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018.

[6] Philipp Bruhn and German Bassi. "Machine Learning Based C-DRX Configuration Optimization for 5G". In: *Mobile Communication - Technologies and Applications; 25th ITG-Symposium*. 2021.

[7] Farnaz Moradi et al. "Flexible DRX Optimization for LTE and 5G". In: *IEEE Transactions on Vehicular Technology* (2020).

[8] Stefano Paris, Klaus Pedersen, and Qiyang Zhao. "Adaptive Discontinuous Reception in 5G Advanced for Extended Reality Applications". In: *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. 2022.

[9] Venkatesh Ramaswamy, Jeffrey T. Correia, and Darcy Swain-Walsh. "Analytical Evaluation of Bandwidth Part Adaptation in 5G New Radio". In: *2021 IEEE PIMRC*. 2021.

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

[11] Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. "Radio resource management in multi-numerology 5G new radio featuring network slicing". In: *ICC 2022*. Ed. by IEEE. Seoul, 2022.

[12] Volodymyr Mnih and al. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. 2016.

[13] Yuhuai Wu et al. "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation". In: *Advances in neural information processing systems* (2017).

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

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