Advanced sleep modes in 5G multiple base stations using non-cooperative multi-agent reinforcement learning

Amal Abdel Razzac, Tijani Chahed, Zahi Shamseddine, Wafik Zahwa

Advanced Sleep Modes in 5G Multiple Base Stations using Non-cooperative Multi-Agent Reinforcement Learning

Amal Abdel Razzac[1], Tijani Chahed[2], Zahi Shamseddine[1] and Wafik Zahwa[1]
[1] *Lebanese University, Rafic Hariri University Campus, Hadath, Lebanon*
[2] *Institut Polytechnique de Paris, Télécom SudParis, SAMOVAR, 19 Place Marguerite Perey, 91120 Palaiseau, France*

***Abstract*—We consider in this paper multiple 5G base stations (BSs) implementing Advanced Sleep Modes (ASM) wherein each base station is able to deactivate some of its components when it does not transport any traffic and save thus energy. Thanks to so-called lean carrier, ASM define four levels of sleep, the deeper the level the larger the energy gain but the more delay to wake-up and serve the incoming user. We specifically study this energy saving versus delay performance trade-off taking into account the effect of inter-cell interference and its impact on whether to wake-up and serve the transmission request immediately upon arrival or to continue to sleep; this latter decision is a main novelty of our work. We treat the case where arrivals of those requests are unknown and a reinforcement learning agent is implemented in each BS in order to (selfishly) derive the optimal sleep policy that achieves a target energy saving versus delay performance trade-off. Our results show the optimal policies in terms of the value of the timer after which the BS goes into sleep, the time spent in each sleep level, and whether the BS should continue to sleep or wake up immediately upon request arrival. We eventually show the corresponding achieved power saving and delay performance.**

***Index Terms*—5G, multiple base stations, Advanced Sleep Modes, Multi-Agent Reinforcement Learning, energy saving versus delay performance trade-off.**

I. Introduction

The fifth generation (5G) and beyond mobile systems promise the delivery of a wide range of revolutionary services. Yet, the predicted increase in traffic volume and the diversity of connected devices point to worrying levels of power consumption [1]. Finding techniques to alleviate the tendency of energy increase has become a major objective for both information and communications technology standardization sectors and network operators. Thereupon, energy efficiency is included as a key performance indicator: 3GPP's 5G specification calls for a 90% reduction in energy consumption as compared to 2010 [2]. Accordingly, various energy-related solutions targeting the reduction of power consumption in the wireless networks are under study, ranging from the deployment of small cells, the use of Multiple Input Multiple Output antennas, to Cloud Radio Access Network (Cloud-RAN) and Edge Computing.

We focus in this work on another such effort, namely the use of Advanced Sleep Mode (ASM) strategy [3] at the base station (BS). Our motivation is based on the fact that the RAN has the lion's share in the predicted global energy consumption, on the order of 50.6% by 2025 [1], with the highest energy consumption experienced at its end, namely at the power amplifier and antenna interface [2]. Hence, turning-off the BS, or at least its radio frequency component, in periods of inactivity, would enable to reduce power consumption of wireless sites. The conventional single-level sleep mode, namely the ON/OFF switching technique, completely turns off under-utilized BS at the expense of long delay needed to reactivate them which results in a negative impact on the users quality of service in terms of delay performance. ASM, on the contrary, correspond to a progressive deactivation of the components of the BS among four sleep levels, ranging from deep to light, allowing thus the implementation of flexible sleep policies that enable targeting fine trade-off between energy saving versus delay performance [4]. ASM have been made possible in the context of 5G thanks to the so-called lean carrier [5].

The challenge in finding the optimal sleep policy for a targeted energy saving versus delay performance is exacerbated by two uncertainties: the arriving time of the requests as well as the randomness in the channel quality and hence the service rate. Indeed, an increased waiting delay due to a sleep policy may be bearable by the arriving packet if it is counter-balanced by a short service time, while no waiting delays can be tolerated if they are to be added to an already long service duration. This channel quality and hence service capacity is also subject to interference from neighboring cells, which results from their state, i.e., if they are active, idle or even in sleep. We hence propose the use of Multi-Agent Reinforcement Learning (MARL), and in particular the non cooperative approach [6] as the BSs are most probably unaware of each other's state, in order to learn these unknowns and derive the optimal sleep policy in the context of multiple BSs, each equipped with ASM capability and RL agent.

There has been several works addressing ASM following the work in [4] which introduced ASM, with four sleep levels, and indicated their durations as well as power consumption for the case of a 2020 technology. In [7], the authors consider a single BS and study the power saving versus delay constraints for several, a priori known, traffic scenarios and loads. In [8] the authors consider a single BS as well and model its power consumption as a function of the sleep depth using a queuing model with vacations. Here too, traffic and sleep

profiles are supposed to be known in advance. The authors in [9] revisit the ASM model proposed in [4] with a more conservative approach for the power model emphasizing that some deactivated components will need longer times than assumed in the latter in order to wake-up and serve users. They conduct their simulations considering one macro BS and several micro BSs setting. In [10], multiple BSs are considered and the sleep policy is investigated via simulations for different network loads. An analytical model is developed to quantify the system level energy saving gains of a BS assuming low load, single transmission at a time with high probability and fixed time between data arrivals. In [11], the authors consider multiple small BSs and make use of prediction for the vacancy versus operational periods of the system. In [12], the authors propose a scalable, Markov Decision Process (MDP)-based model, to derive the optimal sleep policy targeting a given energy saving versus delay performance trade-off for a single BS. In case the next user arrival is unknown, this approach is extended to a Reinforcement Learning (Q-learning) one in [3] in order to pursue the same objective. The authors in [13] consider a single BS and derive the optimal power saving versus delay, via Reinforcement Learning (SARSA), depending on traffic load, in an online fashion. The work in [14] considers multiple BSs and studies the power saving versus delay trade-off using a distributed Q-learning approach, in an online, uncoordinated fashion.

All these works assume that when the network is in sleep, and a new request arrives, the BS wakes up immediately in order to serve it. To the best of our knowledge, none of the works tackling ASM in the literature have considered the possibility that the base station, if in sleep, does not systematically wake-up immediately upon the arrival of a new request but may continue to sleep depending on the targeted trade-off between energy saving versus delay performance which depends on the effect of channel quality as represented by the channel quality indicator (CQI) of this arriving request. We believe that this newly introduced decision point is important, especially in a multi-cell scenario with inter-cell interference that can dictate the deferral of the activation of the BS so as not to sacrifice the gain in power consumption, achieved when remaining in sleep, for an otherwise long service time due to interference and hence delay. This new decision point for the BS is a new feature that we implement in our present work in addition to the original decision point ruling its ASM profile.

The rest of the paper is organized as follows. In section II, we provide an overview of the ASM feature and our proposed sleep and wake-up strategy including the two coupled decisions to be taken by the BS. In section III we describe our multi-agent Q-learning algorithm used to derive the optimal sleep and wake-up policy. Our numerical results are presented and discussed in section IV. We conclude our work in section V and give some hints on future works in this context.

II. System and ASM model and strategy

We consider a network composed of multiple adjacent BSs, each equipped with ASM feature. When a BS finds itself idle, i.e., no user traffic, it starts a timer. If a transmission request arrives before the expiration of this timer duration, the BS serves the arriving packets and restarts the idle mode timer upon the end of transmission of the last one. Otherwise, it moves to a chosen sleep state.

ASM defines four levels of sleep [4] ranging from the lighter one, SM_1, to the deepest, SM_4. They differ in terms of the components that are deactivated and activated, which necessitate different durations and imply different depths of sleep and hence different savings in power consumption. Table I shows the durations for deactivation at each sleep level (which we assume to be equal to the activation time).

TABLE I: Advanced Sleep Modes

SM_1	SM_2	SM_3	SM_4
35,7 μs	0,5ms	5ms	0,5s

These ASM features are made possible within so-called lean carrier [5] in 5G networks which enables grouping and spacing of signaling which makes it possible to have sleep periods beyond the very short SM_1 which is the only one supported in 4G, due to frequent signaling, on the order of 1ms. In 5G however, signaling can be made as spaced out as 160 ms which enables the use of SM 1 through 3.

We implement the ASM strategy proposed in [3] according to which the sleep period starts with the deepest sleep level SM_3 and the BS wakes up gradually to lighter sleep levels until being fully activated. This process repeats upon idle timer expiration. The sleep profile is defined as the idle timer duration, denoted by T_I, and the duration during which the BS remains in each sleep mode SM_i, denoted by T_{SM_i}.

We denote by $\Delta_{D_{i\to j}}$ and $\Delta_{A_{i\to j}}$, respectively, the activation and deactivation duration to switch from operational state i to state j where i and j take values in $\{\text{Idle}, \text{SM}_1, \text{SM}_2, \text{SM}_3\}$. They are given by:

$$\Delta_{D_{i\to j}} = \Delta_{A_{i\to j}} = \Delta_{D_{I\to j}} - \Delta_{D_{I\to j}} \tag{1}$$

where $\Delta_{D_{I\to i}}$ is the duration needed to deactivate the components associated with SM_i indicated in Table I.

The delay experienced by each served packet in this model is actually composed of two parts: i. the waiting delay, which we denote by D_w, and which represents the time an arriving packet has to wait before the BS starts to serve it, it may in turn incorporate two components: the waiting delay until the activation of the BS, if the packet arrival happens during the sleep period, and the waiting delay until the service of all the packets that are served ahead of the tracked one, and ii. the service delay, denoted by D_s, is the time needed to serve the considered packet. The latter delay component is equal to the packet length divided by the bitrate offered to the user, which we denote by R_u: $R_u = W \times \text{SE}(\text{SINR}_u)$ where W is the bandwidth measured in hertz and $\text{SE}(\text{SINR}_u)$ is an increasing function that maps the achieved spectral efficiency, evaluated in bits per second per hertz, to the received signal to interference and noise ratio. Indeed, this latter parameter which we denote by SINR_u, is known at the BS via the Channel

Quality Indicator (CQI) sent by the user to the BS, and according to which this latter chooses the optimal modulation and coding scheme for this connection [16], which corresponds to an achieved spectral efficiency [17]. SINR_u depends on the states of the BSs:

$$\text{SINR}_{\text{u}}(t) = \frac{P^{\text{BS}_x} G_{\text{u}}^{\text{BS}_x}}{\sigma_n + \sum_{y \neq x} P^{\text{BS}_y} G_{\text{u}}^{\text{BS}_y}} \tag{2}$$

with P^{BS_x} and P^{BS_y} respectively the transmission power of BSs x and y, $G_{\text{u}}^{\text{BS}_x}$ and $G_{\text{u}}^{\text{BS}_y}$ respectively the channel gains between the user and BSs x and y and σ_n the Gaussian additive white noise.

Furthermore, and as stated above, we implement an additional decision point upon the detection of the first arrival during the sleep period wherein the BS, which is aware of the quality of the channel of the requesting connection via its CQI (as explained above), can choose to wake-up immediately or to wait till the end of the initially chosen sleep profile.

Note that all packets arriving during the activation period or during the sleep one if the BS chooses to continue to sleep, are buffered and served once the BS is fully activated again.

In summary, each BS has two decisions to take: i. at the beginning of its idle period, it decides the sleep profile composed of $(T_{\text{I}}, T_{\text{SM}_3}, T_{\text{SM}_2}, T_{\text{SM}_1})$ and ii. upon the arrival of transmission request during the considered sleep period, it decides to immediately activate or continue to sleep until the end of the sleep profile. These decisions depend on the pursued objective in terms of power saving versus delay performance. The difficulty in deriving the optimal policy comes, as previously argued, from the randomness and unpredictability of traffic arrivals and inter-cell interference levels. To this end, we propose to tackle the problem using MARL approach, as will be detailed in the next section.

III. MARL Formulation

We develop a model-free MARL framework based on the off-policy algorithm Q-learning [15] according to which the agent/learner observes its environment, chooses an action from the available action set and observes a numerical feedback that describes how good or bad the decision of taking the chosen action was, given the environment state. We refer to this numerical function as the cost function, denoted by $c(s, a)$, resulting from taking action a when the learner is at state s. Based on this feedback, the learning agent will improve its decisions regarding the choice of the actions in the future states. This mechanism is repeated until convergence providing the optimal action for each state and hence the optimal policy.

The parameters for this framework are defined next.

A. Agents and states set

We have multiple independent learners, which are the BSs. Each BS seeks the detection of the two following cell states: the beginning of the idle period and the CQI of the first arriving transmission request during the sleep period. We refer to these states by:

- state *Start-Idle*
- states (n, CQI_k) detected upon the first request arrival during sleep profile n when the CQI of this latter is CQI_k with index k taking values out of the 16 possible CQI indices defined for 5G[1]. Hence, for each sleep profile n out of the N_{SP} potential sleep profiles defined in the next subsection we can have 16 possible (n, CQI_k) states.

Overall, we have $1 + 16 \times N_{\text{SP}}$ states.

B. Decision instants and actions set

The instants at which the BS detects each of the previously introduced states defines the decision instants. The action space refers to the set of possible operation measures the BS can take at each of these time instants.

When the state *Start-Idle* is detected by the BS, this latter has to choose the optimal sleep profile $n \equiv (T_{\text{I}}, T_{\text{SM}_3}, T_{\text{SM}_2}, T_{\text{SM}_1})$ introduced in section II. In the numerical applications, we shall discretize and limit the set of values which these continuous time variables can take so as to have a manageable action set, denoted by $\mathbf{A}_{\text{Start-Idle}}$. The size of this action set is $N_{\text{SP}} = \Pi_i N_i$ sleep profiles where N_i is the number of possible values that T_i can take, $\forall i \in \{\text{I}, \text{SM}_1, \text{SM}_2, \text{SM}_3\}$.

On the other side, the set of possible actions the BS can follow when it is in state (n, CQI_k) are:

$$\mathbf{A}_{(\mathbf{n},\ \mathbf{CQI}_k)} \equiv \{\textit{immediate wake-up}, \textit{stick to sleep profile}\} \tag{3}$$

C. State-transitions and cost function

The action a BS chooses at a given state dictates both immediate cost that arises from taking this action at the given state as well as the next state it transitions to, and hence the long-run accumulated costs.

A BS in the idle state and adhering to a sleep profile $n \equiv (T_{\text{I}}, T_{\text{SM}_3}, T_{\text{SM}_2}, T_{\text{SM}_1})$ returns to this idle state if a transmission request is detected before the expiration of the idle timer T_{I}; the transmission requests are served first and the BS goes back to the state *Start-Idle*. The transition from *Start-Idle* to *Start-Idle* is also possible if no transmission requests arrive before the end of the total sleep duration. Otherwise, the learning agent moves to the state (n, CQI_k) upon the arrival of a packet transmission request with channel indicator CQI_k during sleep. Likewise, a transition from state (n, CQI_k) to the state *Start-Idle* occurs upon the wake-up of the BS either due to an immediate wake-up decision or due to the end of the sleep duration. Again, the buffered transmission requests, if any, are served upon these triggering events.

The cost function $c(s, a)$ should assess the power consumption at the BS versus the experienced delay for possible actions $a \in \{\mathbf{A}_{\textit{Start-Idle}}, \mathbf{A}_{(\mathbf{n},\ \mathbf{CQI}_k)}\}$ for each state $s \in \{\textit{Start-Idle},\ (n, CQI_k)\}$:

$$\begin{aligned} c(s,a) =& (1-\beta)\left(\frac{\sum_{i\in\{I,\text{SM}_1,\text{SM}_2,\text{SM}_3\}} P_i \times D_i}{\sum_{i\in\{I,\text{SM}_1,\text{SM}_2,\text{SM}_3\}} D_i}\right) \\ &+ \beta \max_{\forall \text{pkt}} \Big[D_w(a, \text{pkt}) + D_s(\text{pkt}) \Big] \end{aligned} \tag{4}$$

[1]The CQI index in 5G has 16 possible values which range from 0 to 15 [17]. 0 indicates the worst channel quality and 15 indicates the best one.

where

- β is a priority factor given to delay over power.
- P_i is the power in Watts consumed by the BS when it is in the operational mode $i \in \{I, \text{SM}_1, \text{SM}_2, \text{SM}_3\}$.
- D_i is the duration in seconds during which the BS consumes power P_i. Assuming that the power consumption during the transition from an operational mode i to the directly less consuming mode and vice versa, is equal to the power consumed at state i. Hence, D_i is equal to the summation of T_i and $2 \times \Delta_{D_{i\to j}}$, which are respectively the duration during which the BS remains in level $i \in \{I, \text{SM}_1, \text{SM}_2, \text{SM}_3\}$ and the time needed to perform the transition from this level to the directly lower level $j \in \{I, \text{SM}_1, \text{SM}_2, \text{SM}_3\}$ and then return to state i when this transition is possible according to the sleep profile (i.e., $T_j \neq 0$). $\Delta_{D_{i\to j}}$ are given by eqn. (1).
- $\max_{\forall \text{pkt}} \Big[D_w(s, a, \text{pkt}) + D_s(\text{pkt}) \Big]$ is the maximum experienced delay among the perceived delays by the served packets. For each of these packets (with index **pkt**), the experienced delay incorporates the two components previously detailed in section II: $D_w(s, a, \text{pkt})$ is the waiting delay in the buffer before service given the BS state s and the adopted action a and $D_s(\text{pkt})$ is the service delay. It is worth noting that we are implementing a 1-step Q-learning strategy and hence we are interested in computing the 1-step immediate cost function [15]. Hence, the considered delay component is calculated for packets arriving in the interval between two consecutive decision instants (defined in sec. (III-B)).

D. The Multi-Agent Q-learning Algorithm

Q-learning algorithm is based on the estimation of a tabular action value function that ranks actions according to their merit for each learning state. More precisely, the Q-value which is denoted by $Q(s, a)$ for each possible state s and action a pair evaluates the expected cumulative merit function resulting from adopting action a given the learner state s and consequently the optimal action to be taken at the network state s is the one that gives the best merit value.

The merit function in our network is the expected cumulative power consumption and delay costs resulting from adopting action a given the BS state s, and hence the optimal action is the one that has the minimum merit function.

The $Q(s, a)$ values for each learner l are obtained during the training phase by refining them at each decision instant according to the newly learned costs over a given number of iterations until convergence. Q-values are updated according to the following rule:

$$Q^l(s,a) \leftarrow Q^l(s,a) + \alpha \cdot [c^l(s,a) + \gamma \cdot (\min_{a'} Q^l(s',a')) - Q^l(s,a)] \tag{5}$$

where s denotes the current state of BS l, s' the next state after taking action a, $c^l(s, a)$ is the cost observed by agent l after performing action a given that it was in state s (Eqn. (4)), α is the learning rate, also called step size, it controls the rate at which new learned costs are accumulated, and γ is the discount factor, it takes values in $[0, 1]$ and describes the weight given to future Q values.

In order to accelerate convergence [15], we adopt in this work the decaying learning rate model considered in [18] and according to which the learning rate is updated as follows:

$$\alpha^l(t_u^l) \leftarrow \alpha^l(t_u^l - 1)\left(1 - \frac{1}{1 + M + t_u^l}\right)^{\frac{1}{2}+\xi} \tag{6}$$

with t_u^l the number of update steps (i.e. at step t_u, agent l is updating its Q-value for the t_u times), and M and ξ constants that control the decrease rate of the function.

Likewise, we consider a decaying epsilon-greedy strategy to control the choice of action a at each Q-value update step. Indeed, Q-learning algorithm is based on an iterative update for each of the $Q^l(s, a)$ values. A compromise is needed between exploiting the knowledge accumulated so far and refining further the Q-value of the action with the best estimated merit value or exploring new policies hoping to reach better merit values for these policies in the future. In this context, the epsilon-greedy strategy allows, at each Q-value update step t_u, the selection of the best learned policy so far with probability $1 - \epsilon$ and the exploration of a new policy with probability ϵ. By decaying the parameter ϵ with the update step t_u we allow the learning process to explore more at the beginning of the learning phase and we boost the rate of this process towards convergence by exploiting the learned merits when sufficient experience is accumulated. We specifically implement the stretched exponential decaying epsilon-greedy strategy defined for each BS l as follows [18]:

$$\epsilon^l(t_u^l) = \begin{cases} \max\left(\epsilon_{\min}, \epsilon_0 - \left[\frac{0.9 \cdot \epsilon_0}{\cosh\left(e^{-\frac{t_u^l - A \cdot Z}{B \cdot Z}}\right)} + \frac{t_u^l \cdot C}{Z}\right]\right) & \text{if } t_u^l \leq Z \\ \max\left(\epsilon_{\min}, \frac{\epsilon^* l(t_u^l = Z)}{t_u^l - Z}\right) & \text{otherwise} \end{cases} \tag{7}$$

with t_u^l, again, the number of update steps, $\epsilon_{\min}$ the targeted minimum exploration probability, ϵ_0 the initial exploration probability, Z a time horizon and A, B and C are parameters to control the shape/rate of decaying of the function.

IV. Numerical Applications

We consider, without loss of generality, two learning BSs. Users are uniformly distributed in the cell area of each BS. Table II summarizes the main parameters considered in our simulations. As previously indicated, each sleep profile n in the action set $\mathbf{A}_{Start-Idle}$ is defined by the quadruple $(T_{\text{I}}, T_{\text{SM}_3}, T_{\text{SM}_2}, T_{\text{SM}_1})$. To limit this set, we let T_i for all $i \in \{\text{I}, \text{SM}_3, \text{SM}_2, \text{SM}_1\}$ take limited number of values depending on the request arrival rate. These values, as well as those related to the learning process, were tuned empirically through extensive simulations we run beforehand.

We present in Table III our results for the optimal actions to take at each of the previously introduced BS states, both optimal sleep profile n^* and the action upon the arrival of the first request during the sleep period: wake-up immediately or

continue sleeping. Results are derived for different values of packet arrival rate λ and delay priority parameter β which indicates how much priority we give to delay over power saving. Results hold for the two considered BSs.

TABLE II: Simulation Inputs

Cell radius	500 [m]
Carrier frequency [GHz] (f_c)	3.5 [GHz]
Bandwidth [MHz] (W)	20 [MHz]
Transmitters Power [dBm]	46 [dBm]
Transmitters and Mobile receiver heights [m] (h_{BS} and h_{MS})	$h_{\text{BS}} = 35$ [m] $h_{\text{MS}} = 1.5$ [m]
Pathloss [dB]	3GPP model for the Urban Macro scenario with No Line of Sight [19] depending on f_c and the user distance from the BS
Power consumption at each operational mode [W] [14]	$P_{\text{active}} = 250$, $P_I = 109$, $P_{\text{SM}_1} = 52.3$, $P_{\text{SM}_2} = 14.3$ and $P_{\text{SM}_1} = 9.51$
Shadowing distribution	Log-normal with standard deviation 8 [dB]
Fading	Rayleigh Model
Thermal noise	-174 [dBm/Hz]
Service type	Best Effort
Packet size	1500 bytes
Packet arrivals	Log-normal inter-arrival time with mean $\tau = \frac{1}{\lambda}$ (λ [packet/s] is the mean arrival rate) and a variance $\nu = \frac{\tau}{10}$
Scheduling scheme	First In First Out
CQI/spectral-efficiency mapping	Table 5.2.2.1-2 in [17]
Epsilon-greedy params.	$\epsilon_0 = 1$, $\epsilon_{min} = 0.1$, $A = 0.6$, $B = 0.1$, $C = 0.05$, $Z = 6 \times 10^6$
Learning rate α params.	$\alpha_0 = 0.9$, $M = 3 \times 10^7$, $\xi = 10.5$
Discount factor γ	0.9

TABLE III: Optimal policies for different arrival rates and power saving versus delay trade-off strategies

β	$\lambda = 700$ [packets/s] (Traffic load $\rho = 0.051$)	$\lambda = 1500$ [packets/s] (Traffic load $\rho = 0.11$)	$\lambda = 3000$ [packets/s] (Traffic load $\rho = 0.21$)
0.3	$n^* \equiv (0.5, 10, 0, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.2, 0, 3, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.1, 0, 0, 49.7)$ [ms] Decision upon arrival: *stick to sleep profile* for CQI_0 to CQI_5 *immediate wake-up* for CQI_6 to CQI_{15}
0.6	$n^* \equiv (0.5, 0, 5, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.2, 0, 2, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.1, 0, 0, 7.1)$ [ms] Decision upon arrival *stick to sleep profile* for CQI_0 to CQI_3 *immediate wake-up* for CQI_4 to CQI_{15}
0.9	$n^* \equiv (0.5, 0, 2, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.2, 0, 1, 0)$ [ms] Decision upon arrival: *stick to sleep profile* for all CQI_k	$n^* \equiv (0.1, 0, 0, 5)$ [ms] Decision upon arrival: *immediate wake-up* for all CQI_k
1	$n^* \equiv (0.5, 0, 0, 0)$ [ms] $\equiv$ No sleep	$(0.2, 0, 0, 0)$ [ms] $\equiv$ No sleep	$(0.1, 0, 0, 0)$ [ms] $\equiv$ No sleep

The results show that lighter and shorter sleep periods are allowed when the experienced delay cost component is prioritized over the energy saving at the BS (i.e. increasing β). And this outcome holds for all levels of arrival rates. For instance, for $\lambda = 700$ [packets/s] (i.e traffic load $\rho = 0.051$), a sleep profile of 10 [ms] in the deepest sleep mode SM_3 can be achieved for $\beta = 0.3$ [packets/s], while a sleep duration of 2 [ms] in the medium sleep mode SM_2 is the best possible action a BS can take for $\beta = 0.9$. And as expected, no sleep is the ideal action a BS can take when delay is totally prioritized (i.e. for $\beta = 1$)).

On the other hand, for the same delay priority factor β, the sleep depth becomes lighter with increasing arriving rates. For instance, for $\beta = 0.3$, a sleep profile of 10 [ms] in the deepest sleep mode SM_3 can be achieved for $\lambda = 700$ [packets/s], while the optimal sleep profile for $\lambda = 1500$ [packets/s] is achieved for a sleep duration of 3 [ms] in sleep mode SM_2 and it is a sleep period of 49.7 [ms] in the lightest sleep mode SM_1 for $\lambda = 3000$ [packets/s]. These results are due to the fact that the larger the arrival rate the larger the accumulated delay cost. Hence, the BS will seek to minimize the activation waiting delays to counter-balance the previously mentioned delay costs.

We notice furthermore that the BS chooses to defer the service of demands arriving during the sleep mode as long as the energy saving gained from this deferment period can compensate the cost induced by the overall experienced delays. Hence, for low and medium levels of traffic loads (i.e. $\rho = 0.051$ and $\rho = 0.11$), the best action the sleeping BS would take upon the first request arrival is to stick to the sleep profile whatever the channel quality (i.e. the CQI) of this latter. And this outcome stems from the fact that the transmission delays in such network states are small and hence longer waiting delay can be acceptable for the service and consequently longer sleep duration and increased energy saving levels can be achieved.

While for higher traffic loads (i.e. $\rho = 0.21$), the BS extends its sleep up to a certain duration, when the CQI order of the first arriving request is less than a certain limit, and it wakes-up immediately otherwise. The limit CQI order up to which the BS can defer its service decreases with the increasing delay priority factor β. Namely, the BS defers its activation when the first arriving request has a CQI order less than or equal to CQI_5 when $\beta = 0.3$ and it is immediately activated whatever the CQI order for $\beta = 0.9$. This is due to the fact that, for the considered network loads, the BS adheres to the lightest sleep mode and hence the energy saving gained is marginal when compared to the active state and thus it is better to serve immediately the requests unless their channel quality is relatively bad.

We explore in the following the achieved energy gains versus delays. Specifically, we inject the obtained deterministic optimal policy into a Monte-Carlo, event-based simulator and obtain the energy saving versus delay performance, shown in Figures 1 and 2, respectively, as a function of different values of β and traffic loads.

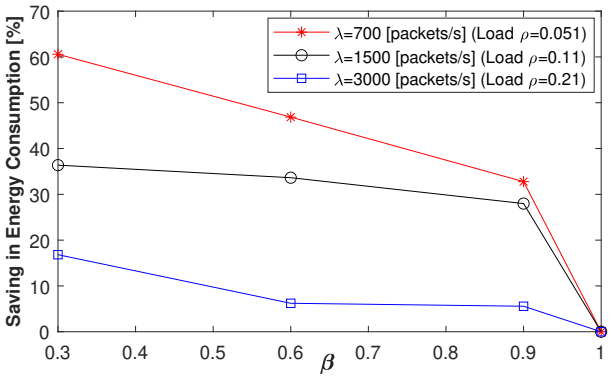

Fig. 1: Energy saving function of trade-off parameter

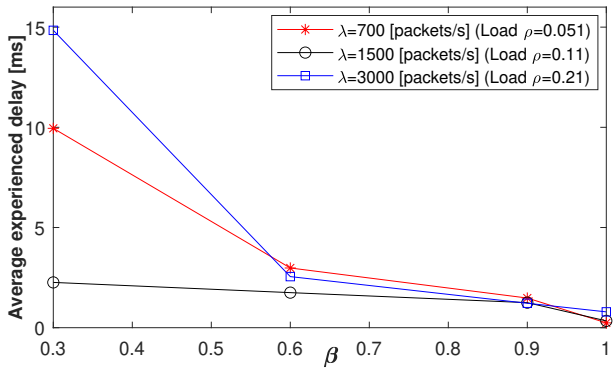

Fig. 2: Average delay function of trade-off parameter

We observe that with the implemented sleep strategy, energy saving up to 60% can be achieved in under-loaded networks (i.e. $\lambda = 700$ [packets/s]) at the expense of longer delay values. But even when the delay performance is prioritized over the energy saving with values of β approaching to 1, energy saving is relatively high with 40% for $\beta = 0.9$ for example. In relatively more loaded network, with $\lambda = 1500$ [packets/s], energy saving is still significant, with a maximum of 36%. For smaller β, the average delay is larger for smaller λ because in this case the system enters deeper sleep and takes more time to wake up and serve new arrivals. As β becomes larger, both delays become smaller. This difference in delay performance across different values of λ does not always follow this trend: for $\lambda = 3000$ for instance, even though the system does not have the opportunity to go deep into sleep, the queuing delay in the buffer becomes more significant and results in a larger delay performance for larger λ.

V. Conclusion and Future Work

We focused in this paper on the issue of putting BSs into sleep when they have no traffic to serve so as to save on the fixed component of the energy consumption, using 5G ASM feature. We considered the realistic case of several BSs which interfere with each other and proposed a multi-agent reinforcement learning approach, with a non-cooperative flavor, wherein each BS derives its own optimal sleep policy in order to achieve a target energy saving versus delay performance. The novelty of our work is that upon the detection of a user activity (i.e. arrival of the first transmission request) while the BS is in sleep, it can choose between waking up immediately and serving the user or to continue to sleep, depending on the desired energy saving versus delay performance trade-off, which includes the impact of the radio conditions of the user which may be poor and would result in this case in a long service duration; we would hence save more energy and also delay by serving it later. Our results show that the optimal sleep policy depends on the traffic arrival intensity, the timer index which decides when to put the BS into sleep in an off period, and the importance given to the power saving versus delay performance. Our next work is on the cooperative case wherein BSs can exchange information and target an overall objective of power saving versus delay performance to be obtained at the global, as opposed to individual, level.

References

[1] J. Lorincz, A. Capone, J. W. Greener, energy-efficient and sustainable networks: State-of-the-art and new trends. Sensors, 19(22):4864, 2019.

[2] I. P. Chochliouros, M-A. Kourtis, A. S. Spiliopoulou, P. Lazaridis, Z. Zaharis, C. Zarakovitis, A. Kourtis. Energy efficiency concerns and trends in future 5g network infrastructures. Energies, 14(17):5392, 2021.

[3] F. E. Salem, T. Chahed, E. Altman, A. Gati, Z. Altman. Optimal policies of advanced sleep modes for energy-efficient 5g networks. IEEE NCA, 2019.

[4] B. Debaillie, C. Desset, and F. Louagie. A flexible and future-proof power model for cellular base stations. IEEE VTC, 2015.

[5] E. Dahlman et al. 5G NR: The Next Generation Wireless Access Technology, chapter 5 - NR Overview. Academic Press, 2018.

[6] L. Busoniu, R. Babuska, B. De Schutter. Multi-agent reinforcement learning: An overview. Innovations in multi-agent systems and applications-1, pages 183–221, 2010.

[7] M. Meo, D. Renga, Z. Umar. Advanced Sleep Modes to comply with delay constraints in energy efficient 5G networks. IEEE VTC, 2021.

[8] O. Onireti, A. Mohamed, H. Pervaiz, M. Imran. Analytical approach to base station sleep mode power consumption and sleep depth. IEEE PIMRC, 2017.

[9] R. Tano, M. Tran, P. Frenger. KPI Impact on 5G NR Deep Sleep State Adaption. IEEE VTC, 2019.

[10] P. Lahdekorpi, M. Hronec, P. Jolma, J. Moilanen. Energy efficiency of 5G mobile networks with base station sleep modes. IEEE CSCN, 2017.

[11] H. Pervaiz, O. Onireti, A. Mohamed, M. A. Imran, R. Tafazolli, Q. Ni. Energy-Efficient and Load-Proportional eNodeB for 5G User-Centric Networks: A Multilevel Sleep Strategy Mechanism. IEEE Vehicular Technology Magazine, Volume: 13, Issue: 4, Dec. 2018.

[12] F. E. Salem, T. Chahed, E. Altman, A. Gati, Z. Altman. Scalable Markov Decision Process Model for Advanced Sleep Modes Management in 5G Networks. VALUETOOLS, 2020.

[13] M. Masoudi, M. G. Khafagy, E. Soroush, D. Giacomelli, S. Morosi, C. Cavdar. Reinforcement Learning for Traffic-Adaptive Sleep Mode Management in 5G Networks. IEEE PIMRC, 2020.

[14] A. El-Amine, M. Iturralde, H. Al Haj Hassan, L. Nuaymi. A Distributed Q-learning Approach for Adaptive Sleep Modes in 5G Networks. IEEE WCNC, 2019.

[15] R. S. Sutton, A. G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[16] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu. Adaptive block level resource allocation in OFDMA networks. IEEE Trans. on Wireless Communications, vol. 10, no. 11, pp. 3966–3972, 2011.

[17] ETSI. TS 138 214 V15.2.0, 5G; NR; Physical layer procedures for data (3GPP TS 38.214 version 15.2.0 Release 15). 2018.

[18] A. Ben-Ameur, A. Araldo, T. Chahed. Cache Allocation in Multi-Tenant Edge Computing via online Reinforcement Learning. IEEE ICC, 2022.

[19] 3GPP. TR 38.901 V14.3.0 Study on channel model for frequencies from 0.5 to 100 GHz (Release 14). 2017.