

# Evaluation of a Multi-cell and Multi-tenant Capacity Sharing Solution under Heterogeneous Traffic Distributions

I. Vilà, J. Pérez-Romero, O. Sallent, A. Umbert

Signal Theory and Communications Department of Universitat Politècnica de Catalunya (UPC), Barcelona, Spain  
irene.vila.munoz@upc.edu, [jorperez, sallent]@tsc.upc.edu, anna.umbert@upc.edu

**Abstract**— One of the key features of the 5G architecture is network slicing, which allows the simultaneous support of diverse service types with heterogeneous requirements over a common network infrastructure. In order to support this feature in the Radio Access Network (RAN), it is required to have capacity sharing mechanisms that distribute the available capacity in each cell among the existing RAN slices while satisfying their requirements and efficiently using the available resources. Deep Reinforcement Learning (DRL) techniques are good candidates to deal with the complexity of capacity sharing in multi-cell scenarios where the traffic in the different cells can be heterogeneously distributed in the time and space domains. In this paper, a multi-agent reinforcement learning-based solution for capacity sharing in multi-cell scenarios is discussed and assessed under heterogeneous traffic conditions. Results show the capability of the solution to satisfy the requirements of the RAN slices while using the resources in the different cells efficiently.

**Keywords**— RAN Slicing; Capacity sharing; Multi-Agent Reinforcement Learning; Deep Q-Network; Multi-cell.

## I. INTRODUCTION

Network slicing is one of the main features of the 5G system architecture that allows simultaneously supporting diverse service types and applications that can be provided by diverse tenants (e.g., mobile network operators or mobile virtual network operators) with heterogeneous needs over a common network infrastructure [1]. This is achieved by provisioning each tenant with a logical network, i.e., network slice, which is optimized to the specific needs of the supported services. To support network slicing at the Radio Access Network (RAN), i.e., the so-called RAN slicing, the common pool of radio resources in each of the existing cells needs to be efficiently managed so that the requirements of the different RAN slices are fulfilled and an efficient use of the available resources is achieved [2]. Given the spatial heterogeneity of the time varying traffic demands of the different RAN slices among the different cells, capacity sharing methods are needed to dynamically distribute the capacity available in the different cells among the existing RAN slices while satisfying their requirements and efficiently using the available resources.

This paper deals with the capacity sharing problem of RAN slicing in multi-cell scenarios. Although some works have proposed heuristic approaches to address this problem, such as [3] and [4], the complexity of 5G networks and the inherent uncertainty of the wireless environment have motivated the use of Deep Reinforcement Learning (DRL) solutions, as they allow optimizing dynamic decision-making problems in real time and

supporting large state and action spaces. In this regard, different DRL methods have been used in the literature to distribute the aggregated system capacity among tenants, such as, Deep Q-Network (DQN) [5], Deterministic Policy Gradients (DPG) solution combined with K-Nearest Neighbors (K-NN) [6] and Generative Adversarial Network (GAN)-Double DQN (DDQN) [7]. In contrast, other DRL-based works are able to assign capacities to the different tenants on a per-cell basis. This is the case of [8] and [9], which firstly distribute the aggregated capacity at network level among the different tenants by means of DQN and then use an heuristic algorithm to determine the cell capacity for each tenant, or our recent work in [10], which proposes a Multi-Agent Reinforcement Learning (MARL) algorithm based on DQN, where each agent is associated to a different tenant and directly determines the capacities provided to this tenant in each cell. However, none of these previous works have explored the impact of the spatial and temporal heterogeneities of the traffic distributions in multi-cell environments, which is considered to be essential to prove the robustness and practicality of DRL-based solutions in cellular networks. This paper addresses this aspect by studying the behavior of a MARL solution in a multi-cell scenario under diverse levels of spatial and temporal heterogeneity in the traffic distributions of the tenants and by analyzing the impact of the Service Level Agreement (SLA) parameters in these situations.

The rest of the paper is organized as follows. Section II describes the considered MARL solution in detail. Then, Section III describes the scenario considered for evaluation and provides the results that analyze the solution under heterogeneous spatial and temporal traffic distributions, and the impact of the SLA parameters. Finally, Section IV summarizes the conclusions.

## II. MARL-BASED CAPACITY SHARING MODEL

The capacity sharing solution considered in this paper has been designed to dynamically distribute the available capacity in a Next Generation (NG)-RAN infrastructure among  $K$  tenants, each of them provided with a RAN slice instance (RSI). The NG-RAN is composed of  $N$  cells, where each cell  $n=1,\dots,N$  has a total of  $W_n$  Physical Resource Blocks (PRBs) providing a cell capacity  $c_n$  (b/s). The total capacity in the system  $C$  is obtained by aggregating  $c_n$  for  $n=1..N$ . The solution aims at fulfilling the SLA established for each tenant while, at the same time, satisfying the traffic demands of the tenants in the different cells and efficiently using the available capacity. The requirements of the RSI of the  $k$ -th tenant are determined by the SLA of this tenant, which is defined in terms of: (a) the Scenario Aggregated Guaranteed Bit Rate,  $SAGBR_k$ , which is the

aggregated capacity to be provided across all cells to tenant  $k$  if requested, and (b) the Maximum Cell Bit Rate,  $MCBR_{k,n}$ , which is the maximum bit rate that can be provided to tenant  $k$  in cell  $n$ , and is defined to avoid that a single tenant uses all the capacity in a cell under highly heterogeneous spatial load distributions with tenants demanding excessive capacity in certain cells.

The considered solution dynamically adjusts the capacity share for each tenant in time steps of duration  $\Delta t$ . The capacity share  $\sigma_k(\mathbf{t})$  of tenant  $k$  at time step  $t$  is defined as  $\sigma_k(\mathbf{t}) = [\sigma_{k,1}(t), \dots, \sigma_{k,n}(t), \dots, \sigma_{k,N}(t)]$ , where  $\sigma_{k,n}(t)$  is the proportion of cell capacity  $c_n$  assigned to the tenant in cell  $n$  and is in the range  $0 \leq \sigma_{k,n}(t) \leq MCBR_{k,n}/c_n$ . For every cell  $n$ , the joint capacity share solution of all the tenants needs to assure that the cell capacity is not exceeded, so that  $\sum_{k=1}^K \sigma_{k,n}(t) \leq 1$  holds.

To dynamically tune  $\sigma_k(\mathbf{t})$  for the different tenants, a MARL approach based on DQN [11] has been considered. In the solution, there is one DQN agent per tenant so that the DQN agent of tenant  $k$  learns the policy  $\pi_k$  that tunes  $\sigma_k(\mathbf{t})$  at each time step  $t$ . This policy is defined as:

$$\pi_k = \underset{a_k}{\operatorname{argmax}} Q_k(s_k, a_k, \theta_k) \quad (1)$$

where  $Q_k(s_k, a_k, \theta_k)$  is the expected cumulative reward when starting at state  $s_k$  and taking action  $a_k$  and is provided by a Deep Neural Network (DNN) with weights  $\theta_k$ .

To learn their policies (i.e., the appropriate weights  $\theta_k$  of the DNN), the different DQN agents interact synchronously with a common network environment. At time step  $t$ , the DQN agent associated to tenant  $k$  obtains the state  $s_k(t)$  from the environment and, accordingly selects an action  $a_k(t)$  that updates the capacity  $\sigma_k(\mathbf{t})$ . This action selection follows an  $\epsilon$ -Greedy strategy that chooses an action based on the currently learnt policy  $\pi_k$  with probability  $1-\epsilon$  and explores a random action with probability  $\epsilon$ . At the next time step  $t+1$ , a reward  $r_k(t+1)$  assessing the suitability of  $a_k(t)$  for the  $s_k(t)$  is obtained as well as the new state  $s_k(t+1)$ . Then, the agent stores the experience tuple  $\langle s_k(t), a_k(t), r_k(t+1), s_k(t+1) \rangle$  in an experience dataset that will be used to update the policy  $\pi_k$ . The definitions of state, action and reward for the  $k$ -th tenant at time step  $t$  are given as follows:

- **State ( $s_k(t)$ ):** It is denoted as  $s_k(t) = [s_{k,1}(t), \dots, s_{k,n}(t), \dots, s_{k,N}(t), SAGBR_k/C, \sum_{k=1, k' \neq k}^K SAGBR_{k'}/C]$ , where each component  $s_{k,n}(t)$  corresponds to the state of the tenant in cell  $n$  given by  $\langle \rho_{k,n}(t), \rho_n^A(t), \sigma_{k,n}(t-1), \sigma_n^A(t-1), MCBR_{k,n}/c_n \rangle$ . The parameter  $\rho_{k,n}(t)$  is the resource usage, computed as the fraction of PRBs used by the tenant in the cell during the last time step  $(t-\Delta t, t)$ ,  $\rho_n^A(t)$  are the available resources not used by any tenant in the cell and  $\sigma_n^A(t)$  is the available capacity share in the cell not assigned to any tenant.
- **Action ( $a_k(t)$ ):** It is given by  $a_k(t) = [a_{k,1}(t), \dots, a_{k,n}(t), \dots, a_{k,N}(t)]$ , where  $a_{k,n}(t)$  is the specific action for each cell  $n$  and can take three different values  $a_{k,n}(t) \in \{\Delta, 0, -\Delta\}$ , which correspond to increasing, maintaining and decreasing the capacity share as  $\sigma_{k,n}(t) = \sigma_{k,n}(t-1) + a_{k,n}(t)$ . This update is performed as long as the resulting capacity share  $\sigma_{k,n}(t)$  is in the range  $0 \leq \sigma_{k,n}(t) \leq MCBR_{k,n}/c_n$ . Otherwise, no update is performed. Moreover, it must be ensured that the capacity shares of all tenants satisfy the condition  $\sum_{k=1}^K \sigma_{k,n}(t) \leq 1$ .

Therefore, when this condition is not satisfied, the available capacity share  $\sigma_n^A(t)$  is computed before applying the actions of the tenants willing to increase (i.e. with  $a_{k,n}(t) = \Delta$ ). If  $\sigma_n^A(t) > 0$ , the capacity shares of these tenants are obtained by distributing  $\sigma_n^A(t)$  among them proportionally to their  $SAGBR_k$ . Otherwise, the actions of these tenants are not applied.

- **Reward ( $r_k(t)$ ):** The obtained reward is given by:

$$r_k(t) = \delta_k^{(1)}(t)^{\varphi_1} \cdot \delta_k^{(2)}(t)^{\varphi_2} \quad (2)$$

which considers two main factors,  $\delta_k^{(1)}(t)$  and  $\delta_k^{(2)}(t)$ , defined in the following, and their corresponding weights,  $\varphi_1$  and  $\varphi_2$ . The first factor,  $\delta_k^{(1)}(t)$ , promotes the satisfaction of the SLA of tenant  $k$  and is given by the ratio between the aggregated throughput of the tenant among all cells  $T_k(t)$  and the aggregated offered load of the tenant among all cells  $O_k(t)$ , as long as the aggregate offered load in the system among all tenants,  $O(t)$ , is lower than the total capacity in the system  $C$ . Instead, if  $O(t)$  is greater than  $C$ ,  $\delta_k^{(1)}(t)$  is computed as the ratio between  $T_k(t)$  and  $\min(SAGBR_k + \beta_k(t), O_k(t))$ , where  $\beta_k(t)$  is the amount of assigned capacity that is left unused by the other tenants. The second factor,  $\delta_k^{(2)}(t)$ , measures the capacity overprovisioning and is defined by the ratio between  $T_k(t)$  and the assigned capacity to the tenant among all cells (i.e. the summation of  $c_n \cdot \sigma_{k,n}(t-1)$  for all  $n=1 \dots N$ ).

The experience tuples collected by the DQN agent as a result of the interaction with the environment are stored in the experience dataset and used for updating the weights  $\theta_k$  of the DNN in every time step during the training process. Following the process described in [11], this is performed according to the mini-batch gradient descent of the average Mean Square Error (MSE) loss of  $Q_k(s_k, a_k, \theta_k)$  over a mini-batch  $U$  of experiences randomly selected from the experience dataset.

### III. PERFORMANCE EVALUATION

#### A. Considered scenario

The scenario considered for evaluation is comprised of a NG-RAN infrastructure in a 3 km x 3 km area with  $N=5$  cells, which serve the users of  $K=2$  tenants, denoted as *Tenant 1* and *Tenant 2*. The scenario has been configured according to the parameters in Table I, which includes the cells configuration, their position in the area under consideration and the SLA parameters of each tenant.

To generate heterogeneous spatial and temporal distributions of the offered load of the two tenants in the different cells, it is assumed that at time step  $t$  the offered load density (Mb/s/km<sup>2</sup>) of tenant  $k$  is spatially distributed according to the sum of a constant offered load density  $\mu_k$  and a bivariate Gaussian distribution centered at the position  $(x_k(t), y_k(t))$  with standard deviation  $d_k$  and offered load density in the center  $m_k$ . The center of the Gaussian distribution  $(x_k(t), y_k(t))$  moves horizontally along the scenario with speed  $v_k$ . Then, the offered load of tenant  $k$  in cell  $n$  at time step  $t$ ,  $o_{k,n}(t)$ , is obtained by aggregating the offered load density over the cell service area determined by the Voronoi tessellation.

TABLE I. SCENARIO CONFIGURATION

Parameter	Value	
Number of tenants ( $K$ )	2	
Number of cells ( $N$ )	5	
Area	3 km x 3 km	
Cell position (km)	Cell $n=1$	(1.5, 1.5)
	Cell $n=2$	(0.5, 2.5)
	Cell $n=3$	(2.5, 2.5)
	Cell $n=4$	(0.5, 0.5)
	Cell $n=5$	(2.5, 0.5)
PRB Bandwidth ( $B$ )	360 kHz	
Number of PRBs per cell ( $W_n$ )	78 PRBs	
Average spectral efficiency	5 b/s/Hz	
Cell capacity ( $c_n$ )	140 Mb/s	
Total system capacity ( $C$ )	700 Mb/s	
$SAGBR_k$	Tenant 1	420 Mb/s (60% of system capacity)
	Tenant 2	280 Mb/s (40% of system capacity)
$MCBR_{k,n}$	Tenant 1	112 Mb/s (80% of cell capacity $c_n$ )
	Tenant 2	

Based on this methodology, the MARL model has been evaluated under four different offered load situations that reflect different levels of heterogeneity, denoted as *Situations 1-4*, whose configuration parameters have been included in Table II. For each situation, the offered load of each tenant in each cell has been obtained during a day. The level of heterogeneity in the different situations is varied through the values of offered load density in the center  $m_k$  and deviation  $d_k$ . As a result, Situation 1 corresponds to a situation where the spatial distribution of the offered load of one tenant among the different cells is nearly homogeneous. Then, the level of heterogeneity is increased in Situations 2-4, being Situation 4 the one with the most unbalanced load among cells. To illustrate this, Fig. 1 plots the maps with the offered load densities for *Tenant 1* and *Tenant 2* in Situation 4 at some illustrative times. The black triangles indicate the positions of the 5 cells.

The MARL model has been developed in Python by using the library *TF-Agents* [12], which provides tools for the development of DRL models including DQN, and has been trained according to the parameters of Table III. The data used for the training has been generated by obtaining a wide range of spatial and temporal distributions of the offered loads of both tenants, obtained by modifying the values of the parameters of the Gaussian distributions of Tenant 1 and Tenant 2, including their initial position and speed, and diverse combinations of their SLA parameters.

### B. Performance under diverse heterogeneity levels

In order to evaluate the capability of the policies learnt by the DQN agents of the two tenants to adapt to different levels of heterogeneity, they have been applied to the offered loads  $o_{k,n}(t)$  in the Situations 1-4. Fig. 2 compares the resulting average offered load and the average assigned capacity (both expressed

TABLE II. CONFIGURATION OF OFFERED LOAD SITUATIONS

Parameter	Tenant 1	Tenant 2
Initial position( $x_k(0), y_k(0)$ ) (km)	(0, 0.5)	(1.5, 2.5)
Speed ( $v_k$ ) (km/h)	0.125	-0.29
Offered load density configuration ( $m_k$ (Mb/s/km <sup>2</sup> ), $d_k$ (km))	Situation 1	(24, 5)
	Situation 2	(28, 3)
	Situation 3	(36, 1)
	Situation 4	(72, 1)
Constant offered load density ( $\mu_k$ ) (Mb/s/km <sup>2</sup> )	20	16

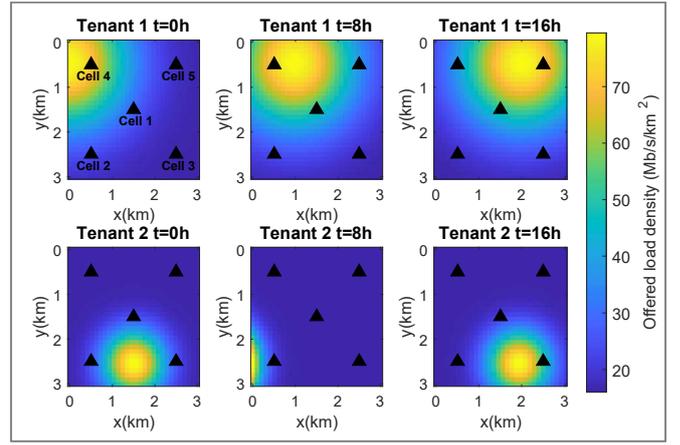


Fig. 1. Offered load density maps of Tenant 1 and 2 during a day.

TABLE III. MARL MODEL PARAMETERS

Parameter	Value
Initial collect steps	5000
Maximum number of time steps for training	$2 \cdot 10^6$
Experience Replay buffer maximum length ( $l$ )	$10^7$
Mini-batch size ( $J$ )	256
Learning rate ( $\tau$ )	0.0001
Discount factor( $\gamma$ )	0.9
$\epsilon$ value ( $\epsilon$ -Greedy)	0.1
DNN nodes	100 nodes x 1 layer
Reward weights ( $\phi_1, \phi_2$ )	(0.5, 0.4)
Time step duration ( $\Delta t$ )	5 min
Action step ( $\Delta$ )	0.03

as a percentage of the cell capacity  $c_n$ ) in the different cells for Tenant 1 and Tenant 2 in Situations 1-4. The aggregated offered load and the aggregated assigned capacity of each tenant at system level (i.e., among all cells) is also included as a percentage of the total capacity. Results reveal that the assigned capacity takes close values to the offered load requirements both at cell and system levels for the different situations, regardless of the level of heterogeneity. In fact, the obtained differences between the offered loads and the assigned capacities are lower than 8% for all cases, which are mainly due to the incremental action design, which makes that the assigned capacity fluctuates around the offered load within a margin between  $\Delta$  and  $-\Delta$ . The highest differences are observed for cell 4 in Situations 2 and 3 and for cells 2 and 5 in Situation 4, since their total offered load in these cells exceeds  $c_n$  during some periods, so the offered load of both tenants in those cells cannot be satisfied all the time (this can be seen when looking at the period between 19h and 22h of the second graph of Fig. 3 that depicts the cell offered load normalized to the cell capacity against the assigned capacity for cell 2). Moreover, results show that in certain cases when the traffic among cells is unbalanced and, in some cells, the offered load is higher than the relative  $SAGBR_k$ , the policy is able to support this load by smartly distributing the assigned capacity in accordance with the spatial traffic distribution. For example, the average offered load of Tenant 1 in Situation 4 in cells 4 and 5 exceeds the relative  $SAGBR_k$  of 60% but the policy is able to support it since the offered load in the rest of cells is much lower than 60%. These results highlight the capability of the proposed solution to satisfactorily adapt to diverse levels of offered load heterogeneity among cells.

Table IV includes the average reward, the average SLA satisfaction and the average assigned capacity ratio. The average SLA satisfaction is defined as the average ratio (bounded to 1) between the aggregate throughput  $T_k(t)$  and the minimum of the aggregate of  $\min(o_{k,n}(t), MCBR_{k,n})$  over all cells and  $SAGBR_k$ . The average assigned capacity ratio is defined as the ratio between the assigned cell capacity and the cell offered load over all the cells in the scenario. These metrics have been obtained for both tenants considering the average over one day. Results show that the learnt policies achieve high average reward for both tenants in all situations. The good performance is also reflected in the values of SLA satisfaction, which are higher than 0.96 for Tenant 1 and 0.93 for Tenant 2. The fact that slightly lower SLA satisfaction is obtained for Tenant 2 is that the traffic level of Tenant 2 is generally lower than that of Tenant 1, so the SLA satisfaction is more affected by the increases and decreases in action steps  $\Delta$  than in the case of Tenant 1. Finally, in relation

to the assigned capacity ratio, the obtained values are close to 1, with maximum deviations of 8%. This indicates that the assigned capacity properly matches the offered load with little overprovisioning.

### C. SLA parameters impact

In this section, the impact of the SLA parameters on achieved behaviour of the proposed approach in the highly unbalanced Situation 4 is discussed.

Fig. 3 compares the temporal evolution of the capacity shares  $\sigma_{k,n}(t)$  in cell 2 for Situation 4 from 12 hours to 24 hours against the cell offered load  $o_{k,n}(t)$  normalized to the cell capacity  $c_n$ . Results are shown for different values of  $MCBR_{k,n}$  considering that the same  $MCBR_{k,n}$  is configured for all cells and tenants. The rest of the parameters have been configured according to Table I. During the analyzed period, the offered load of Tenant 2 presents very high values between 19 hours until 21 hours, requiring nearly all the capacity in the cell. Instead, the offered load of Tenant 1 remains low, requiring around 35% of the cell capacity. Results show that the trained policies are able to assign the capacity without exceeding the  $MCBR_{k,n}$  limit in any case. It is observed that depending on the  $MCBR_{k,n}$  value, the assigned capacity shares of the tenants are differently distributed. While for  $MCBR_{k,n}=60\%$  and  $MCBR_{k,n}=80\%$ , the assigned capacity of Tenant 2 is limited and the offered load of Tenant 1 is satisfied, for  $MCBR_{k,n}=100\%$ , the offered load of Tenant 1 is not satisfied since nearly all the capacity is assigned to Tenant 2. Also, it is observed that for  $MCBR_{k,n}=60\%$  the capacity in the cell is not efficiently assigned since around 15% of the cell capacity remains unused from 18 hours to 23 hours, although this capacity could have been assigned to Tenant 2. These results show the relevance of including the  $MCBR_{k,n}$  parameter in the SLA to deal with unbalanced offered loads among cells, contributing to a more efficient and fairer distribution when is adequately configured.

Fig. 4 includes the aggregated offered load among all cells,  $O_k(t)$  (i.e., the summation of  $o_{k,n}(t)$  for  $n=1..N$ ) and the aggregated assigned capacity among all cells,  $A_k(t)$  (i.e., the summation of  $c_n \cdot \sigma_{k,n}(t)$  for  $n=1..N$ ), both normalized to the total system capacity  $C$ , for both tenants with different values of  $MCBR_{k,n}$  for the same period of Fig. 3. The  $SAGBR_k$  values expressed as percentages of the system capacity are also depicted. For a low value of  $MCBR_{k,n}$ , such as  $MCBR_{k,n}=60\%$ , the aggregated offered loads of both tenants at system level are not satisfied, leading to an inefficient use of the system capacity, since no more than the 60% of the cell capacity can be assigned to any tenant. In turn, when the value of  $MCBR_{k,n}$  is increased the aggregated capacity shares  $A_k(t)$  of both tenants take closer values to  $O_k(t)$ . However,  $A_k(t)$  still presents lower values than  $O_k(t)$ , since in some cells there is not enough capacity to satisfy both tenants. Quantitatively, the average system utilization (i.e. the average of the ratio between the summation of  $A_k(t)$  of both tenants and the system capacity) for  $MCBR_{k,n}=60\%$  is 77% while for  $MCBR_{k,n}=80\%$  and  $MCBR_{k,n}=100\%$  is 83% and 82%, respectively. Also, good average SLA satisfaction ratios are obtained for both tenants, ranging from 0.9 up to 0.97 for the different analysed  $MCBR_{k,n}$  values. The obtained results reinforce the relevance of choosing an appropriate value of  $MCBR_{k,n}$  since it also affects the performance at system level.

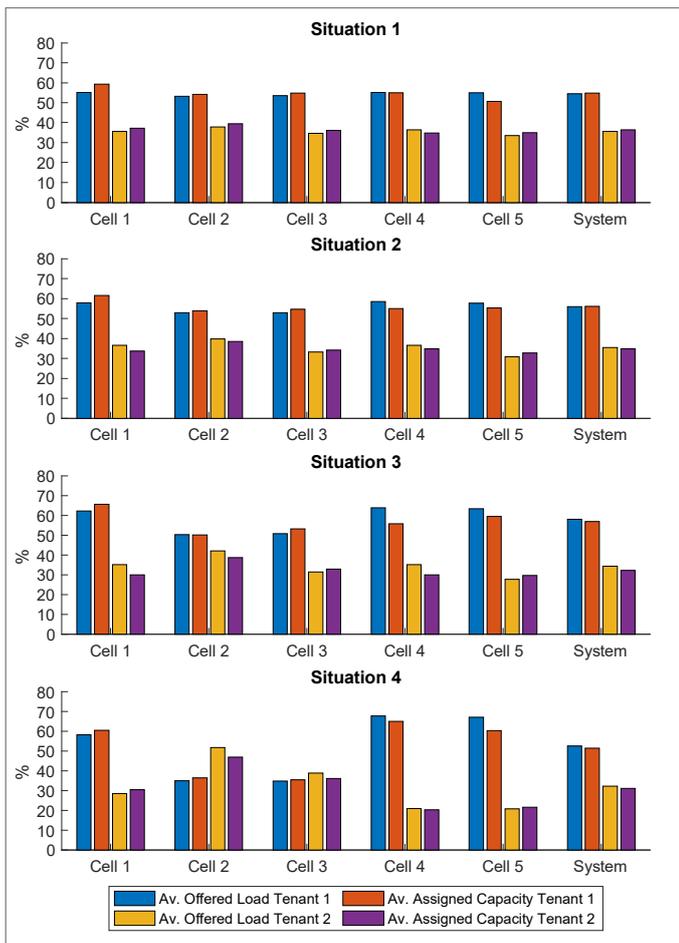


Fig. 2 Average offered load and assigned capacity per cell and at system level for each considered situation.

TABLE IV. SYSTEM-LEVEL PERFORMANCE INDICATORS

Performance Indicator		Situation 1	Situation 2	Situation 3	Situation 4
Av. Reward	Tenant 1	0.96	0.97	0.96	0.96
	Tenant 2	0.95	0.94	0.94	0.94
Av. SLA satisfaction	Tenant 1	0.97	0.98	0.97	0.96
	Tenant 2	0.94	0.97	0.95	0.93
Av. assigned capacity ratio	Tenant 1	1.04	1.01	1.01	1.08
	Tenant 2	1.04	1.02	1.08	1.05

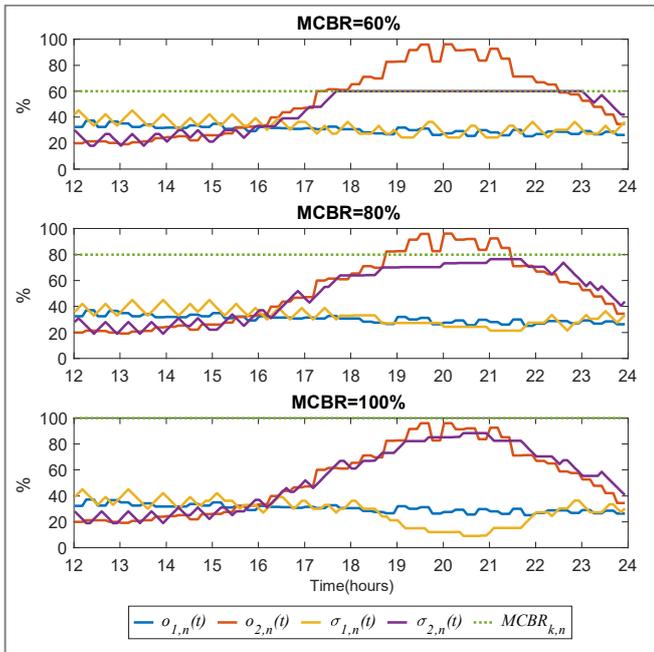


Fig. 3. Capacity shares vs offered load of tenants for cell 2 in Situation 4 and for diverse values of MCBR.

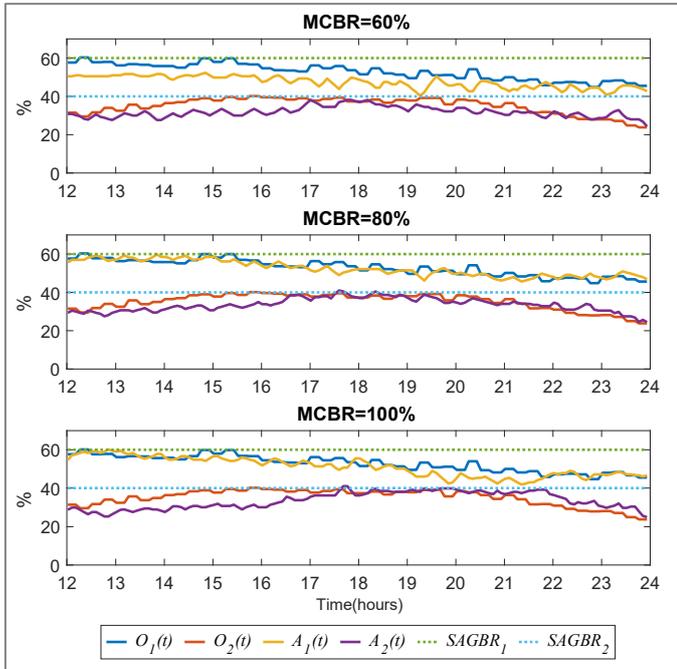


Fig. 4. Aggregated offered loads and assigned capacities in the system

#### IV. CONCLUSIONS

This paper has studied a capacity sharing solution for RAN slicing in multi-cell scenarios, designed as a multi-agent reinforcement learning model based on Deep Q-Network (DQN). Each DQN agent in the solution is associated to a different tenant and learns the capacity to be assigned to the tenant in the different the cells of the scenario so that the traffic demands are fulfilled while satisfying the Service Level Agreement (SLA) and making an efficient use of the resources in the different cells.

The model has been trained considering a multi-cell scenario with two tenants that present heterogeneously distributed traffic demands among the different cells, which variate during time. The trained model has been evaluated considering diverse levels of temporal and spatial traffic heterogeneity among cells and the impact of the SLA parameters has been assessed. Results have shown that the considered solution adapts the assigned capacity to each tenant in each cell to the traffic requirements, proving the robustness of the solution. This is achieved while satisfying the SLA of the tenants with average satisfaction ratios above 0.93, and efficiently using the available resources. Also, the relevance of including the maximum cell bit rate parameter in the SLA has been shown since it contributes to a more efficient and fair capacity assignment under highly unbalanced traffic situations.

#### ACKNOWLEDGMENT

This work has been supported by the Spanish Research Council and FEDER funds under SONAR 5G grant (ref. TEC2017-82651-R), by the European Commission's Horizon 2020 5G-CLARITY project under grant agreement 871428 and by the Secretariat for Universities and Research of the Ministry of Business and Knowledge of the Government of Catalonia under grant 2020FI\_B2 00075.

#### REFERENCES

- [1] P. Rost, et al. "Mobile network architecture evolution toward 5G," in *IEEE Communication Magazine*, May, 2016.
- [2] R. Ferrús, et al., "On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration", *IEEE Communications Magazine*, May, 2018, pp.184-192.
- [3] J. Shi, et al., "Hierarchical Auction and Dynamic Programming Based Resource Allocation (HA&DP-RA) Algorithm for 5G RAN Slicing," 2018 *24th Asia-Pacific Conference on Communications (APCC)*, Ningbo, China, 2018, pp. 207-212.
- [4] J. Gang, et al., "Optimal resource sharing in multi-tenant 5G networks," 2018 *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, 2018, pp. 1-6.
- [5] R. Li et al., "Deep Reinforcement Learning for Resource Management in Network Slicing," in *IEEE Access*, vol. 6, pp. 74429-74441, 2018.
- [6] C. Qi, et al., "Deep Reinforcement Learning With Discrete Normalized Advantage Functions for Resource Management in Network Slicing," in *IEEE Communications Letters*, vol. 23, no. 8, pp. 1337-1341, Aug. 2019.
- [7] Y. Hua, et al., "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," in *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 334-349, Feb. 2020.
- [8] G. Sun, et al., "Dynamic Reservation and Deep Reinforcement Learning Based Autonomous Resource Slicing for Virtualized Radio Access Networks," in *IEEE Access*, vol. 7, pp. 45758-45772, 2019.
- [9] G. Sun, et al., "Autonomous Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network," in *IEEE Systems Journal*, vol. 13, no. 3, pp. 2454-2465, Sept. 2019.
- [10] I. Vilà, et al., "A Novel Approach for Dynamic Capacity Sharing in Multi-tenant Scenarios," 2020 *IEEE 31st International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, London, United Kingdom, 2020, pp. 1-6.
- [11] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.
- [12] S. Guadarrama, et. al (2018). TF-Agents: A library for Reinforcement learning in TensorFlow. Available at: <https://github.com/tensorflow/agent>.