A Deep Q Network-based Multi-Connectivity Algorithm for Heterogeneous 4G/5G Cellular Systems

J. J. Hernández-Carlón [1*], J. Pérez-Romero[1], O. Sallent[1], I. Vilà[1] and F. Casadevall[1]

[1] Department of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC) Barcelona, 08034, Spain,

*correspondence
{juan.jesus.hernandez*,jordi.perez-romero,
irene.vila.munoz}@upc.edu, {sallent,ferranc}@tsc.upc.edu

Abstract. Multi-connectivity, which allows a user equipment to be simultaneously connected to multiple cells from different radio access network nodes that can be from a single or multiple radio access technologies, has emerged as a useful feature to handle the traffic in heterogeneous cellular scenarios and fulfill high data rate and reliability requirements. This paper proposes the use of deep reinforcement learning to optimally split the traffic among cells when multi-connectivity is considered in a heterogeneous 4G/5G networks scenario. Obtained results reveal a promising capability of the proposed Deep Q Network solution to select quasi optimum traffic splits depending on the current traffic and radio conditions in the considered scenario. Moreover, the paper analyses the robustness of the obtained policy in front of variations with respect to the conditions used during the training.

Keywords: Multi-connectivity, deep reinforcement learning, Deep Q Network, heterogeneous networks

1 Introduction

With the advent of 5G Mobile Network Operators (MNOs) face further increase in network deployment heterogeneity with different cell types (e.g. macrocells, indoor and outdoor small cells) based on multiple Radio Access Technologies (RATs) (e.g., 2G, 3G, 4G and 5G New Radio (5G NR)), operating in different spectrum bands (e.g. sub 6 GHz bands used by all RATs and millimeter wave (mmW) bands used by 5G New Radio). In this context, Multi-Connectivity (MC) technology enables a User Equipment (UE) to be simultaneously connected to multiple nodes of the Radio Access Network (RAN), e.g. eNodeBs (eNB) operating with LTE and/or gNodeBs (gNB) operating with 5G NR [1,2]. There is one master node (MN) responsible for the radio-access control plane and one, or in the general case multiple, secondary node(s) (SN) that provide additional user-plane links. In this way, a UE can aggregate the radio resources from multiple eNBs/gNBs, which allows efficiently achieving the 5G requirements of high data rate and ultra-reliability. The literature has considered different problems in relation to MC, such as the resource allocation in [3,4] or the traffic split [5-8].

This paper addresses the traffic split multi-connectivity problem in multi RAT scenarios by exploiting Deep Q Network (DQN) technique [9] to obtain a policy that allows optimally distributing the traffic of a UE across the different RATs and cells while fulfilling the QoS requirements and optimizing the bandwidth consumption of the UE, so that overload situations are avoided in the involved cells. Deep reinforcement learning techniques such as DQN are useful for optimizing dynamic decision-making problems that depend on a large number of input variables taking a wide range of possible values. This is the case of the MC problem formulated in this paper, for which a DQN solution is presented and assessed by means of simulations. In addition, the results presented in this paper pay particular attention to the capability of the solution to generalize the knowledge learnt during the training phase. In this direction, the robustness of the learnt policy is analyzed when the conditions experienced by the algorithm differ from the ones that were considered during the training.

The rest of the paper is organised as follows. Section 2 presents the system model and formulates the considered multi-connectivity problem. The proposed DQN-based solution is presented in Section 3 and different performance results are provided in Section 4. Finally, Section 5 summarises the conclusions.

2 System model and problem definition

Let us consider a heterogeneous RAN where different UEs with multi-connectivity capabilities are camping. A given UE *u* considers *M* different RATs and *N* different cells per RAT as candidates for the multi-connectivity. Then, let us denote as $A_u=\{C_{m,n}\}$ the set of candidate cells detected by the UE *u*. $C_{m,n}$ denotes the *n*-th cell of the *m*-th RAT with *n*=1, ..., *N* and *m*=1,...,*M*. It is worth mentioning that, due to the mobility of the UE, the specific cells that the UE detects in a given RAT may change with time. In this respect, it is assumed that the *N* cells of a RAT correspond to the best *N* cells detected by the UE at a certain time based on measurements averaged during a time window ΔT.

Through the use of multi-connectivity, the traffic of the *u*-th UE is split across multiple RATs/cells of the set A_u. It is assumed that, at a certain time, the UE can be simultaneously connected to a maximum of N_{max} cells among the $M \cdot N$ candidates. The multi-connectivity configuration for the *u*-th UE can be expressed as the $M \times N$ matrix $\mathbf{B}=\{\beta_{m,n}\}$ where $\beta_{m,n}\in[0,1]$ defines the fraction of total traffic of UE *u* that is delivered through the *n*-th cell of the *m*-th RAT. Then, the objective is to find the optimal configuration $\mathbf{B=\{}\beta_{n,m}\}$ to be applied in a time window of ΔT s that allow ensuring the Quality of Service (QoS) requirements with minimum resource consumption and avoiding overload situations in the different RATs/cells. In this respect, it is assumed that the QoS requirements of the user *u* are expressed in terms of a required bit rate R_u (b/s) to be provided.

To formalize the problem, let us denote as $T_u(\mathbf{B})$ the total throughput or bit rate obtained by user u during the last time window period ΔT with the multi-connectivity configuration $\mathbf{B}$. Let us also denote $a_{m,n}(\beta_{m,n})$ as the number of physical resource blocks (PRBs) in the m-th cell and n-th RAT assigned to the u-th UE to transmit the traffic corresponding to $\beta_{m,n}$. Considering that $b_{m,n}$ corresponds to the bandwidth of one PRB in the m-th cell and n-th RAT, the bandwidth allocated to the user u in this RAT, denoted as $\gamma(\beta_{m,n})$, is given by:

$$\gamma(\beta_{m,n}) = a_{m,n}\big(\beta_{m,n}\big). b_{m.n} \tag{1}$$

In addition, the total fraction of occupied PRBs in a RAT/cell accounting for all the UEs connected to that cell is denoted as $\rho_{m,n}(\beta_{m,n})$. Then, the considered problem to be solved for the u-th UE is formally defined as:

$$\mathbf{B} = arg\,\underset{\boldsymbol{B}}{min}\left[\frac{1}{w_{max}}\sum_{m=1}^{M}\sum_{n=1}^{N}\gamma\big(\beta_{m,n}\big)\right] \tag{2}$$

$$\text{s.t.} \quad T_u(\boldsymbol{B}) \geq R_u \ , \quad \rho_{m,n}\big(\beta_{m,n}\big) \leq \rho_{max} \quad \forall m, n$$

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\beta_{m,n} = 1$$

where w_{max} is the maximum possible bandwidth to be assigned to the user u and $\rho_{max}\in[0,1]$ is the maximum threshold established to avoid overload situations in a cell.

Fig. 1 depicts the architectural components to enforce the multi-connectivity configuration $\mathbf{B}$ in the network, obtained as a result of the above problem. The figure illustrates an example for the downlink traffic transmitted to a UE served by two cells of RAT m=1 (e.g., 5G). The cell n=1 is handled by the MN and the cell n=2 by the SN. The traffic between these cells is split at the Packet Data Convergence Protocol (PDCP) layer of the MN using dual connectivity feature. The multi-connectivity configuration is determined by an MC controller that takes as an input different measurement from the RATs/cells as it will be explained in Section 3. The output of the MC controller is the configuration $\mathbf{B}=\{\beta_{m,n}\}$ with the weights $\beta_{m,n}$ to be configured at the PDCP layer of the MN to split the traffic between cells 1 and 2. Finally, the Medium Access Control (MAC) scheduler in each 5G NR or LTE cell will allocate the necessary amount of PRBs $a_{m,n}(\beta_{m,n})$ to the UE to transmit the fraction of traffic $\beta_{m,n}$ corresponding to the cell. The specific design of the MAC scheduler is out of the scope of this work, but in general it will consider aspects such as the propagation and interference conditions observed by the UE, the QoS requirements, the amount of UEs in the cell, etc.

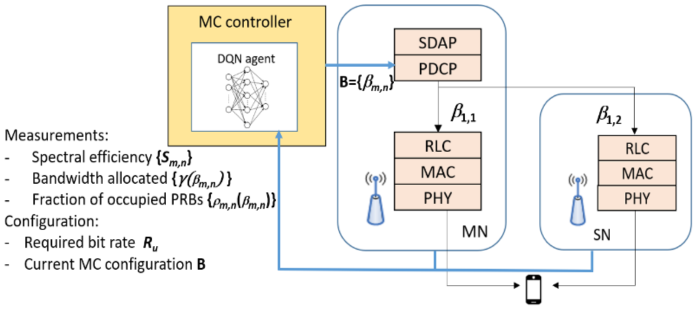
MC controller
DQN agent
$\mathbf{B}=\{\beta_{m,n}\}$
SDAP
PDCP
$\beta_{1,1}$
RLC
MAC
PHY
MN
$\beta_{1,2}$
RLC
MAC
PHY
SN
Measurements:
- Spectral efficiency $\{S_{m,n}\}$
- Bandwidth allocated $\{\gamma(\beta_{m,n})\}$
- Fraction of occupied PRBs $\{\rho_{m,n}(\beta_{m,n})\}$
Configuration:
- Required bit rate R_u
- Current MC configuration B

Fig. 1. Architectural components of the considered approach

3 DQN-based solution

A DQN approach is considered in this paper for solving the MC problem formulated in previous section. In this approach, the learning process is conducted dynamically by a DQN agent at the MC controller of Fig. 1 that makes decisions for the different UEs. The agent operates in discrete times with granularity equal to the time window duration ΔT. These discrete times are denoted as t, t+1, ..., t+k,... At time t the DQN selects an action $\boldsymbol{a(t)}$ that contains the MC configuration to be applied for a given UE in the next time window. The action selection is based on the current state at time t, denoted as $\boldsymbol{s(t)}$ and on the decision-making policy available at this time. Then, as a result of applying the selected MC configuration, a reward signal $r(t+1)$ is provided to the DQN agent at the end of the time window. This reward signal measures how good or bad was the last performed action and therefore it is used to improve the decision-making policy. The different components of this process are detailed in the following.

3.1 State, action and reward specification

The state $\boldsymbol{s}(t)$ is a vector that includes the following components for a given UE u:

- Requirements of UE u: R_u.
- Spectral efficiency per RAT/cell $\{S_{m,n}\}$ of UE u.
- Current configuration $\mathbf{B}=\{\beta_{m,n}\}$, which corresponds to the configuration applied at time t-1.
- Bandwidth occupied by the UE u in each RAT/cell $\{\gamma(\beta_{m,n})\}$.
- Fraction of total occupied resources in each RAT/cell $\{\rho_{m,n}(\beta_{m,n})\}$.

All the values $S_{m,n}$, $\gamma(\beta_{m,n})$ and $\rho_{m,n}(\beta_{m,n})$ are average values measured during the last time window of duration ΔT, i.e. between discrete times t-1 and t.

Each action $\boldsymbol{a(t)}\in\mathcal{A}$ represents a matrix $\mathbf{B}=\{\beta_{m,n}\}$ that corresponds to the MC configuration to be applied during the next time window ΔT. The action space $\mathcal{A}$ includes all the MC configurations and is defined by considering that the possible $\beta_{m,n}$ values are discretized with granularity $\Delta\beta$ and the aggregate of all $\beta_{m,n}$ values in matrix $\mathbf{B}$ equals 1. Moreover, the action space considers that the UE can be simultaneously connected to at most N_{max} cells of the $N{\cdot}M$ candidates, i.e. that at most N_{max} values of $\beta_{m,n}$ can be different from 0.

The reward $r(t+1)$ intends to measure how good or bad was the performance obtained by the last action $\boldsymbol{a(t)}$ for the state $\boldsymbol{s}(t)$ in relation to the target of the optimization. Then, considering the optimization problem (2), and that the last action $\boldsymbol{a(t)}$ is given by MC configuration $\mathbf{B}=\{\beta_{m,n}\}$, the reward is defined as:

$$r(t+1) = \left(1 - \frac{1}{w_{max}} \sum_{m=1}^{M} \sum_{n=1}^{N} \gamma(\beta_{m,n})\right) \cdot min\left(1, \frac{T_u(\mathbf{B})}{R_u}\right) \tag{3}$$

$$\cdot \prod_{\substack{m,n \\ \beta_{m,n}>0}} min\left(1, \frac{\rho_{max}}{\rho_{m,n}(\beta_{m,n})}\right)$$

The first term in $r(t+1)$ captures the total bandwidth assigned to the UE u in all the cells/RATs, so the lower the amount of bandwidth assigned the higher will be the reward. The second term represents a penalty introduced when the achieved throughput $T_u(\mathbf{B})$ is lower than the minimum requirement R_u. The last term introduces a penalty for each cell/RAT in which the UE has transmitted traffic (i.e. $\beta_{m,n}>0$) and the cell is overloaded. Note that the values of $\gamma_{m,n}(\beta_{m,n})$, $\rho_{m,n}(\beta_{m,n})$ and $T_u(\mathbf{B})$ correspond to the averages obtained during the time window ΔT between discrete times t and $t+1$.

3.2 Policy learning process

The DQN agent dynamically learns the decision-making policy π used to select the different actions based on the rewards obtained from previous decisions. This is done by means of the DQN algorithm of [9] particularised to the state, action and reward signals presented above. In summary, the algorithm aims at finding the optimal policy that maximises the discounted cumulative expected reward by approximating the optimum action-value function with a deep neural network (DNN) denoted as $Q(\boldsymbol{s}, \boldsymbol{a}, \theta)$, where $\boldsymbol{s}$ is the observed state, $\boldsymbol{a}$ is one of the possible actions that can be selected and θ are the weights of the interconnections between the different neurons in the DNN. Given the DNN, the decision making policy consists in selecting the action a with the highest value of $Q(\boldsymbol{s}, \boldsymbol{a}, \theta)$ for a given state.

The decision-making policy is updated progressively by modifying the weights θ based on the experiences gathered by the DQN agent. For this purpose, at a certain time t the DQN agent observes the state of the environment $\boldsymbol{s}(t)$ for a given UE and it triggers an action by selecting with probability 1- ε the action $\boldsymbol{a}(t)$ with the highest value of $Q(\boldsymbol{s},\boldsymbol{a},\theta)$ and with probability ε a random action. As a result, the DQN agent gathers the obtained reward and the new state at time $t+1$ and stores this experience (i.e., $\boldsymbol{s}(t)$, $\boldsymbol{a}(t)$, $r(t+1)$, $\boldsymbol{s}(t+1)$) in an experience dataset. The information collected in this dataset is then used to update the weights θ of the DNN using the expressions detailed in [9].

4 Performance evaluation

This section evaluates the performance of the proposed solution by means of system level simulations.

4.1 Scenario description

The considered scenario is a square area of 500 m x 500 m composed by four 5G NR cells and two LTE cells. The relevant parameters of the cells are presented in Table 1. The scenario assumes a non-homogeneous traffic distribution with MC-capable UEs moving at 1 m/s along the scenario and have an active session during the whole simulation duration with a required bit rate R_u=50 Mb/s. The candidate cells of the UEs can connect to M=2 RATs and N=2 cell per RAT, and the maximum number of cells that the UE can be connected to using MC is N_{max} =2. Additional background traffic is considered, with UEs generating Poisson session arrivals with aggregate generation rate 0.8 sessions/s and exponentially distributed session duration with average 120s. A background UE remains static during a session. 50% of the background UEs are randomly located inside a square hotspot of 250 m x 250 m centred at the middle of the scenario. The rest of background UEs are randomly distributed in the whole scenario. Background UEs connect to the RAT/cell with the highest Signal to Interference and Noise Ratio (SINR). To capture the different bit rates achievable by the two technologies, when a background UE is connected to LTE, its serving cell allocates the needed resource blocks to achieve a bit rate of 2.5 Mb/s, and when it is connected to 5G NR, the allocation is to achieve a bit rate of 40 Mb/s.

The DQN model parameters are detailed in Table 2. The DQN model has been developed in Python using the *TF-agents* library [10].

Table 1. Cell configuration parameters

Parameter	**Value**	
Type of RAT	LTE	5G NR
Cells position [x, y] m	[62, 250] [437,250]	[187, 125] [187,375] [312,125] [312,375]
Frequency	2100 MHz	26 GHz
Subcarrier separation	15 kHz	60 kHz
Nominal channel bandwidth	20 MHz	50 MHz
Number of available PRBs	100	66
Base station transmitted power	49 dBm	21 dBm
Base station antenna gain	5 dB	26 dB
Base station height	25 m	10 m
UE antenna gain	5 dB	10 dB
Overload threshold ρ_{max}	0.95	0.95
UE noise figure	9 dB	
UE height	1.5 m	
Path loss model	Model of Sec 7.4 of [11]	
w_{max}	95.04 MHz (corresponds to the case when MC is done with 2 cells of 5G NR)	

Table 2. DQN algorithm parameters

Parameter	Value
Initial collect steps	5000
Number of policy updates during learning	1e6
Experience Replay buffer maximum length (*l*)	1e5
Mini-batch size (*J*)	256
Discount factor(γ)	0.9
Learning rate (τ)	0.0003
ε value (ε-Greedy)	0.1
DNN architecture	Input layer: 17 nodes Two hidden layers: 100 and 50 nodes Output layer: 58 nodes
Time window (*ΔT*)	1 sec
Granularity $\Delta\beta$	0.1

4.2 Training Evolution

The training process of the DQN algorithm is performed by considering a MC-capable UE moving along the scenario following trajectories according to random walk and with required bit rate R_u=50 Mb/s, while at the same time background UEs also generate traffic as explained in section 4.1. The DQN agent decides the MC connectivity configuration of the UE and, based on the obtained rewards, the decision making policy is progressively updated as explained in section 3.2. In order to illustrate this learning process, the policies that are obtained every 2500 weight updates (i.e. training steps) are applied to an evaluation scenario in which an illustrative MC-capable UE follows a specific trajectory of duration 400 seconds, starting from point [X_1=50, Y_1=300] and following a straight trajectory up to the point [X_2=450, Y_2=300] at 1 m/s. Fig. 1 presents the evolution of the average reward obtained with the application of these policies as a function of the number of training steps. The results of Fig. 2 show that as the number of training steps increase average reward values tend to increase until 40x10^4 training steps, when the average reward values stabilize.

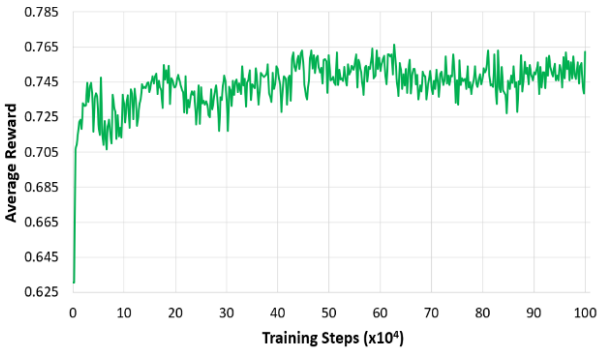

Fig. 2. Evolution of the average reward as a function of the training steps

4.3 Performance evaluation of the DQN-based strategy

In order to assess the benefits brought by the proposed DQN-based MC strategy, this sub-section compares the performance obtained by the proposed approach against two benchmarking approaches, namely the *optimum strategy* and the *SINR-based strategy*. The former consists of applying an exhaustive search process to select in each time step the MC configuration with the maximum reward, while the later considers that all the traffic of a UE is served by the cell with the highest SINR.

The comparison is performed by simulating a UE of interest following one hundred different trajectories of duration 400 s in the evaluation scenario and applying in each time window the MC configuration according to each of the evaluated schemes. For the DQN approach, the results correspond to the policy learnt by the DQN agent after a training consisting of 1E6 policy updates according to the procedure of Section 3.2.

Fig. 3 shows the obtained average reward for each one of the trajectories with all the considered strategies. It is observed that the DQN-based strategy achieves a performance very close to the optimum one in all the studied cases, which confirms the good behavior of the proposed approach. In turn, Fig. 3 also shows that the DQN-based strategy outperforms the classical SINR-based strategy in all the studied cases thanks to the better distribution of the traffic of the UE among the cells that avoids overload and enhances the obtained bit rate.

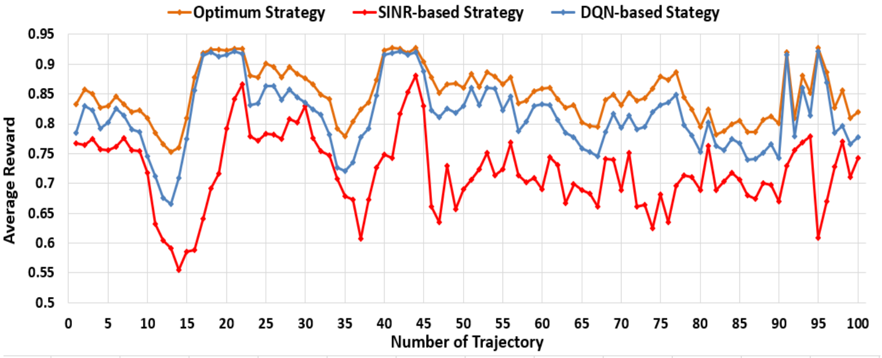

Fig. 3. Average reward for different trajectories.

4.4 Analysis of the robustness of the learnt DQN-policy

This section aims at evaluating the robustness of the DQN-policy when it is applied under conditions that differ from the ones that were considered during the training. For this purpose, considering that the training process has been done with a required bit rate R_u=50 Mb/s, the following results assess the generalization capability of the learnt policy when it is applied to different R_u values ranging between 15 and 65 Mb/s.

As a relevant metric for this assessment, a DQN-policy efficiency metric is considered, defined as the ratio between the reward of the DQN policy and that of the optimum strategy. Fig. 4 depicts the DQN-policy efficiency as a function of the required bit rate R_u value. The results of policy efficiency correspond to the average for the one hundred trajectories considered in the study. For the R_u value of 50 Mb/s that was considered during the training it is observed in Fig. 4 that the efficiency of the original policy is around 95.75%. Then, it tends to decrease for higher and lower values of required bit rate. In fact, it is worth mentioning the effects of decreasing R_u, because even when the amount of required radio resources is less, the efficiency losses are higher. For example, the policy efficiency with R_u=15 Mb/s is 13.42% less than with 50 Mb/s. From the red line in Fig. 4, we can realize that efficiency variations lower than 1% are observed when the R_u value changes from 40 Mb/s up to 65 Mb/s, i.e. -10/+15 Mb/s with respect to the value used for training. This reflects that the learnt DQN policy is robust in front of variations of around 20-30% of the R_u value used in the training.

Based on these results, another training process has been conducted with the same parameters of Table 2 but now changing the R_u value during training. Specifically, R_u at the beginning of the training is 50 Mb/s and then it is changed between 5 Mb/s and 60 Mb/s with steps of +/- 10%. Moreover, the number training steps has been increased up to 3E6 in order to account for a higher number of possible situations to learn. The green line in Fig. 4, shows the obtained efficiency with the new learnt policy (denoted as retrained policy) in comparison to the original policy. It is clearly observed that the new policy achieves a good efficiency for all the considered RBR values.

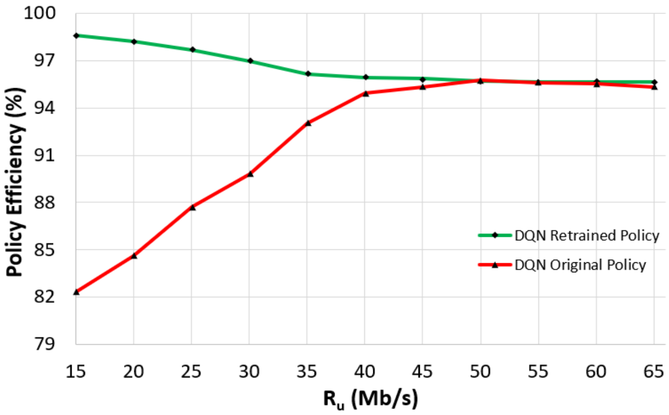

Fig. 4. DQN-Policy efficiency for different required bit rate (original vs retrained)

5 Conclusion

This paper has presented a novel approach based on Deep Q- Network for splitting the traffic of a UE among cells when using multi-connectivity depending on the current traffic and radio conditions experienced by the UE in the involved cells. The strategy intends to minimize the bandwidth consumption, the overload situations in the cells and enhancing throughput. The proposed strategy has been evaluated and compared against the optimum case and against a classical SINR-based approach. Results have shown the capability of the DQN agent to learn a quasi-optimal policy that in certain conditions outperforms the SINR-based approach in up to 33% in terms of reward, obtaining as a result better throughput performance with an optimized bandwidth assignment.

This paper has also analyzed the robustness of the learnt policy when being applied with a required bit rate value different than the one that was considered during the training stage. It has been observed that the learnt policy is able to work properly with variations of the required bit rate of around 20%-30% of the value considered in the training. In turn, by conducting a training that considers a wider range of values of required bit rate, it is possible to increase the performance of the obtained policy.

Acknowledgement

This paper is part of ARTIST project (ref. PID2020-115104RB-I00) funded by MCIN/AEI/10.13039/ 501100011033. The work is also funded by the Spanish Ministry of Science and Innovation under grant ref. PRE2018-084691.

References

1. A. Maeder et al (2016) “A Scalable and Flexible Radio Access Network Architecture for Fifth Generation Mobile Networks”, *IEEE Communications Magazine*, November. doi: 10.1109/MCOM.2016.1600140CM
2. E.Dahlman, S. Parkvall, J. Sköld (2018) "5G NR the Next Generation Wireless Access Technology", Academic Press Elsevier.
3. M. Yan, G. Feng, J. Zhou and S. Qin (2018) "Smart Multi-RAT Access Based on Multiagent Reinforcement Learning," in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4539-4551. doi: 10.1109/TVT.2018.2793186
4. V. F. Monteiro, et al (2019) “Distributed RRM for 5G Multi-RAT Multiconnectivity Networks”, IEEE Systems Journal, Vol. 13, No. 1, March. doi: 10.1109/JSYST.2018.2838335
5. M. Gerasimenko et al (2017) “Adaptive Resource Management Strategy in Practical Multi-Radio Heterogeneous Networks”, IEEE Access, February. doi: 10.1109/ACCESS.2016.2638022
6. P. K. Taksande, A.Roy, A. Karandikar (2018) “Optimal Traffic Splitting Policy in LTE-based Heterogeneous Network”, *IEEE Wireless Communications and Networking Confernece (WCNC)*. doi: 10.1109/WCNC.2018.8377096
7. B. Zhang et al (2019) “Goodput-Aware traffic Splitting Scheme with Non-ideal Backhaul for 5G-LTE Multi-Connectivity”, *IEEE Wireless Communications and Networking Confernece (WCNC)*. doi: 10.1109/WCNC.2019.8885728
8. J. Elias, F. Martignon and S. Paris (2021) "Optimal Split Bearer Control and Resource Allocation for Multi-Connectivity in 5G New Radio", Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), pp. 187-192. doi: 10.1109/EuCNC/6GSummit51104.2021.9482505
9. V. Mnih, et al (2015) “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533.
10. S. Guadarrama et al (2018), TF-Agents: A Library For Reinforcement Learning in TensorFlow.
11. 3GPP TS 38.901 v16.1.0 (Dec. 2019) "Study on Channel Model for Frequencies From 0.5 to 100 GHz (Release 16)".