Hierarchical RL for Load Balancing and QoS Management in Multi-Access Networks

Antonio Ornatelli*
Department of Computer, Control and Management Engineering "Antonio Ruberti"
Sapienza University of Rome
ornatelli@diag.uniroma1.it

Andrea Tortorelli
Department of Computer, Control and Management Engineering "Antonio Ruberti"
Sapienza University of Rome
tortorelli@diag.uniroma1.it

Alessandro Giuseppi
Department of Computer, Control and Management Engineering "Antonio Ruberti"
Sapienza University of Rome
giuseppi@diag.uniroma1.it

Francesco Delli Priscoli
Department of Computer, Control and Management Engineering "Antonio Ruberti"
Sapienza University of Rome
dellipriscoli@diag.uniroma1.it

***Abstract* — This paper deals with the problem of resource management in Multi-Access Networks. A Reinforcement Learning based hierarchical control strategy is presented. The main contribution of the proposed approach is its capability of simultaneously tacking the load balancing and QoS management problems in a scalable, dynamic and closed-loop way. The effectiveness of the proposed solution has been proved in a specific case study in the context of which the performances of the proposed algorithm have been compared with a standard load balancing controller.**

Keywords — Load balancing; Reinforcement Learning; Multi-Access Networks.

I. INTRODUCTION

This paper deals with load balancing algorithms aiming at maximizing the exploitation of the air interface of a 5G multi-access network. The air interface band, used to link Mobile Terminals (MTs) and Access Points (APs), is a valuable and limited resource. Multi-access networks empower the *always best-connected* concept [1] consisting in the capability of providing users with the best available connection in heterogenous scenarios involving several Radio Access Networks (RANs) and Technologies (RATs). In this context, load balancing algorithms are aimed at optimizing network resources' usage by taking in consideration users' requirements, in terms of Quality of Service (QoS) parameters, and network conditions, in terms of, e.g., APs' congestion levels and power consumption.

In 5G networks, the flow of packets relevant to a given connection can be dynamically split into several sub-flows that follow different paths (i.e., network accesses) and are recombined at the destination [2], [3]. As an example, consider the uplink flow of a connection originated by a given MT which, at a given time k, can transmit to three different APs possibly belonging to different RATs (e.g., 4G, 5G, satellite, Wi-Fi). Said flow can be split into three sub-flows each directed to one of the available APs. This feature allows to match the challenging 5G requirements in terms of QoS parameters by exploiting all the available RATs and APs. However, to fully exploit this heterogeneous set of resources, it is necessary to develop intelligent, dynamic and scalable resource allocation strategies.

Motivated by these considerations, the present work is aimed at developing an efficient and scalable load balancing strategy aimed at maximizing the exploitation of the Air Interface (AI) of a 5G multi-access network. In particular, the proposed solution adopts a reinforcement learning based hierarchical control architecture allowing to dynamically select the most appropriate routing over the air interface of the packets relevant to connections in progress while respecting QoS requirements of said connections.

The reminder of the paper is organized as follows: Section II presents a review of the literature with respect to multi-access networks and reinforcement learning techniques in the context of load balancing; Section III describes the proposed control architecture and introduces the mathematical formalization of the considered problem; Section IV describes the adopted control strategy; Section V presents the considered case study and is devoted to validate the proposed approach; in Section VI the presented results are wrapped up and future developments discussed.

II. STATE OF THE ART

Multi-access networks allow reliable communications, and increased coverage and hand-hover management capabilities in line with the challenging requirements of the fifth-generation mobile network [4]–[7]. Many approaches have been proposed to efficiently implement these features and, more specifically, to tackle the RAN selection problem. Instances of said approaches belongs to utility theory, multi attributes decision making, fuzzy logic, game theory, combinatorial optimization, Markov chains or a mix of these [8]–[11]. In [7], the authors provide a comprehensive analysis of the mentioned approaches and classify them into two groups namely i) fast and easy to implement but typically static, open-loop, centralized and with lower precision methods and ii) slower and complex to implement, but dynamic, closed-loop, distributed and high precision methods. To the first group belong approaches based on utility theory, multi attributes decision making, and fuzzy logic while to the second group belong approaches based on game theory, combinatorial optimization and Markov chains.

The approach proposed in this paper has the ambition to exploit the advantages of the second group of methods (i.e., the dynamic, closed-loop and distributed nature) while guaranteeing

* Corresponding author

This work has been carried out in the framework of the 5GALLSTAR project, which has received funding from the European Union Horizon 2020 EU-Korea programme under grant n. 815325 and is supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT No. 2018-0-00175, 5G AgiLe and fLexible integration of SaTellite And cellulaR). This work reflects only the author's view. The EU Commission, the Research Executive Agency and the IITP are not responsible for any use that may be made of the information it contains.

high scalability. To achieve this, a hierarchical control architecture has been developed where part of the control (and, thus, of the computational cost) is demanded to local distributed controllers each independently solving a Reinforcement Learning (RL) problem. In a RL problem, an agent (i.e., the entity taking decisions) interacts with the environment and receives a reward based on the action performed and environment's state. Based on said reward, the agent is able to learn the actions maximizing the expected reward given the environment states [12]–[16]. The main difference between RL and other machine learning methods is that the learner is not told which actions to take but instead must discover which actions yield the most reward by trying them out directly in the target environment.

RL problems are typically formulated as Markov Decision Processes (MDPs) which can be modelled as a tuple (S, A, δ, r) where

- S is a finite set of states,
- A is a finite set of actions,
- $\delta = P(S \times A \times S)$ is a probability distribution over state transitions and
- $R(s,a): S \times A \to \mathbb{R}$ is the reward function associating a scalar value to state-action pairs.

From the interaction with the environment, the agent has to learn the optimal policy π i.e., the mapping between states $s \in S$ and actions $a \in A$ which when implemented guarantees the maximum expected value of the cumulative reward. In other words, a RL agent is able to decide the actions to be taken maximizing the long-term effect of actions. The optimality of a given policy π can be evaluated considering the state value function $V^{\pi}(s)$ which, at a given time t, provides a measure of the cumulative reward that can be obtained following the policy π starting from state s; the state value function is thus defined as

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s\right] \quad (1)$$

where γ and R are the discount factor weighting future rewards and the rewards, respectively. Similarly, it is possible to define the policy Q-Function (or state-action value function) as

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = 0\right] \quad (2)$$

In other words, the Q-Function allows to quantify the cumulative reward that the agent receives given the current state s_t and action a_t following policy π. The optimal policy π^* can be thus defined as

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^{\pi}(s), \qquad \forall s \in S \quad (3)$$

Many methods have been developed to solve the above described RL problem. When the environment is perfectly known, dynamic programming methods proved to be very efficient. In the scenario considered in this paper, however, it is assumed that the agent does not have a complete and accurate model of the environment and thus must learn from the interactions with the environment itself. In this case, Temporal Difference (TD) methods such as Q-Learning and SARSA proved to be very effective. Indeed, such model-free methods, allow the agent to learn the optimal value functions and thus, in turn, the optimal policy π^* [13], [17].

III. PROBLEM FORMALIZATION

The load balancing problem considered in this paper concerns the dynamic allocation of connections to the available APs considering QoS constraints. With *connection*, it is meant a mono-directional flow of packets characterized by the same origin and the same destination. Each connection is assumed to have a specific *QoS profile* (inherited by the packets of said connection) consisting of a set of constraints regarding, for example, the throughput, latency, Bit Error Ratio (BER), jitter, mobility and so on.

Variable	*Meaning*
$\boldsymbol{i}$	Index used to identify APs and, consequently, cells
$\boldsymbol{I}$	Number of Local Controllers and, consequently, APs coordinated by the Global Controller
$\boldsymbol{p}$	Index used to identify QoS profiles
$\boldsymbol{P}$	Number of QoS profiles considered by the Global Controller
$\boldsymbol{Y_p}$	Minimum bit rate to be associated to QoS profile $\boldsymbol{p}$
$\boldsymbol{c}$	Index used to identify connections
$\boldsymbol{C_p(k)}$	Total number of connections of profile $\boldsymbol{p}$ active in the Global Controller Area
$\boldsymbol{A_{p,c}(k)}$	Set of APs whose cells cover the MT involved in connection $\boldsymbol{c}$ belonging to profile $\boldsymbol{p}$
$\boldsymbol{x_{i,p}(k)}$	Target bit rate assignment computed by the i-th local controller for connections belonging to profile p
$\boldsymbol{y_{i,p,c}(k)}$	Actual bit rate assigned from the Global Controller to the $\boldsymbol{i}$-th local controller for connection $\boldsymbol{c}$ belonging to profile $\boldsymbol{p}$
$\boldsymbol{z_{i,p}(k)}$	Actual bit rate assigned from the Global Controller to the $\boldsymbol{i}$-th local controller for all connections belonging to profile $\boldsymbol{p}$
$\boldsymbol{L}$	Discrete number of traffic levels
$\boldsymbol{\Lambda}$	Discrete number of target bit rate assignments levels
$\boldsymbol{j}$	Index used to identify MTs
$\boldsymbol{J}$	Number of MTs in the Global Controller Area

Table 1. Nomenclature

To address this problem, a hierarchical control architecture has been envisaged (see Figure 1). The underlying idea is to associate a *Local Controller* to each AP and, consequently, to the cell covered by said AP. A set of local controllers willing to share their resources, referred to as *Local Controller Sharing Set*, are then coordinated by a single *Global Controller* located in the Cloud RAN. The cells covered by the APs associated to the local controllers in the Local Controller Sharing Set define the *Global Controller Area*. From now on, the symbol i will be used to refer to a generic AP in a given Local Controller Sharing Set and I will denote the number of Local Controllers coordinated by a given Global Controller.

Concerning QoS profiles, it is assumed that in a pre-operational phase the most suitable number of profiles P has been identified. Such operation can be performed, for instance, using k-means algorithms exploiting available historical data sets. The number of QoS profiles should be identified trading off QoS personalization (increasing the number of profiles) and complexity (reducing the number of profiles). Profiling is expected to provide a set of classification rules allowing,

whenever a new connection c is triggered, to classify this connection as belonging to a given profile p. A fundamental QoS constraint characterizing a given profile p relates to the minimum bit rate Y_p which, in any traffic condition, must be assigned to any connection belonging to the profile p.

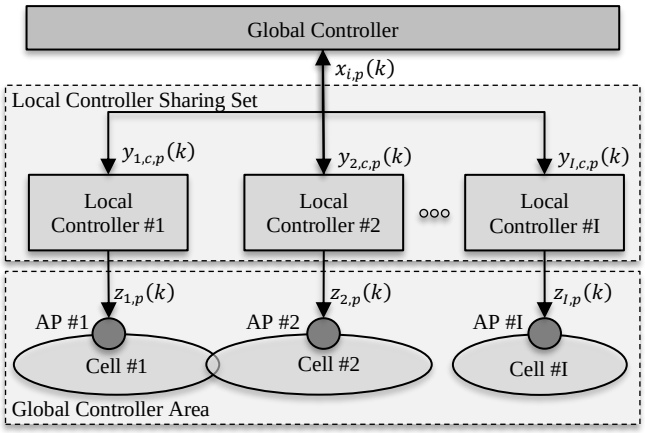

Figure 1. Control Architecture

At each discrete time k, the following variables are of interest for the considered problem:

- $A_{p,c}(k)$ denoting the set of APs whose cells cover a given MT involved in a connection c belonging to profile p; note that it is assumed that the Global Controller is aware of (i) the profile of each in-progress connection in the Global Controller Area and (ii) the sets of APs covering each connection c of any profile p (i.e., the Global Controller has a complete view of the sets $A_{p,c}(k)$ for all c and p);
- $x_{i,p}(k)$ representing the target bit rate assignment computed by the i-th local controller for connections belonging to profile p (i.e., represents the sum of bit rates that should be assigned to connections of profile p served at time k by the i-th AP); such target value, transmitted to the Global Controller, is computed independently by each local controller based on the performances experienced by all the in-progress connections served by the i-th AP;
- $y_{i,p,c}(k)$ representing the actual bit rate assigned from the Global Controller to the i-th local controller for connections belonging to profile p; note that, at any time, the actual bit rate $y_{i,p,c}(k)$ assigned by the Global Controller to the i-th Local Controller for connection c of profile p must be higher of Y_p;
- $z_{i,p}(k)$ denoting the overall bit rate assigned from the Global Controller to the i-th local controller for all connections belonging to profile p.

Note that the goal of the Global Controller is to compute the actual bit rates to be assigned to the local controllers in such a way that they approach, as far as possible, the target bit rates computed by the local controllers (i.e., ideally it should be $z_{i,p}(k) \cong x_{i,p}(k)$ since the latter tries to optimize the i-th AP's capacity utilization from a local perspective).

IV. CONTROL STRATEGY

A. Local Controllers

The proposed control strategy demands part of the computation to the Local Controllers. As anticipated, each local controller solves, independently, a RL problem for addressing the load balancing problem and transmit to the Global Controller the target bit rate assignment $x_{i,p}(k)$. In the following, the RL problem that each local controller has to solve will be formalized. For sake of clarity, only the uplink (i.e., the flow of packets from the MT to the AP) will be considered; the extension to downlinks is straightforward.

1) State space modelling

The state of a generic AP is defined as the measured *traffic level* for each profile p. Said traffic level is assumed to be described by L discrete levels. By doing so, the *state* of each local controller $s_i(k) \in \mathbb{R}^P$ can be described as a row vector

$$s_i(k) = [l_{i,1} \quad \dots \quad l_{i,P}] \in \mathbb{R}^P \tag{4}$$

where the generic scalar entry $l_{i,p}$ of the state represents the traffic level experienced by the generic AP for connections of profile p.

Note that the state space, as it has been defined, guarantees high flexibility since it is possible to trade-off state's description capabilities and computational costs by increasing or decreasing the number of discrete levels L, respectively. Indeed, with this modelling choices, the number of possible states is equal to L^P which is a relatively small number.

As a final remark, note that it is possible to generalize the proposed formulation by considering different discrete traffic levels for each profile (in this case, of course, one should define the number of discrete levels L_p for each profile p).

2) Action space modelling

From the problem formulation (see Section **Errore. L'origine riferimento non è stata trovata.**), it follows that the control variables are the target bit rate assignments $x_{i,p}(k)$ computed by the local controllers. In order to

- induce a smooth convergence between the target and actual bit rate assignments, i.e., $x_{i,p}(k)$ and $z_{i,p}(k)$, respectively, and
- reduce the dimension of the local controllers' action spaces,

instead of directly considering the target bit rate assignments $x_{i,p}(k)$ it is possible to consider as *control actions* $a_i(k) \in \mathbb{R}^P$ the following row vectors:

$$a_i(k) = [\lambda_{i,1} \quad \dots \quad \lambda_{i,P}] \in \mathbb{R}^P \tag{5}$$

where the generic scalar entry $\lambda_{i,p}$ represents the *target variation*, with respect to the previous discrete time instant, of the bit rates assigned from the i-th local controller to connections of profile p; furthermore, such variations are limited to a small number of discrete levels Λ. With this modelling choices, the total number of possible states is Λ^P which is a relatively small number. Note that it is possible to generalize the proposed formulation by considering different discrete traffic levels for each profile (in this case, of course, one should define the number of discrete levels Λ_p for each profile p).

3) Rewards shaping

The objective of the control strategy proposed consists in (i) keeping each cell, as far as possible, far from congestion for any profile p and (ii) assuring that the cell's capacity is exploited. Following on these considerations, it is possible to consider *rewards* depending on the state and on the profile as follows:

$$r_{i,p}(k) = b_{i,p}\left(s_p(i)\right) \tag{6}$$

where $s_p(i)$ is the generic entry of the i-th local controller state (see equation (5)) and $b_{i,p}(\cdot)$ are functions of the state shaped in such a way to provide hard penalizations to congested states and mild penalization to idle states.

B. Global Controller

The control problem, from the Global Controller point of view, consists in providing to the local controllers the actual bit rate assignments $y_{i,p,c}(k)$ (i.e., for each local controller i and each (c,p) couple connection-profile) based on the received target variations of bit rate assignments $\lambda_{i,p}$ (see equation (5)).

Hence, the Global Controller must minimize the following performance index $J(k)$

$$\begin{aligned} J(k) &= \sum_{i=1}^{I}\sum_{p=1}^{P}\sum_{c=1}^{C_p(k)}\left(y_{i,p,c}(k) - x_{i,p}(k)\right)^2 \\ &= \sum_{i=1}^{I}\sum_{p=1}^{P}\left(z_{i,p}(k) - x_{i,p}(k)\right)^2 \end{aligned} \tag{7.1}$$

while guaranteeing, for each local controller i and each connection-profile couple, that

$$\sum_{i\in A_{p,c}(k)} y_{i,p,c}(k) \geq Y_p \tag{7.2}$$

$$y_{i,p,c}(k) = 0 \text{ if } i \notin A_{p,c}(k) \tag{7.3}$$

$$y_{i,p,c}(k) \geq 0 \tag{7.4}$$

where

- (7.1) is the performance index to be minimized; note that said index is lower when the target and actual bit rate assignments are closer;
- (7.2) allows to satisfy QoS constraints in terms of the minimum bit rate required by each profile p (the structure of equation (7.2) can be replicated to take into account additional QoS constraints);
- (7.3) specifies that, if the i-th AP does not cover the connection c of profile p at time k, the Global Controller cannot assign a bit rate to said connection-profile couple;
- (7.4) specifies that the actual bit rate assignments cannot be negative.

The actual bit rate assignments $y_{i,p,c}(k)$ computed by the Global Controller as output of the optimization problem (7.1) – (7.4), are transmitted at each discrete time k to the MTs through the serving APs. Said MTs can transmit toward the APs at a bit rate $T_{i,p,c}(k)$ which is not higher than $y_{i,p,c}(k)$.

V. Simulation Results

A. Case study

The considered case study concerns load balancing and mobility management. More in detail, the problem consists in associating APs and moving MTs taking into account spatial considerations. Hence, the goal is to allocate enough bandwidth to the MTs and to exploit as much as possible APs capacities (avoiding congestions) while considering the distance between given APs and MTs. In other words, a given AP should assign less bandwidth to MTs whose distance is higher than a fixed threshold. This condition allows to reduce the power consumption required for transmission and, at the same time, to minimize the probability that an AP assigns bandwidth to a MT which is likely to exit from its coverage area.

To simultaneously tackle the load balancing and mobility management problems, the local controllers' state defined in Section III is augmented and has two components:

- the first component, $s_{i,1}$, represents the distance between the i-th AP and the MTs
- the second component, $s_{i,2}$, represent the traffic level as defined in equation (4)

B. Scenario description

The considered scenario envisages the presence of two MTs exploiting the resources provided by four cells (i.e., four APs and, consequently, four local controllers). The two moving MTs are assumed to belong to two different QoS profiles (hence, the number of QoS profiles considered by the Global Controller is $P = 2$). In other words, it is assumed that the two moving MTs have different QoS requirements due to their different motion characteristics. In this case, said requirements can be referred to as *QoS mobility profiles* since the MTs' motion degrades several QoS indicators such as error rate, energy consumption and service continuity. Note that this scenario can be generalized by considering, instead of single MTs, clusters of moving MTs involving several connections. Indeed, nowadays, vehicles are equipped with a wide set of sensors for guidance support, multimedia systems and, also, passengers' terminals. The local controllers can consider all the connections in the same vehicle (cluster) as belonging to the same QoS mobility profile p.

As depicted in Figure 2, it is assumed that the two moving MTs (represented as a white and a black car) change their coverage area during the simulations. More in detailed, for $k = 1, \ldots, 200$ the black car is covered by the cells associated to APs #1 and #3 while the white car is covered by the cells associated to APs #2 and #3 and, for $k > 200$, vice versa.

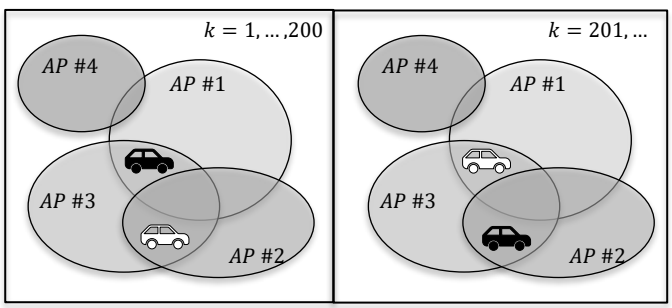

Figure 2. APs' coverage and MTs' position (left: $k \leq 200$; right: $k > 200$)

Concerning the local controllers' state definition, let

- $\rho^{\text{distance}}_{\text{high}\ i,p}$ and $\rho^{\text{distance}}_{\text{low}\ i,p}$ be two fixed thresholds specifying the maximum distance, between the i-th AP and given MT;
- $\rho^{\text{traffic}}_{\text{high}\ i,p}$ and $\rho^{\text{traffic}}_{\text{low}\ i,p}$ be two fixed thresholds specifying the traffic level above which the i-th local controller is considered overloaded with respect to connections of profile p and vice versa, respectively.

Said thresholds allow to define a limited set of discrete levels for characterizing the local controllers' congestion level (i.e., used to solve the load balancing problem) and the convenience for a given local controller to serve a generic connection-profile couple (i.e., used to perform mobility management).

On the ground of these considerations, it is possible to define the generic entries of the two components of the augmented local controllers state ($d^{(j)}_{i,p}(k)$ and $l_{i,p}(k)$ respectively) as

$$d^{(j)}_{i,p}(k) = \begin{cases} 0 & \text{if } \delta^{(j)}_{i,p}(k) < \rho^{\text{distance}}_{\text{low}\ i,p} \\ 1 & \text{if } \rho^{\text{distance}}_{\text{low}\ i,p} \le \delta^{(j)}_{i,p}(k) \le \rho^{\text{distance}}_{\text{high}\ i,p} \\ 2 & \text{if } \delta^{(j)}_{i,p} > \rho^{\text{distance}}_{\text{high}\ i,p} \end{cases} \quad (8)$$

$$l_{i,p}(k) = \begin{cases} 0 & \text{if } t_{i,p}(k) < \rho^{\text{traffic}}_{\text{low}\ i,p} \\ 1 & \text{if } \rho^{\text{traffic}}_{\text{low}\ i,p} \le t_{i,p}(k) \le \rho^{\text{traffic}}_{\text{high}\ i,p} \\ 2 & \text{if } t_{i,p} > \rho^{\text{traffic}}_{\text{high}\ i,p} \end{cases} \quad (9)$$

where $d^{(j)}_{i,p}(k)$ and $l_{i,p}(k)$ represent the i-th local controller state with respect to the mobility management and load balancing problems, respectively, and $\delta^{(j)}_{i,p}(k)$ and $t_{i,p}(k)$ are the relative distance between the i-th local controller's AP and the j-th MT and the traffic level experienced by the i-th local controller with respect to connections of profile p, respectively.

The rewards defined in equation (6) can be particularized for the considered scenario for taking into account the augmented state defined in equations (8)-(9) as follows:

$$r_{i,p}(k) = \sum_{j=1}^{J} \frac{K_1}{1 + e^{\alpha_j * d^{(j)}_{i,p}(k)}} * t_{i,p} + \frac{K_2}{1 + e^{-\beta * l_{i,p}(k)}} \quad (10)$$

where K_1 and K_2 are two constants used to weight the two state's components and α_j and β are positive constants. Rewards have been shaped using sigmoid functions since this function, by properly setting α and β, is able to represent the effect of the distance and traffic level on the system performances. This is possible thanks to the sigmoid structure which allows the definition of three input variable intervals (characterized by α and β) returning low, medium, or high values of the output (i.e., the reward value).

Concerning the action space, it is assumed that the i-th local controller can either (i) increment the bandwidth allocated to connections of profile π of a positive constant Δ, (ii) do not vary the allocated bandwidth or (iii) decrement the allocated bandwidth of $-\Delta$. It follows that the dimension of the action space is $\Lambda^P = 9$ since there are $P = 2$ QoS (mobility) profiles and $\Lambda = 3$ discrete levels of allocated bandwidth variations (i.e., $+\Delta, 0, -\Delta$).

C. *Simulations and results*

Simulations were performed on a laptop equipped with an Intel i5 processor and 12GB RAM.

In the simulations, the proposed RL-based hierarchical control strategy is compared with a Nearest Not-Full (N-NF) controller allocating bandwidth to the MTs through the nearest, not congested, AP.

Simulations show that the proposed hierarchical control strategy is able to effectively tackle both the load balancing and mobility management problems. More in detail, concerning load balancing, from Figure 3 it is clear that the proposed RL-based approach guarantees a fair load distribution between APs while the N-NF approach under/overloads them. Furthermore, the proposed RL approach is able to keep APs' loads in the optimal range defined by the upper and lower thresholds (i.e., $\rho^{\text{traffic}}_{\text{low}\ i,p}$ and $\rho^{\text{traffic}}_{\text{high}}$) which in the figure are represented by the green and red lines, respectively.

Concerning mobility management, when the two MTs change their coverage area (i.e., at $k = 200$), the N-NF approach experiences drastic changes in the allocated bandwidth (which translates in performing heavy handover procedures that degrades the system performances) while the proposed RL approach guarantees a smoother transition (see Figure 3). Said transition stops when the loads reach a new equilibrium in terms of loads and of the mutual distances between APs and MTs. Figure 4 shows the bandwidth allocation for the two MTs computed by the proposed RL and N-NF approaches. As it can be seen, both approaches vary the bandwidth allocation considering the relative position between the APs and MTs. However, the proposed RL approach is able to exploit all the available resources and to avoid congestions while respecting the QoS (mobility) constraints.

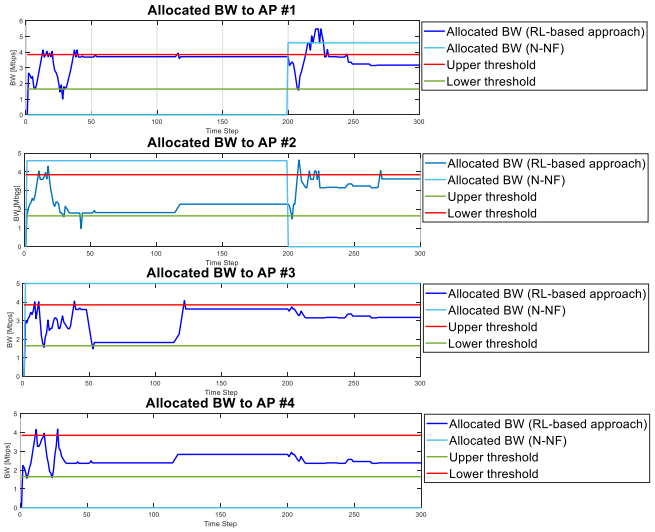

Figure 3. APs allocated bandwidth

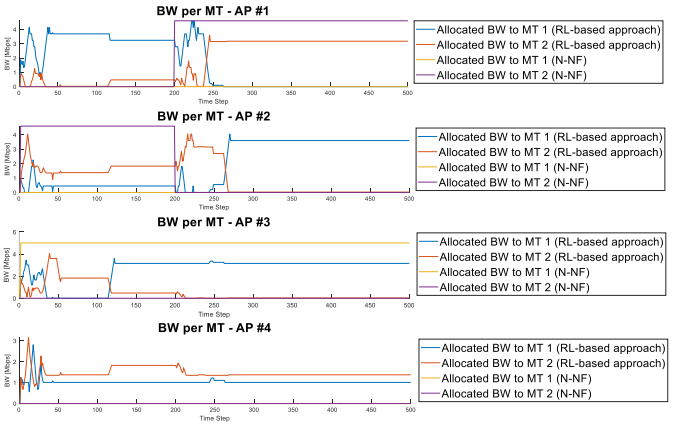

Figure 4: MTs allocated bandwidth

VI. CONCLUSIONS

In this paper, a RL-based hierarchical control strategy to simultaneously tackle the load balancing and QoS management problems in multi-access networks has been presented. The adoption of distributed RL agents (i.e., the local controllers), together with the proposed hierarchical control architecture, empowers the scalability of the proposed approach. Scalability is further guaranteed by means of the adoption of discrete levels used to reduce the state and action spaces.

The scalable, dynamic and closed-loop nature of the proposed control strategy has been validated in a specific use-case concerning moving (clusters of) MTs.

ACKNOWLEDGMENT

The authors wish to thank all the CRAT and all the members of the 5G-ALLSTAR project.

REFERENCES

[1] E. Gustafsson and A. Jonsson, "Always best connected," *IEEE Wirel. Commun.*, vol. 10, no. 1, pp. 49–55, 2003, doi: 10.1109/MWC.2003.1182111.

[2] C. Kilinc *et al.*, "5G Multi-RAT Integration Evaluations Using a Common PDCP Layer," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–5, doi: 10.1109/VTCSpring.2017.8108594.

[3] D. S. Michalopoulos, I. Viering, and L. Du, "User-plane multi-connectivity aspects in 5G," in *2016 23rd International Conference on Telecommunications (ICT)*, 2016, pp. 1–5, doi: 10.1109/ICT.2016.7500422.

[4] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015, doi: 10.1109/ACCESS.2015.2461602.

[5] A. Morgado, K. M. S. Hug, S. Mumtaz, and J. Rodriguez, "A survey of 5G technologies: regulatory, standardization and industrial perspectives," *Digit. Commun. Networks*, vol. 4, no. 2, pp. 87–97, 2018, doi: 10.1016/j.dcan.2017.09.010.

[6] S. Chandrashekar, A. Maeder, C. Sartori, T. Hohne, B. Vejlgaard, and D. Chandramouli, "5G multi-RAT multi-connectivity architecture," in *2016 IEEE International Conference on Communications Workshops (ICC)*, 2016, pp. 180–186, doi: 10.1109/ICCW.2016.7503785.

[7] A. Ravanshid *et al.*, "Multi-connectivity functional architectures in 5G," in *2016 IEEE International Conference on Communications Workshops (ICC)*, 2016, doi: 10.1109/ICCW.2016.7503786.

[8] L. Wang and G.-S. Kuo, "Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks — A Tutorial," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 271–292, 2013, doi: 10.1109/SURV.2012.010912.00044.

[9] Q. Song and A. Jamalipour, "Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques," *IEEE Wirel. Commun.*, vol. 12, no. 3, pp. 42–48, 2005, doi: 10.1109/MWC.2005.1452853.

[10] A. Al Sabbagh, R. Braun, and M. Abolhasan, "A comprehensive survey on rat selection algorithms for heterogeneous networks," *World Acad. Sci. Eng. Technol.*, vol. 73, pp. 141–145, 2011.

[11] A. Ornatelli, A. Tortorelli, and A. Giuseppi, "Iterative MPC for Energy Management and Load Balancing in 5G Heterogeneous Networks," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020, pp. 467–471, doi: 10.1109/UEMCON51285.2020.9298113.

[12] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Syst. Mag.*, vol. 12, no. 2, pp. 19–22, 1992, doi: 10.1109/37.126844.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, Second Edi. MIT Press, 2018.

[14] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, 2018, doi: 10.1109/TCCN.2018.2881442.

[15] F. L. Lewis and D. Liu, *Reinforcement learning and approximate dynamic programming for feedback control*. John Wiley & Sons, 2013.

[16] N. Vučević, J. Pérez-Romero, O. Sallent, and R. Agustí, "Reinforcement learning for joint radio resource management in LTE-UMTS scenarios," *Comput. Networks2*, vol. 55, no. 7, pp. 1487–1497, 11AD, doi: 10.1016/j.comnet.2010.12.029.

[17] C. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.