Traffic Steering and Network Selection in 5G Networks based on Reinforcement Learning

Alessandro Giuseppi*, Student *Member, IEEE,* Antonio Pietrabissa*, *Member, IEEE,*
Francesco Liberati, *Member, IEEE,* Roberto Germanà, and Francesco Delli Priscoli, *Member, IEEE*

Abstract— This paper presents a controller for the problem of Network Selection in 5G Networks, based on Reinforcement Learning. The problem of Network Selection and Traffic Steering is modeled as a Markov Decision Process and a Q-Learning based control solution is designed to meet 5G requirements, such as Quality of Experience (QoE) maximization, Quality of Service (QoS) assurance and load balancing. Numerical simulations preliminarily validate the proposed approach on a simulated scenario considered in the European project H2020 5G-ALLSTAR.

I. INTRODUCTION

TRAFFIC steering is a fundamental functionality of 5G networks and consists in the ability of routing, or *steering*, a given traffic flow over one of the several different Radio Access Technologies (RATs) available to the User Equipment (UE). In fact, in 5G scenarios it is envisaged that modern UEs may connect to several different technologies (e.g., 5G, LTE, satellite networks, …), potentially at the same time, in what is defined as the *heterogeneous network* framework. Traffic steering is then heavily linked to the concept of network selection, as the routing/steering decisions over the Access Points (APs) of the available RATs shall be driven by a feedback-based analysis of the network state and performances, potentially also taking into account user preferences.

In this paper, the state of the network will be represented by the downlink cell allocated bitrate, while the performances will be measured in terms of connection quality to capture the satisfaction level of the users. The traffic steering problem will be modeled with the Markov Decision Processes (MDP) formalism and Reinforcement Learning (RL) was selected for the controller design due to its ability to deal with complex scenarios without requiring an explicit model. The main motivation behind this work is then to characterize and show the effectiveness of such an approach through its application to a simulated 5G network scenario. For the model development, in order to account for services which can be offered with multiple bitrates and consequently with different qualities, several classes of utility functions were defined to capture the Quality of Experience (QoE) perceived by the users. Additionally, the MDP was formulated, to overcome the scalability problem, utilizing Approximate Dynamic Programming (ADP) techniques, and, in particular, state space aggregation.

The paper is organized as follows: Section II presents the state of the art on Traffic Steering in 5G-networks, as well as the proposed novelties; Section IV presents the problem modelling and the network selection algorithm; Section IV shows some simulation results to validate the proposed algorithm; Section V draws the conclusions and highlights future works.

The work presented in this paper was carried out within the H2020 5G-ALLSTAR project (www.5g-allstar.eu), aimed at ensuring the seamless integration of satellite and cellular connections in a heterogeneous network framework.

II. STATE OF THE ART, INNOVATIONS AND LIMITATIONS OF THE PROPOSED APPROACH

Traffic steering is the process of distributing traffic load in order to exploit the available network resources on a set of heterogeneous RATs that constitute a Radio Access Network (RAN) [1]. The enabling procedure for the process of traffic steering is the so-called *network selection* [2], [3], a feedback-based analysis of the network aimed at identifying the best APs for the connections, also referred to as Protocol Data Unit (PDU) sessions. Such feedback analysis is conducted based on the level of usage of the various APs, together with their characteristics (e.g., operating prices, reliability) and both user and operator preferences.

In 5G networks, the network selection shall be done in such a way that the Quality of Service (QoS) requirements of the

This work was supported by European Commission in the framework of the H2020 EU-Korea project 5G-ALLSTAR (5G AgiLe and fLexible integration of SaTellite And cellular, www.5g-allstar.eu/) under Grant Agreement no. 815323.

* A. Giuseppi and A. Pietrabissa are co-first authors, emails: {giuseppi,pietrabissa}@diag.uniroma1.it

A. Giuseppi, A. Pietrabissa, F. Liberati, R. Germanà and F. Delli Priscoli are with the Department of Computer, Control, and Management Engineering Antonio Ruberti, University of Rome La Sapienza, via Ariosto 25, 00162, Rome, Italy, and with the Space Research Group of the Consortium for the Research in Automation and Telecommunication (CRAT), via G. Nicotera 29, 00195, Rome, Italy.

various connections (e.g., minimum required bitrate or maximum tolerated delay) are satisfied, and, to univocally define such QoS requirements, the concept of QoS-flow was introduced [4]. Each PDU session is divided into several QoS-flows, each characterized by a standardized set of QoS requirements depending on its service characteristics [4], leading to the identification of, as of now, 22 different QoS flow types identified by a corresponding 5G QoS Identifier (5QI).

Several approaches were already studied for the problems of traffic steering and network selection, ranging from Multiple Attribute Decision Making (MADM) [5], [6] and Fuzzy Logic [7] to Game Theory [8], [9]. MDPs and RL were also already explored, for example, in [10] and [11].

An important feature for traffic steering in 5G networks is the ability to support multiple bitrates for the supported services. An example of such service is multi-codec video streaming, whose streaming quality should be dynamically varied based on user needs, preferences and on network conditions. As introduced in [12] for generic resource allocation problems, a possible approach is to associate utility functions to the services to model how the amount of assigned resources impacts on the user satisfaction.

From the modelling perspective, this work takes into account 5G requirements by supporting a heterogeneous network scenario while also being able to offer flexibility in service requirements. In this last respect, the network controller presented in this work aims at maximizing the perceived connection quality of the network users, thanks to an *ad hoc* utility function-based modelling of the performances of the considered classes of services, inspired by [12] and expanding the modelling of [11], which defined user satisfaction based on a throughput threshold. From the MDP and RL perspective, this work expands the algorithm presented in [10], thanks to the concepts of state aggregation and ADP, aimed at avoiding the so-called curse of dimensionality (e.g., see [13]), which affects the solutions based on Dynamic Programming (DP) and tabular RL in realistic scenarios.

The evaluation of RL approaches relies on the availability of a realistic environment, i.e., in our case, of a representative 5G network simulator. This paper presents a preliminary evaluation of the proposed concepts by using numerical simulations considering a limited set of 5G network characteristics. Moreover, by considering some of the neglected characteristics, such as, for instance, the user mobility, the developed RL algorithm would probably be not adequate but its ideas will serve as a starting point for other improvements, as specified in the future work part of Section V.

III. Preliminaries on Markov Decision Processes and Reinforcement Learning

A MDP is a discrete-time stochastic control process defined by the tuple $\{S, A, T, r, \Sigma, \gamma\}$, where S is a discrete, finite state set, $A = \bigcup_{s \in S} A_s$ is the finite action set in state s, T is the state transition probability matrix, r is the reward function, such that $r(s, a, s')$ is the immediate reward obtained in s when action $a \in A_s$ is taken and state s' is the next state, Σ is the initial state distribution over the state space S, and $\gamma \in (0,1)$ is the discount factor, which weights immediate rewards versus future rewards. Standard MDP definitions rely on the Markovian (or memory-less) property and on the stationary distribution of the stochastic process. Under these assumptions, the transition probabilities are stationary.

A stationary policy u is a mapping of each state to an action, i.e., $u(s) = a, s \in S, a \in A_s$. The MDP problem is aimed at finding an optimal policy $u^*: S \to A$ that maximizes, in the long run, the expected discounted reward:

$$R(u) \coloneqq E_{u,\Sigma}\left\{\sum_{t=0,1,\dots,\infty} \gamma^t r\big(s(t), a(t), s(t+1)\big)\right\}, \qquad (1)$$

where $s(t)$ and $a(t)$ denote the state and the action at time t, respectively, and $E_{u,\Sigma}\{\cdot\}$ denotes the expected value under policy u with initial state distribution Σ.

The value function $V_u(s)$ is the expected discounted reward starting from s and following policy u thereafter, and the action-value function $Q_u(s, a)$ is the expected discounted reward, starting from s, taking action $a \in A_s$ and following policy u thereafter:

$$V_u(s) \coloneqq E_u\left\{\sum_{t=0,1,\dots,\infty} \gamma^t \cdot r\big(s(t), a(t), s(t+1)\big) \mid s(0) = s\right\}, \qquad (2)$$

$$Q_u(s, a) \coloneqq E_u\left\{\sum_{t=0,1,\dots,\infty} \gamma^t \cdot r\big(s(t), a(t), s(t+1)\big) \mid s(0) = s, a(0) = a\right\}, \qquad (3)$$

where $E_u\{\cdot\}$ denotes the expected value under policy u.

As mentioned in Section 1, the paper interest is in solving the MDP by means of RL algorithms. Let the system be in a given state $s \in S$; RL algorithms take an action $a \in A_s$ based on a the current policy and then, after the transition, observe the next state $s' \in S$ and the obtained reward $r(s, a, s')$. Based on the observations, the RL algorithms update an estimate of the value function of state s or of the action-value function of the pair (s, a).

RL algorithms differ by the rule used to decide the control action and by the rule used to update the value (or action-value) function. In this paper, the Q-learning algorithm is

considered, but more complex RL algorithms are foreseen for the algorithm improvements (see Section VI). The Q-Learning update rule is

$$Q\big(s(t),a(t)\big) \leftarrow \big(1-\alpha(t)\big)Q\big(s(t),a(t)\big) + \alpha(t)\cdot \cdot \left[r\big(s(t),a(t),s(t+1)\big) + \gamma \max_{a' \in A_{s(t+1)}} Q(s(t+1),a')\right]. \quad (4)$$

In equation (4), $\alpha(t) > 0$ is the learning rate and is the key parameter for the algorithm convergence: if $\sum_{t=1,\dots,\infty} \alpha(t) = \infty$ and $\sum_{t=1,\dots,\infty} \big(\alpha(t)\big)^2 < \infty$, the estimate (4) converges to the optimal action-value function as $t \to \infty$ [13]. The action is then decided based on the current estimate of the state-action value function, and, at time t, the current best policy is:

$$u(s(t)) = \underset{a \in A_s}{\operatorname{argmax}}\, Q(s(t),a)\,, s \in S. \quad (5)$$

As the estimate (4) converges to the optimal action-value function, the policy (5) converges to an optimal policy.

To guarantee a certain degree of exploration of the state space set, an ε-greedy rule is followed for the action selection: in state $s \in S$, the current best action (5) is taken by the controller with probability $1-\varepsilon$, where $\varepsilon \in (0,1)$ is the exploration rate; a random action $a \in A(s)$ is chosen with probability ε, i.e.:

$$u(s) \leftarrow \begin{cases} \underset{a \in A(s)}{\operatorname{argmax}}\, Q(\mathrm{s},\mathrm{a})\,, \text{with prob. } 1-\varepsilon \\ \mathrm{rand}\{a \in A(s)\}\,, \text{with prob. } \varepsilon \end{cases}, s \in S. \quad (6)$$

A large value of ε guarantees that different policies with respect to the current best one are explored, and thus avoids that the system remains stuck in a local minimum. A small value of ε, on the other hand, lets the system choose the best action based on the current estimates of the action-value function and favors the exploitation of the current best policy.

The choice of $\alpha(t)$ and ε depends on the specific application.

IV. Proposed RL-Based Traffic Steering Algorithm

A. Problem Model

Let I be the set of UEs connected within a RAN, let K be the set of different services available to each UE, let P be the set of different APs of the RAN and let $P^i \subseteq P$ be the set of APs available to UE $i \in I$.

Each AP is characterized by the amount of available resources in terms of bitrate, denoted with $W_p, p \in P$. Similarly, the services are characterized in terms of required bitrate. Different types of services are considered: elastic services, such as web browsing, for which the user-perceived quality improves with the assigned bitrate, and non-elastic services, such as augmented reality streams, for which a fixed bitrate is needed for the transmission. Among the elastic services it is also possible to discern services in which the service quality varies depending on the encoding that is available at the considered bitrate, as for example multi-codec video streams.

Let w_{pk} be the amount of bitrate allocated on AP p for a service of type k and $r_{pk}(w_{pk})$ be the perceived quality of connection experienced by the users. It is possible to characterize the three types of services as follows.

Elastic traffic: this kind of QoS-flows benefits from having more dedicated resources, therefore its user-perceived quality function grows with the allocated bitrate w_{pk} starting from a minimum value and up to a maximum one:

$$r_{pk}(w_{pk}) = \begin{cases} 0 \text{ if } w_{pk} < w_k^1 \\ f(w_{pk}) \text{ if } w_k^1 \le w_{pk} \le w_k^2 \\ f(w_k^2) \text{ otherwise} \end{cases}. \quad (7)$$

An example of this kind of utility function is represented in Figure 1, where the utility function is proportional to the allocated bitrate, i.e., $r_{pk}(w_{pk}) = cw_{pk}$ up to a maximum bitrate w_k^1.

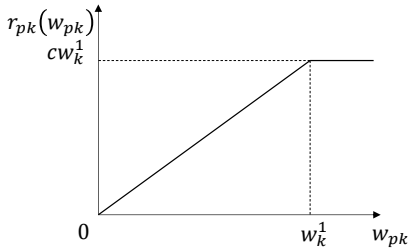

Figure 1. User-perceived quality of connection for service with elastic traffic.

Real-time traffic with guaranteed bitrate (fixed bitrate service): this kind of QoS-flows requires a fixed amount of bitrate, therefore its utility function is positive and constant if enough bitrate is allocated onto a suitable access network, 0 otherwise, as reported in Figure 2.

$$r_{pk}(w_{pk}) = \begin{cases} 0 \text{ if } w_{pk} < w_k^1 \\ r_{pk}^1 > 0 \ \ otherwise \end{cases}, \quad (8)$$

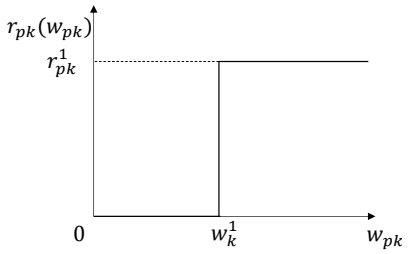

Figure 2. User-perceived quality of connection for service with transmission bitrate threshold.

Multi-codec traffic: this kind of QoS-flows improves its quality depending on its encoding. The available encodings depend on the amount of resources allocated for the service, according to a distribution with multiple thresholds as reported in Figure 3.

$$r_{pk}(w_{pk}) = \begin{cases} 0 & \text{if } w_{pk} < w_k^1 \\ r_{pk}^1 > 0 & \text{if } w_k^1 \le w_{pk} < w_k^2 \\ r_{pk}^2 > r_{pk}^1 & \text{if } w_k^2 \le w_{pk} < w_k^3, \\ r_{pk}^c > r_{pk}^{c-1} & \text{if } w_{pk} \ge w_k^C \end{cases} \quad (9)$$

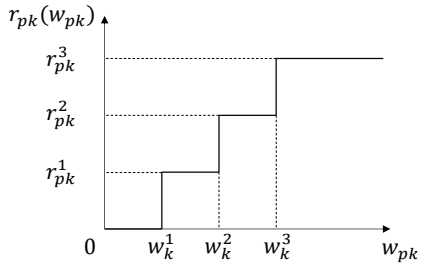

Figure 3. User-perceived quality of connection for service with multi-codec traffic.

The reward associated with each state is represented by the cumulative QoE of the network users, obtained by summing all perceived quality of connection for all ongoing PDU sessions.

At time t, we denote with $n_{pk}^c(t)$ the number of on-going QoS-flows of type k on AP p, considering the level c of the allocated bitrate, and with

$$n_{pk} = \sum_{c=1,\dots,C_k} n_{pk}^c,$$

where C_k is the number of bitrate threshold levels that characterize the reward function r_{pk}.

By defining $\eta_p^1(s(t))$ as the minimum amount of bitrate required to support the on-going QoS-flows at the minimum bitrate level in state s, i.e,

$$\eta_p^1(s(t)) = \sum_{k\in K} n_{pk} w_k^1, p \in C, \quad (10)$$

the state space S is defined as

$$S = \left\{ s = (n_{pk})_{p\in P, k\in K} \middle| \eta_p^1(s) \le W_p \right\} = \{s_1, s_2, \dots, s_{|S|}\}. \quad (11)$$

Since the considered resource (bitrate) is additive, and since the AP resources W_p are finite, the discrete state space S is finite as well. With little abuse of notation, the number of QoS-flows of type k on AP p in state s is denoted with $n_{pk}(s)$.

At each service request, the RAN controller has to decide whether to admit or reject the service request, and also, in case of admission, the AP which has to transmit the service to the UE. Let δ_{pk} be a $|P| \cdot |K|$ vector of all zeros but the element associated to the AP p and the service type k. Then, in each state s, a request of service k can be allocated on AP p only if $s + \delta_{pk} \in S$; otherwise, the request must be rejected. The action set in state $s_j \in S$ is then defined as

$$A_{s_i} = \left\{ a = (a_{pk})_{p\in P, k\in K} \middle| a_{pk} \in \{0,1\} \text{ if } s_i + \delta_{pk} \in S, a_{pk} = 0 \text{ if } s_i + \delta_{pk} \notin S, \forall p \in P, \forall k \in K, \sum_{p\in P} a_{pk} \le 1, \forall k \in K \right\}, i = 1, \dots, |S|, \quad (11)$$

where a_{pk} denotes the action of admitting a request of service k on AP p. The constraints

$$\sum_{p\in P} a_{pk} \le 1, \forall k \in K$$

in (11) states that, for each service k, the request can be either allocated to a single AP p or rejected.

For the sake of the analysis, it is assumed that, for each service type $k \in K$, the service requests arrive according to a Poisson distribution in time with intensity ν_k and their duration is exponentially distributed with mean termination frequency μ_k. Under a given policy u, when the system is in state s, the transition frequencies between states occur according to the arrival and termination frequencies and to the admission decisions:

- if the action $u(s)$ is to allocate the service k on AP p, the transition from state s to state $s + \delta_{pk}$ occurs with frequency ν_k;
- since a mapped service k on AP p terminates with termination frequency μ_k, the transition from state s to state $s - \delta_{ck}$ occurs with frequency $n_{ck}(s)\mu_k$, regardless of the policy u.

For the sake of the analysis of the MDP properties, a

procedure known as *uniformization* can be applied to transform a continuous-time MDP to a discrete-time one, and it can be shown (see, e.g., [14]) that the two MDPs are equivalent. The procedure consists in defining discrete-time transitions, i.e., in defining a transition matrix T, by following a two-step procedure: in the first step, the transition frequencies of each state must be divided by constant value which is larger than the sum of the outgoing transition frequencies of any state $s \in |S|$; in the second step, self-transitions are added to each state in such a way that the outgoing probability is 1. Thanks to the RL approach, the computation of the transition probabilities is not necessary.

Given a state $s(t)$, we are also interested in computing the bitrate which is actually used by the APs for the on-going services and in the fact that the user experience for some services can be improved if the APs assign more bitrate than the minimum required one, in order to maximize the associated reward. Therefore, an allocation procedure must be defined at every admission decision or QoS-flow termination to decide the allocation of the bitrate exceeding the minimum one $\eta_p^1(s(t))$ to the on-going QoS-flows. In state $s(t)$, the bitrate allocation procedure returns $n_{pk}^c(t)$ and r_{pk}^c (the number of QoS-flows of service k on-going on AP p with granted bitrate level c and their associated rewards), for all $p \in P, k \in K$ and $c = 1, \dots, C_k$. It is then possible to associate each state $s(t)$ with an amount of bitrate required at the time *t* by all the allocated QoS-flows on AP *p*:

$$\eta_p(s(t)) = \sum_{k \in K} \sum_{c=1,\dots,C_k} n_{pk}^c(t) w_k^c, p \in P. \quad (12)$$

Consistent with this idea, we define a state-dependent reward $r(s)$ in state s considering the whole capacity allocation and not only the minimum one $\eta_p^1(s)$:

$$r(s) = \sum_{p \in P} \sum_{k \in K} \sum_{c=1,\dots,C_k} n_{pk}^c(t) r_{pk}^c, \quad (13)$$

The described stationary MDP is ergodic and unichain (i.e., under all stationary policies, it is aperiodic and has a single recurrent class and possibly a non-empty set of transient states, see [15] for details), since the transitions are stochastic – and, therefore, aperiodic – and the transitions due to service terminations are always positive and independent of the policy. Thus, the expected state sojourn times under policy u, denoted with y_u's, exist and are finite. In this paper, the interest is in maximizing the expected discounted reward (1)[1], computed as

$$R(u) = (1-\gamma) \sum_{j=1,\dots,N} y_u(s_j) r(s_j), u \in U. \quad (14)$$

B. Proposed Algorithms

We note that the described MDP could be computed by means of DP algorithms or of its linear programming formulation [15], which however would require the knowledge of the transition matrix. Even if the transition matrix were known, the MDP model described in Section IV.A would not be tractable by standard DP methods due to scalability reasons. Therefore, this paper proposes to apply ADP techniques to reduce the problem dimension and RL to obtain a data-driven algorithm.

Concerning ADP, we reduce the state space dimension by aggregating the states with similar minimum capacity allocation. For every AP $p \in P$, by defining a granularity Δ_p and using $\lfloor \rfloor$ as the truncation operator to the next lower integer, the aggregate state set with $\left\lfloor \frac{W_p}{\Delta_p} \right\rfloor$ bitrate levels is obtained as

$$\tilde{S} = \left\{ \tilde{s} = (l_p)_{p \in P} \middle| l_p = \left\lfloor \frac{\sum_{k \in K} n_{pk} w_k^1}{\Delta_p} \right\rfloor, \text{with } (n_{pk})_{k \in K} \text{ s.t.} \sum_{k \in K} n_{pk} w_k^1 < W_p, \forall p \in P \right\}. \quad (15)$$

Clearly, the number of states decreases as the granularities Δ_p's grow.

Due to the state space aggregation, also the action space (which we recall is dependent on the state) needs to be changed. Since different bitrate levels can be associated to a single aggregate state $\tilde{s}$, it might happen that, at two time instants t' and t'', for a given service c, the system is in state $\tilde{s}$ with minimum load $\eta_p(\tilde{s}(t'))w_k^1 < W_p - w_{pk}$ and $\eta_p(s(t''))w_k^1 > W_p - w_{pk}$, respectively: only at time t'', the system could accept a new service k request. Since the action set is associated to a state, standard approaches would be either to consider the admission action for service c as not available in state $\tilde{s}$, regardless of the availability of the necessary capacity, or to "disaggregate" these states.

To account for these occurrences without increasing the dimension of the state space, we define a state-dependent action space which also depends on the actual measured AP transmission bitrate:

[1] With some awareness, the proposed method is applicable also to the undiscounted and finite-horizon cases.

$$\tilde{A}(\tilde{s}(t)) = \left\{ (\tilde{a}_{pk})_{p\in P, k\in K} \middle| \tilde{a}_{pk} \in \{0,1\} \text{ if } \eta_p(\tilde{s}(t)) w_k^1 < W_p - w_{pk}, \tilde{a}_{pk} = 0 \text{ otherwise}, \forall k \in K, \sum_{p\in P} \tilde{a}_{pk} \le 1, \forall k \in K \right\}. \quad (16)$$

Correspondingly, at time t, the observed reward in state $\tilde{s}(t)$ is computed as

$$r(t) = \sum_{p\in P} \sum_{k\in K} \sum_{c=1,\dots,C_k} n_{pk}^c(t) r_{pk}^c, \quad (17)$$

The action set approximations introduced so far require modifications of the standard Q-learning algorithm, since an action might not be available at every visit of state $\tilde{s}$. In particular, the update rule (4) of the Q-table needs to be modified. At time t, for each state $\tilde{s}$ and service k, let $n_{\tilde{s}}(t)$ be the number of visits of state $\tilde{s}$ and $n_{\tilde{s},\tilde{a}}(t)$ be the number of visits of state $\tilde{s}$ when the action $\tilde{a}_{pk}$ was available. Let $\tilde{s}(t) = \tilde{s}'$ and $\tilde{a}(t) = \tilde{a}'$; the quantity $N_{\tilde{s}',\tilde{a}'}(t) = \frac{n_{\tilde{s}'}(t)}{n_{\tilde{s}',\tilde{a}'}(t)}$ is then used in the update rule:

$$Q(\tilde{s}', \tilde{a}') = \big(1 - \alpha(t)\big) Q(\tilde{s}', \tilde{a}') + \alpha(t) \Bigg(r(t) + + \left(\sum_{n=1,\dots,\lfloor N_{\tilde{s}',\tilde{a}'} \rfloor} \gamma^n + \gamma^{\left(N_{\tilde{s}',\tilde{a}'} - \lfloor N_{\tilde{s}',\tilde{a}'} \rfloor\right)} \right) \Bigg) \max_{\tilde{a}\in\tilde{A}(\tilde{s}(t+1))} Q(\tilde{s}(t+1), \tilde{a}). \quad (18)$$

In this way, the actions which are less often available in a state $\tilde{s}'$ are not penalized. An example is presented to clarify the proposed update rule. Let $\tilde{a}'$ an action that is always available in state $\tilde{s}'$, i.e., $N_{\tilde{s}',\tilde{a}'} = 1$. When $\tilde{s}(t) = \tilde{s}'$ and $\tilde{a}(t) = \tilde{a}'$, the standard Q-learning update rule is enforced and $Q(\tilde{s}', \tilde{a}')$ is updated using the best next-value of Q discounted by γ:

$$Q(\tilde{s}', \tilde{a}') \leftarrow \big(1 - \alpha(t)\big) Q(\tilde{s}', \tilde{a}') + + \alpha(t) \left[r(t) + \gamma \max_{\tilde{a}\in\tilde{A}(\tilde{s}(t+1))} Q(\tilde{s}(t+1), \tilde{a}) \right].$$

Conversely, let another action $\tilde{a}''$ be available, on the average, about half of the times the state $\tilde{s}'$ is visited, with $N_{\tilde{s}',\tilde{a}''} = 2.3$. The rule (18) states that, if $\tilde{s}(t) = \tilde{s}'$ and $\tilde{a}(t) = \tilde{a}''$, $Q(\tilde{s}', \tilde{a}'')$ is updated with the best next-value of Q discounted by a larger γ, as if the pair $(\tilde{s}', \tilde{a}'')$ was visited 2.3 times:

$$Q(\tilde{s}', \tilde{a}'') \leftarrow \big(1 - \alpha(t)\big) Q(\tilde{s}', \tilde{a}'') + + \alpha(t) \left[r(t) + (\gamma + \gamma^2 + \gamma^{0.3}) \max_{\tilde{a}\in\tilde{A}(\tilde{s}(t+1))} Q(\tilde{s}(t+1), \tilde{a}) \right]$$

The proposed rule avoids that action $\tilde{a}''$ is not chosen even if available because of infrequent visits.

V. Simulations

We assume that the three considered services are prioritized, with the CBR service having the maximum priority and the elastic service having the minimum priority, and that, for each service, the service sessions have the same priority. Therefore, for a given AP p, if $\eta_p^1(s) < W_p$, the remaining bitrate is allocated in a fair way to the multi-coded services; if all the multi-coded service instances receive their maximum bitrate, the remaining resource is equally shared among the elastic service instances. The pseudo-code in Table 1 details the bitrate implemented allocation algorithm, which allocates the available resources to the services with strict priorities (e.g., all service connections $k+1$ are served before service connections k, $k = 1, \dots, K-1$) and in a fair way among the connections of the same service.

Table 1. Bitrate allocation algorithm for AP $p \in P$ in state $s \in S$.

- Assign the minimum required bitrate to all the on-going connections of AP p:

$$n_{pk}^c(s) = \begin{cases} n_{pk} \text{ if } c = 1 \\ 0 \text{ otherwise} \end{cases}, k = 1, \dots, K$$

- For $k = 1, \dots, K$
  - Set $c = 2$
  - While $n_{pk}^c = n_{pk}$ and $c \le C_k$
    - Compute the bandwidth already assigned to all the connections of each service

$$\eta_p(s) = \sum_{k\in K} \sum_{c=1,\dots,C_k} n_{pk}^c w_k^c$$

    - Share the leftover bandwidth $W_p - \eta_p(s(t))$ in a round-robin fashion among the n_{pk}^{c-1} multi-codec PDU sessions on-going on AP p (after the round, all the connections codec are upgraded only if $n_{pk}^c = n_{pk}$)
    - Evaluate r_{pk}^c
    - Set $c = c + 1$
  - End
- End

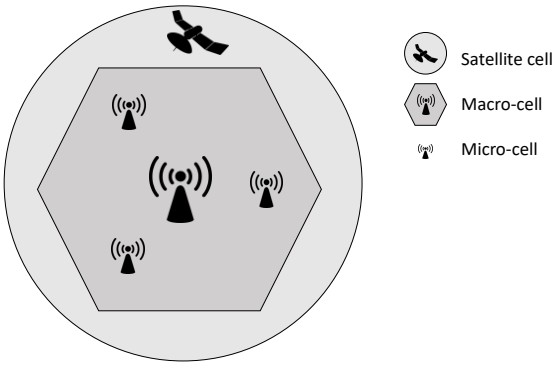

Figure 4. Reference scenario used in the simulations.

The simulations ran over a scenario of 1 hour, during which the various service types arrive according to Poisson distributions of mean values $2s$, $6s$ and $4s$ for service 1, 2 and 3, respectively. Service dwelling time was determined, for each service type, according to exponential distributions of mean values $30s$, $120s$ and $90s$.

The Considered RAN is reported in Figure 4 and is characterized by the presence of three Micro-cells, one Macro-cell and the availability of satellite coverage in the whole area. The UE of the connection requests was uniformly distributed in the area covered by the Macro-cell; depending on the position of the UE, it can connect to either one of the Micro-cell, or to a pair of Micro-cells or to all three of them. It was assumed that Micro-cells offer 2GBps, whereas the Macro-cell and the satellite cell are limited to 1GBps.

The reward associated to the elastic services is set as

$$r_{p1}(w_{p1}) = \begin{cases} 0 \text{ if } w_{p1} < 0.01\ GBps \\ 200 w_{p1} \text{ if } 0.01\ GBps \leq w_{p1} < 0.01\ GBps \\ 20 \text{ if } w_{p1} \geq 0.1\ GBps \text{ otherwise} \end{cases}.$$

Multi-codec services are characterized by a reward of

$$r_{p2}(w_{p2}) = \begin{cases} 0 \text{ if } w_{p2} < 0.1\ GBps \\ 6 \text{ if } 0.1\ GBps \leq w_{p2} < 0.12\ GBps \\ 12 \text{ if } 0.12\ GBps \leq w_{p2} < 0.18\ GBps \\ 25 \text{ if } w_{p2} \geq 0.18\ GBps \end{cases}.$$

Finally, fixed bitrate services have the following reward:

$$r_{p3}(w_{p3}) = \begin{cases} 0 \text{ if } w_{p3} < 0.2\ GBps \\ 20 \text{ if } w_{p3} \geq 0.2\ GBps \end{cases}.$$

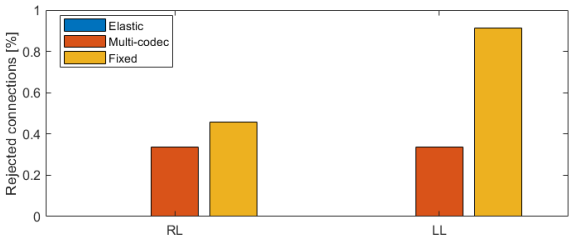

Figure 5. Percentage of blocked connections for the RL and the LL controllers.

Additionally, the rewards were scaled depending on the AP on which their corresponding service was allocated, to capture also the trade-off between user satisfaction and operative cost incurred by the operator, which changes depending on RATs and the specific APs. The scaling factors were set to $0.75, 1, 1.25$ for the micro cells, 2 for the Macro-cell and 0.5 for the satellite.

The design parameters of the Q-Learning controller were set as $\gamma = 0.9$, $\varepsilon = 0.05$ and $\alpha(t) = 1/(1 + \lfloor \tau(t)/100 \rfloor)$ where $\tau(t)$ represents the number of connection requests processed by the system at time t.

The simulation results in terms of total cumulative reward, averaged over 100 simulation runs, showed that the proposed RL controller attains an increase in performance of around 5.3% compared to a baseline Least-Loaded (LL) balancer controller, which allocate each new connection on the currently least-loaded AP.

Figure 5 shows how the strategies of the two controllers (RL and LL) impact on the connection blocking rates. It can be noted that, not being forced to accept any incoming call on the least loaded APs, the RL controller manages to attain a higher reward by reserving the most profitable (in terms of reward scaling factor) network resources for the services associated with the highest rewards. From the figure, it can be observed that, using the RL controller, the rejection rate for services of type 3 halves, meaning that overall the network resources are better managed.

VI. Conclusions and Future Works

This paper presents a Reinforcement Learning controller for the problem of traffic steering and network selection in the 5G framework of Heterogeneous Networks. The problem is modelled as a Markov Decision Process with a novel state-space aggregation approach, and a load-balancing algorithm to allocate the network resources is designed, taking into account three classes of services typical of telecommunication networks (fixed bitrate, multi-codec and elastic traffic).

Future research directions could cover the introduction of the Deep Reinforcement Learning framework to the problem

to deal with more realistic scenarios (e.g., with moving users, on larger scales) and the algorithm validation on the 5G-ALLSTAR testbed.

ACKNOWLEDGEMENTS

The authors acknowledge the CRAT team working in the project 5G-ALLSTAR and the whole consortium.

REFERENCES

[1] M. Dryjanski and M. Szydelko, "A unified traffic steering framework for LTE radio access network coordination," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 84–92, Jul. 2016.

[2] 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.

[3] L. Wang and G.-S. 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.

[4] 3GPP TS 23.501, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2 (Release 15)," 2018.

[5] L. Hui, W. Ma, and S. Zhai, "A novel approach for radio resource management in multi-dimensional heterogeneous 5G networks," *J. Commun. Inf. Networks*, vol. 1, no. 2, pp. 77–83, Aug. 2016.

[6] N. Zhang, S. Zhang, S. Wu, J. Ren, J. W. Mark, and X. Shen, "Beyond Coexistence: Traffic Steering in LTE Networks with Unlicensed Bands," *IEEE Wirel. Commun.*, vol. 23, no. 6, pp. 40–46, Dec. 2016.

[7] A. Wilson, A. Lenaghan, and R. Malyan, "Optimising Wireless Access Network Selection to Maintain {QoS} in Heterogeneous Wireless Environments," in *International Symposium on Wireless Personal Multimedia Communications 2005 (WPMC 2005)*, 2005, pp. 1236–1240.

[8] M. Cesana, N. Gatti, and I. Malanchini, "Game Theoretic Analysis of Wireless Access Network Selection: Models, Inefficiency Bounds, and Algorithms," in *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, 2008, p. 6.

[9] J. Antoniou and A. Pitsillides, "4G Converged Environment: Modeling Network Selection as a Game," in *2007 16th IST Mobile and Wireless Communications Summit*, 2007, pp. 1–5.

[10] X. Gelabert, J. Perez-Romero, O. Sallent, and R. Agusti, "A Markovian Approach to Radio Access Technology Selection in Heterogeneous Multiaccess/Multiservice Wireless Networks," *IEEE Trans. Mob. Comput.*, vol. 7, no. 10, pp. 1257–1270, Oct. 2008.

[11] N. Vučević, J. Pérez-Romero, O. Sallent, and R. Agustí, "Reinforcement learning for joint radio resource management in LTE-UMTS scenarios," *Comput. Networks*, vol. 55, no. 7, pp. 1487–1497, May 2011.

[12] S. Kalyanasundaram, E. K. P. Chong, and N. B. Shroff, "Optimal resource allocation in multi-class networks with user-specified utility functions," *Comput. Networks*, vol. 38, no. 5, pp. 613–630, Apr. 2002.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA, 1998.

[14] D. P. Bertsekas, *Dynamic Programming deterministic and stochastic models*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.

[15] M. L. Puterman, *Markov Decision Processes*. New York: John Wiley & Sons, Inc., 1994.