Satellite Integration into 5G: Deep Reinforcement Learning for Network selection

Emanuele De Santis[1] Alessandro Giuseppi[1] Antonio Pietrabissa[1]
Michael Capponi[1] Francesco Delli Priscoli[1]

[1]Department of Computer, Control and Management Engineering "Antonio Ruberti", University of Rome La Sapienza, Rome

Abstract: This paper proposes a Deep-Q-Network (DQN) controller for network selection and adaptive resource allocation in heterogeneous networks, developed on the ground of a Markov Decision Process (MDP) model of the problem. Network selection is an enabling technology for multi-connectivity, one of the core functionalities of 5G, and for this reason the present work considers a realistic network model that takes into account path-loss models and intra-RAT (Radio Access Technology) interference.

Numerical simulations validate the proposed approach and show the improvements achieved thanks to the DQN algorithm with respect to a classic Reinforcement Learning algorithm and baseline approaches in terms of connection-flows' acceptance, resource allocation and load balancing.

Keywords: Network Selection, HetNet, Deep Reinforcement Learning, Deep-Q Networks, 5G Communications.

1. Introduction

The exponential increase in bandwidth, coverage and data rate demands, along with the diversification of use cases that are planning to use cellular Radio Access Networks (RANs) to provide connectivity, have prompted the development of the fifth generation (5G) Radio Access Technology (RAT). Through the support for higher mobile bandwidths complemented with low latency and more reliable communications, the 5G RAT is expected to address the significant increase in data rate demands that network operators are expecting and to support the diversification of services required by User Equipment (UE) during the coming years. Moreover, 5G specifications, starting from release 16 [1] will include other RATs in the 5G environment, such as 4G LTE and Satellite Access Points (APs). In this system where the connections demand continues to increase, an appropriate network resources management is required since an optimal allocation of those resources will guarantee better performances and will help to ensure user requirements in terms of Quality of Experience (QoE) without overloading the network.

In this paper a Network Selection technique relying on Markov Decision Procesess (MDPs) and on Deep-Q-Network (DQN) [2] algorithm has been studied. A centralized controller will take care of allocating in the best way requests coming from UE analyzing the network state in terms of APs load and UE perceived transmission power. The goal of this study is to show the effectiveness of the proposed Deep Reinforcement Learning approach by simulations with a realistic multi-RAT (5G/4G/Sat.) network scenario. Moreover, several classes of user requests have been modeled, in order to represent different connection service requirements in terms of downlink bitrate, Quality of Service (QoS) requirements and QoE profiles.

The remainder of the paper is organized as following: section 2 provides an overview of the state of the art and the main contributions of the paper; in section 3 a sketch of the control algorithm is presented, while in section 4 some preliminaries on MDPs and DQN are introduced; in section 5 the problem modelling is discussed and section 6 reports the simulation results and the validation of the proposed algorithm. Finally, section 7 draws the conclusions and highlights future works.

2. State of the art, innovations and limitations of the proposed approach

Network selection plays a fundamental role in the provision of stable connections with an adequate level of QoS and hence network operators and providers commonly exploit several advanced techniques to select the best AP to allocate new connections. Among the various techniques proposed in the literature, Multiple Attribute Decision Making (MADM), proved to be one of the most flexible solutions to capture user preferences and QoE related aspects in the decision process [3]–[7]. In MADM solutions, the information characterizing the decision making is made by the so-called *attribute values* and *attribute weights*: the first ones describing characteristics, qualities and performances of different alternatives, whereas the latter ones are used to measure the relevance of attributes.

Modelling the network selection problem as a MADM, it is then possible to decide the trade-off among service QoS requirements, user preferences and overall network congestion.

A similar approach is followed in the present work, in which a different QoE profile is associate to the various connections, depending on its specific service characteristics

Among the other solutions, we mention approaches based on fuzzy logic [8]–[11], a methodology that allow fast decision making but heavily rely on operator's knowledge

Research Article

Manuscript received date; accepted date; published online date

Recommended by Associate Editor xxxx

and best practices, and Game-Theory [12]–[16]

In Game theory-based approaches, the problem is modelled as a set of players/agents coupled with a set of network states and possible agent actions, commonly utilising the Markov Decision Process (MDP) framework [17]. The main idea behind this method is that player's actions are influenced by the choices and actions of the other players. The interaction among the players can either be adversarial, i.e., each agent tries to maximise its own performance, or cooperative, when agents share a common objective.

The approaches mentioned so far are typically employed in scenarios in which the controller is provided with a model of the network and user behavior, such as a statistical distribution of the incoming connection requests and QoE profiles, like in [18], [19] where the authors studied how to maximise QoE/QoS for specific services (e.g., Video Streaming applications. On the contrary, this work employs Reinforcement Learning (RL) [17] a model-free control methodology that allows the network controller to automatically acquire the knowledge on the system by interacting with it and *experiencing* its response to different control policies.

RL has been extensively applied in the network control domain [20]–[25] and has become particularly appealing over the last few years thanks to the innovations bought by its deep learning based variant, namely *Deep Reinforcement Learning* (DeepRL) [2], that allowed RL-based controllers to address problems previously challenging due to their complexity and high dimensionality [26].

The main contributions of this work are:

- The design of a two-step network control algorithm based on Deep Reinforcement Learning for the problem of network selection and optimal resource management in the heterogeneous 5G networks setting, also envisaging the presence of satellite communication systems.
- The inclusion in such control framework of QoE maximisation by considering three different service types with different QoS-QoE relations.
- The development of an open-source network simulator [27] able to model several different radio access technologies, including satellite systems, in terms of network resource usage.

3. Sketch of the Control Algorithm

The algorithm designed in this work is a 2-step process: first, the controller that governs the RAN receives a connection request and determines on which available AP it should be allocated. The AP reserves for the allocation the network resources needed to satisfy the connection minimum QoS requirements to guarantee service provision; then, the distributed controllers that oversee the various APs distribute the remaining network resources to the connections they sustain to improve the QoE of their users. Figure 1 reports a functional diagram of the proposed control scheme, highlighting the flow-chart of the algorithm and the related data flow.

The first part of the control algorithm proposed will be based on a Deep Reinforcement Learning agent, whereas the network resource allocation will distribute the available resources over the various connections according to their priority.

The next section provides the reader with the needed background on MDP and DeepRL.

4. Markov Decision Process, Q-Learning and Deep Q-Networks

A MDP is defined as the tuple $\{S, A, T, R, \Sigma, \gamma\}$, where S and A are the (continuous or discrete) finite state and action sets, respectively, T is the transition probability function $T : S \times A \times S \rightarrow [0, 1]$, with $T(s, a, s')$ denoting the probability that the next state is s' when the current state is s and the chosen action is a and with $\sum_{s' \in S} T(s, a, s') = 1$, R is the one-step reward function $R: S \times A \times S \rightarrow \mathbb{R}_+$, Σ is the initial state distribution and $\gamma \in (0, 1)$ is the *discount factor* that weights future rewards against immediate ones. The set of actions might be state-dependent as not all the actions might be available at each state; the set of actions available at a given state $s \in S$ will be denoted by $A(s) \subseteq A$.

MDPs rely on the Markov Property (also known as the memory-less property), according to which the future evolution of a system given an action and state pair does not depend on the previous actions and states that the system incurred into.

A deterministic policy $\pi : S \rightarrow A$ selects one action for each state. Let Π be the set of feasible policies π such that $\pi(s) \in A(s)$ for all $s \in S$. The expected discounted reward obtained by starting from state s and following policy π thereafter is represented by the state-value function, defined as

$$V_\pi(s) = \mathbb{E}_\pi\left(\sum_t \gamma^t R(s_t, a_t, s_{t+1}) \;|\, s_0 = s\right) \qquad (1)$$

where $\mathbb{E}_\pi$ is the expected value under policy π and s_t and a_t represent the state and action at time t. Similarly, the state-action-value function

$$Q_\pi(s, a) = \mathbb{E}_\pi\left(\sum_t \gamma^t R(s_t, a_t, s_{t+1}) \;|\, s_0 = s, a_0 = a\right) \qquad (2)$$

represents the expected discounted reward obtained by following policy π when starting from state s and taking action $a \in A(s)$.

Solving the MDP means to find the optimal policy π^* that maximises the expected cumulative discounted reward, i.e., $\pi^* = argmax_{\pi \in \Pi} V_\pi(s)$. Dynamic programming [17] approaches can be used to exactly determine π^*, but they typically require the complete knowledge of the MDP dynamics – in particular of T and R – and their computing time exponentially increases with the dimensions of state

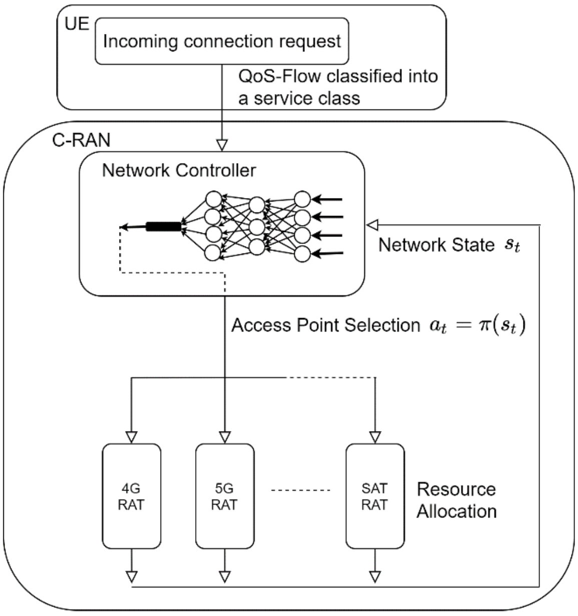
UE
Incoming connection request
QoS-Flow classified into a service class
C-RAN
Network Controller
Network State s_t
Access Point Selection $a_t = \pi(s_t)$
4G RAT
5G RAT
SAT RAT
Resource Allocation

Figure 1: Flow-chart of the control algorithm

and action sets.

Conversely, RL algorithms, such as Q-Learning, aim at obtaining an estimate of the optimal state-action-value function Q_{π^*} based on the experience the controller gathers by interacting with the environment, in the form of the (state, action) pairs it observes and the rewards it collects. RL algorithms in general assume no knowledge of the environment dynamics, and start interacting with it with mostly random policies in a process known as *exploration*. As the RL agent obtains a better knowledge of the environment, it starts *exploiting* its knowledge to determine what it considers to be the better actions, until the estimated Q_{π} converges to Q_{π^*}, from which it is possible to retrieve the optimal policy as $\pi^* = argmax_{a'} Q_{\pi^*}(s, a')$.

The standard update rule for Q-Learning is

$$\begin{aligned} Q(s_t, a_t) &= (1 - \alpha_t) Q(s_t, a_t) + \\ &+ \alpha_t (r_t + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a)) \end{aligned} \tag{3}$$

where $r_t = R(s_t, a_t, s_{t+1})$ is the *measured* reward obtained at time t and $\alpha_t > 0$ is the *learning rate*, which, in order to assure convergence, is subject to the conditions $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.

The balancing between exploration and exploitation is controlled by the parameter $\varepsilon_t \in [0, 1]$ in the so-called ε-greedy policies: at any time t, the agent chooses a random action with probability ε_t, whereas it chooses the action that maximizes the state-action-value function (i.e., $argmax_{a \in A(s)} Q(s, a)$) with probability $(1 - \varepsilon_t)$.

It is worth noting that in standard RL approaches the Q function is updated only for the visited state-action pairs, so, in order to have a complete estimation of the optimal Q function it is needed to visit at least once every state-action pair. This implies that the state space S and the action space A must be finite and discrete, and if their dimensions increase also RL algorithms incur in the so-called *curse of dimensionality*.

To address these issues, the Deep Q-Network (DQN) algorithm was proposed in [2] as a Deep Learning solution for function approximation-based [17] Q-Learning. DQN approximates the Q function by the means of a Deep Neural Network able to approximate high-dimensional functions with a low-dimensional representation. The training process for the Neural Network is detailed in [2] and, despite having included some technical solutions to address the Neural Network limitations, such as target network and memory buffers, conceptually it remains the same as in the standard Q-Learning, with equation (3) replaced by the training process of the Neural Network and in particular by the updates of its weights.

The main advantage of using DQN is its ability to cope with continuous state spaces and it proved capable of solving complex problems, such as playing videogames. Note that DQN still considers discrete action sets; actor-critic solutions such as Deterministic Deep Policy Gradient (DDPG) should be used when dealing with continuous actions.

5. Problem modelling

This section presents the modelling of the network selection problem as an MDP. In particular, the subsections from 5.1 to 5.3 formulate the sets and functions required for the MDP formalism, while the subsection 5.4 and 5.5 detail the physical processes that allow the conversion of network resources into bitrate provision.

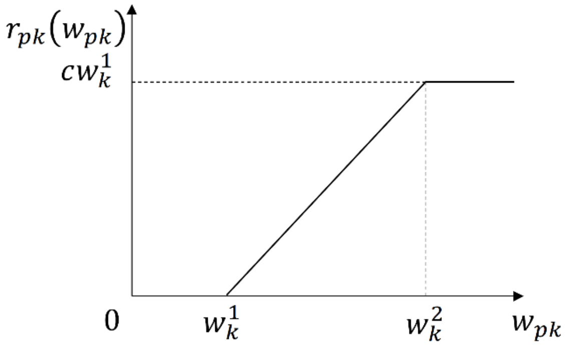

Figure 2: QoE profile of elastic services (k=1)

Let I be the set of User Equipments (UEs) connected within a RAN constituted by a set P of Access Points (APs). Each UE $i \in I$ is connected to an AP $p \in P$ of the RAN, that is characterized by a certain amount W_p of physical resource blocks (PRBs) available. In addition, let $P^i \subseteq P$ be the set of APs available at UE i, depending on its position and antennas. Moreover, let K be the set of different service types considered, each one characterized by a different minimum bitrate $B_k, k \in K$. Finally, let n_{pk} be the number of requests of type k allocated to an AP p.

Three different types of services are here considered, as in [28], namely: elastic services, non-elastic services and multi-codec ones, each characterised by a different QoE profile.

Let b_{pk}^{i}be the bitrate allocated on AP p for the service k requested by the UE i. We can model the three QoE profiles as the functions $r_{pk}^{i}(b_{pk}^{i})$ depicted in Figure 2 - Figure 4. In particular:

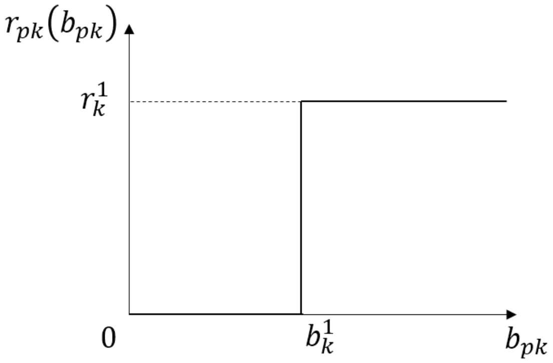

Figure 3: QoE profile of non-elastic services (k=2)

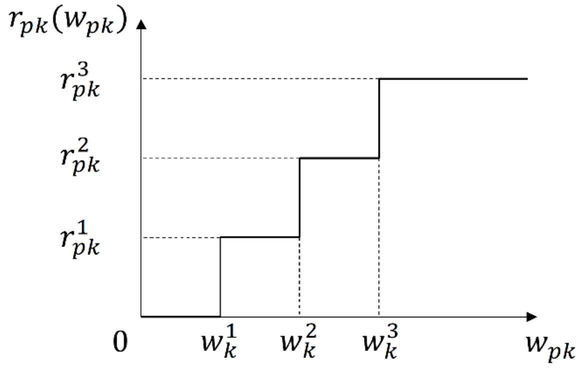

Figure 4: QoE profile of multi-coded services (k=3)

- Elastic services have a linear QoE behaviour with respect to the allocated bitrate, starting from a minimum level b_k^1 up to a maximum bitrate b_k^2 where the perceived quality saturated, as depicted in Figure 2. This service captures applications such as of web-surfing and file downloading.
- Non-elastic services have a threshold-like behavior with respect to the allocated bitrate, so if the bitrate is less than b_k^1 the perceived quality is 0, otherwise is maximal, as depicted in Figure 3. This service type represents well real-time applications with guaranteed bitrate requirements.
- Multi-codec services have a stair-like QoE profile, as the perceived quality has different thresholds corresponding to the utilised codec, that depends on the amount of bitrate allocated b_k^1, b_k^2, b_k^3 , as reported in Figure 4. This service type represents multi-codec video and audio streaming.

The proposed modelling of the services is compliant with the 5G standards, as the so-called QoS-flows that constitute the various connections can be associated with one of the three service types introduced above depending on their QoS requirements and characteristics.

5.1. State space definition

As already introduced, each AP is characterized by the amount of its physical resources available for allocation, denoted as $W_p,\ p \in P$.

To allow the controller to take an optimal decision on the allocation of a new incoming connection request from a given UE, the stat of the network should contain information regarding: (i) the congestion level of the physical resources over the various APs; (ii) the coverage quality that the APs provide to the UE; (iii) the service class, to infer its associated QoE profile, and its bitrate requirements.

In this sense, the minimum quantity of physical resources that need to be allocated to sustain a single QoS-flow i of type k on a given access point p is denoted as w_{pk}^{i}, with $i \in I_{pk}$, where I_{pk} is defined as the set of QoS-flows of type k related to AP p. Note that, referring to Figure 2 to Figure 4, this quantity represents the amount of resources needed to provide the UE with a connection with an associated bitrate b_k^1.

Let $\eta_p^1(t)$ denote the amount of resources allocated at time t to sustain the allocated services (i.e., the amount of physical resources required to support the on-going QoS-flows at their minimum bitrate level). By definition:

$$\eta_p^1(t) = \sum_{k \in K} \sum_{i \in I_{pk}} w_{pk}^{i}(t), \qquad p \in P \tag{4}$$

Let $l_p(t)$ be the load level of an AP p, defined as the allocated physical resources over the total available ones:

$$l_p(t) = \frac{\eta_p^1(t)}{W_p}, \qquad p \in P \tag{5}$$

Given a UE $i \in I$ requesting a service of type $k \in K$, the state space is then given by the following three quantities:

- the load level related to each AP $p \in P$;
- the Reference Signals Received Power (RSRP) value $\mathcal{P}_{i,p}$ for each AP p, measured by the UE itself;
- the minimum amount of bitrate required for the requested service class B_k (b_k^1 in the figures).

The state set can then be defined as:

$$S = \left\{ s = \left\{ (l_p)_{p \in P}, (\mathcal{P}_{i,p})_{i \in I, p \in P}, (B_k)_{k \in K} \right\} \right\} \tag{6}$$

The resulting state $s \in S$ is a vector with $2|P| + 1$ elements. With little abuse of notation, we will denote by $l_p(s), \mathcal{P}_{i,p}(s)$ and $B_k(s)$ as the load level of AP p, the RSRP value and the minimum required amount of bitrate in state s, respectively.

5.2. Action space definition

When a new connection request arrives to the network controller, there are two possible outcomes: (i) the controller accepts the request and allocate it to (exactly) one AP p; (ii) the connection is rejected as there are no APs that can handle it due to insufficient resources. The RAN controller is then required to act as an advanced Connection and Admission Controller (CAC).

We now define the action set similarly to [28]. Let δ_p be a vector with $2|P|+1$ values, i.e., the same dimension of the state vector $s \in S$, where all the values are zeros but the element associated to the AP p. The single non-zeros element is δ_p represents the extra load (5) that would be added to access point p in the case the new connection request is accepted. It follows that, in each state s, a request service may be allocated on AP p if and only if $s + \delta_p \in S$, i.e., by allocating the new request to the AP p the new generated state still belongs to S.

The action set available in a state $s \in S$ is then defined as:

$$A(s) = \left\{(\zeta_1, \zeta_2, \dots, \zeta_{|P|}) \mid \sum_{j \in 1,\dots,|P|} \zeta_j = 1, \zeta_j \in \{0,1\}\, \forall j\right\} \cup \mathbf{0} \tag{7}$$

where $\mathbf{0}$ is a P-vector of zeroes, and the action is a vector whose only non-zero element is equal to one and indicates which AP has been selected for the allocation. The special case in which $a_i = \mathbf{0}$ represents a condition in which the connection request must be rejected due to a lack of network resources, as no AP can allocate the incoming request assuring its minimum required bitrate.

In the simulation in section 6, we will assume that the requests of the service type $k \in K$ for each UE arrive according to a Poisson distribution in time with mean value v_k and that their termination rates follow an exponential distribution with mean termination frequency μ_k.

5.3. Reward function definition

In the presented definition of the states and actions, it was assumed that the network controller only allocates the network resources needed to satisfy the minimum amount of bitrate required by the various connections. As introduced in Section 3, the network control algorithm follows a two-step procedure: firstly, it selects which AP will serve the incoming connection request; then, each AP distributes its remaining resources $\eta_p^1(s)$ over its connections, according to some prioritization order that may take into account the user tariff or operator preferences.

In our simulations, the APs will firstly distribute uniformly their available resources to the multi-codec services, so that each connection receives a bitrate up to b_3^3, and afterwards the remaining resources are uniformly distributed to the elastic services up to a bitrate of b_1^1. Non-elastic services, due to their threshold-like behaviour, are always given a bitrate of b_2^1.

To define the reward function, we have to introduce S_{pi} as the amount of additional bitrate that the AP p is able to provide to the connection i using a share of its remaining resources. This quantity is then directly linked to the QoE associated to the connection, as the function r_{pk} of Figure 2 to Figure 4 takes in general as an argument the quantity $b_k^1 + \mathcal{S}_{pi}$ that represents the total bitrate available to the service i of class k.

The reward function shall then capture three cases:

- the connection request is rejected (i.e., no AP allocates the connection);
- the connection is allocated on an AP with a low resource usage;
- the connection is allocated on an AP that is already providing several other connections.

To capture those three cases, the reward $r_t(s_t, a_t, s_{t+1})$ obtained by the controller when allocating a connection i of class k of AP p can be defined as:

$$r_t(s_t, a_t, s_{t+1}) = \\ = \begin{cases} -r^0 < 0, & if\ a_t = \mathbf{0} \\ r_{pk}\left(b_k^1 + \mathcal{S}_{pi}\right), & if\ l_p(t+1) \le 0.5 \\ r_{pk}\left(b_k^1 + \mathcal{S}_{pi}\right) - r^{sat}, & if\ l_p(t+1) > 0.5 \end{cases} \tag{8}$$

The negative reward $-r^0$ represents a penalty given to the agent if the allocation is rejected to capture the cost incurred by the network operator in failing to provide a connection. The term $r_{pk}\left(b_k^1 + \mathcal{S}_{pi}\right)$ is a positive reward, shaped depending on k as in Figure 2 to Figure 4, that captures the QoE of the new user and the term $-r^{sat}$ is a negative reward subtracted from $r_{pk}\left(b_k^1 + \mathcal{S}_{pi}\right)$ in case the new allocation is destined to an AP whose saturation level is higher than the desired threshold (50% in our case).

The long-term maximization of this reward allows the network controller to maximise the overall QoE of its users while keeping the connection rejection rate minimised.

5.4. 5G NR and 4G LTE resource allocation description

To relate the physical resources that appear in the state definition with the transmission bitrate needed by the reward function to estimate the QoE level, it is now needed to detail their relation and how one translates into the other for both terrestrial and satellite APs.

5G New Radio (NR) APs have a limited set of resources [29], both in terms of frequency bandwidth and time to allocate UE requests. The minimum allocation unit for a 5G NR AP is the Physical Resource Block (PRB), each composed by 12 frequency subcarriers with a $2^{\mu} \cdot 15$kHz bandwidth and a time duration of $2^{-\mu} \cdot 1$ms, where $\mu \in \{0,1,2,3,4\}$ is the parameter called *numerology* defined by 5G NR standards. The number of PRBs available on AP p depends on the available total bandwidth on the AP and on its numerology, as defined by 5G NR standards [29].

For 4G LTE APs the definition of PRB still stands, but the numerology parameter is constrained to $\mu = 0$, so there is no flexibility on using less/more subcarrier bandwidths and more/less time slot durations.

The receiving power, or RSRP, $\mathcal{P}_{i,p}$ that appears in the states of equation (6), represents the transmission power measured by the UE $i \in I$ between itself and the AP $p \in P$ is computed as follows:

$$\mathcal{P}_{i,p} = \mathcal{P}_p \cdot G_p \cdot L_p \cdot L_{i,p} \tag{9}$$

where $\mathcal{P}_p$ is the AP's antenna power, G_p is the AP's antenna gain, L_p is the AP's feeder losses and $L_{i,p}$ is the path loss between UE i and AP p.

In our simulations, the path loss $L_{i,p}$ is computed through the COST-HATA model [30] that is a statistical model that considers many factors as the buildings density (rural, suburban, urban), the carrier frequency used for the communications and the relative heights of UE and AP.

In order to estimate the number of resource blocks to be allocated by the AP $p \in P$ for the communication with the UE $i \in I$, the *signal-over-interference-plus-noise-ratio* (SINR) has to be computed. The thermal noise part can be computed according to:

$$\mathcal{N}_p = k_b T^{env} B_p \Theta_p \tag{10}$$

$$\Theta_p(t) = \frac{\sum_{\tau \in (t-T,t)} \sum_{j \in I \setminus i} C_{j,p}(\tau) N_{j,p}(\tau)}{T \cdot \#R_p} \tag{11}$$

where $\Theta_p(t)$ is the Resource Blocks Utilization Ratio (RBUR) of AP p at time t, k_b is the Boltzmann constant, T^{env} is the environmental temperature, B_p is the total bandwidth for the AP p, T is the length of the moving average, $C_{j,p}(t)$ is equal to 1 if UE j is connected to AP p at time t and 0 otherwise and $N_{j,p}(t)$ is the number of PRB allocated by AP p to UE j and $\#R_p$ is the total number of resource blocks of AP p.

The interference part is computed as follows:

$$\mathcal{I}_{i,p} = \sum_{p' \neq p} F_{p,p'} P_{i,p'} \cdot \Theta_{p'}(t) \tag{12}$$

where $F_{p,p'}$ is 1 if AP p and p' share the same carrier frequency and 0 otherwise.

Using (10) and (12) it is possible to compute the SINR, and so it is possible to estimate the data-rate that can be transmitted allocating one PRB to UE i using the Shannon formula

$$r_{i,p} = 2^{-\mu} 10^{-3} B_{PRB} \log_2\left(1 + SINR_{i,p}\right) \tag{13}$$

B_{PRB} is the bandwidth of a single PRB and it can be computed as $B_{PRB} = 12 \cdot 2^{\mu} 15$ kHz.

Now, given a certain bitrate request b_p^i from UE i, it is possible to compute the number of resource blocks to be allocated by AP p to satisfy the request: $n_{i,p}^{PRB} = \lceil (b_{pk}^i / r_{i,p}) \rceil$.

5.5. Satellite resource allocation description

Contrary to ground APs, the satellite APs use Time Division Multiple Access (TDMA) in order to serve multiple UEs at the same time. In this case, the minimum allocation unit is a *block of symbols* that occupies a certain time slot in the satellite time-frame.

The receiving power $P_{i,p}$ can be still computed as (9), but in this case the path loss function will be the Free Space Path Loss

$$L_{i,p}^{FSPL} = \left(\frac{4\pi d_{i,p} f}{c}\right)^2, \tag{14}$$

where $d_{i,p}$ is the Euclidean distance between UE i and AP p, f is the carrier frequency used and c is the speed of light.

The thermal noise can be computed as (10) and the interference can be computed as (12). Using the Shannon formula (considering that this time the bandwidth is the total bandwidth of the satellite AP since the TDMA utilises all the bandwidth only for a certain amount of time), one has that the bitrate obtainable by a single block of symbols is

$$r_{i,p} = bB \log_2\left(1 + SINR_{i,p}\right), \tag{15}$$

where b is the ratio between the number of symbols in a single block and the total number of symbols of the satellite AP. The number of blocks to be allocated for a requested bitrate b_{pk}^i from UE i is then computed as $n_{i,p}^{blocks} = \lceil (b_{pk}^i / r_{i,p}) \rceil$

6. Simulation results and validation

In order to demonstrate the effectiveness of the proposed approach, a simulative environment has been built up according to the model definition previously introduced.

6.1. Scenario definition

We developed a scenario consisting of four terrestrial access points (NR1 and NR2 are 5G NR APs and the remaining two are 4G LTE APs) and a satellite access point in a 2.5 × 2.5Km area, as shown in Figure 5.

In particular for 5G NR access points we considered a carrier frequency of 1.7GHz (band n66) with numerology $\mu = 2$, while for 4G LTE access points we considered a carrier frequency of 800MHz (band 20). All the terrestrial APs have 20dB power, 16dB antenna gain and 3dB feeder losses. For the satellite access point we considered the Inmarsat implementation from example 6.6.2 of [31]. A total of 100 UEs has been considered in the given area, and each of them follows a Poisson distribution for requesting data with a certain service type and for the duration of such request; the parameters for each service type are described in Table 1. Moreover, we considered $\gamma = 0.9$, $\varepsilon = 1$, ε-decay $= 0.9995$ and ϵ-min $= 0.01$. As for the DQN parameters, we considered a replay buffer of 2000 tuples, a batch size of 64 tuples and the update of target network weights happens every 50 steps. Finally, the DNN hidden layers have a *tanh* activation function, the learning rate of the DNN is 10^{-4} and the network performs $4 \cdot 10^{4}$ training steps before finishing the training.

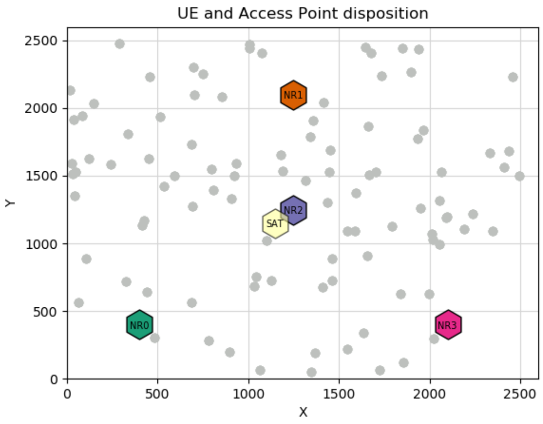

Figure 5: Considered network scenario

Table 1 Service type requests

	Elastic	*Non-elastic*	*Multi-codec*
Bitrate (Mbps)	10	200	100
Arrival rate (sec)	2	6	4
Dwelling time (sec)	30	120	90

6.2. Simulation results

Results displayed in the following graphs will focus on the performances of the controller in terms of QoS-flows allocation and their management. In order to validate the results of the proposed DQN algorithm, a set of other approaches have been simulated. In particular a classical, tabular, Q-Learning (QL in the figures) approach has been simulated, together with a Least Loaded (LL in the figures) approach, where a new request will be allocated to the least loaded AP, and a Max-RSRP (MR) approach, where a new request will be allocated to the AP with the maximum receiving power. The Q-Learning approach shares the same MDP representation as the one presented for the DQN, save for the fact that the state-space needed to be discretised so that the AP loads and the RSRP values contained in the states in (6) were uniformly quantised into four levels.

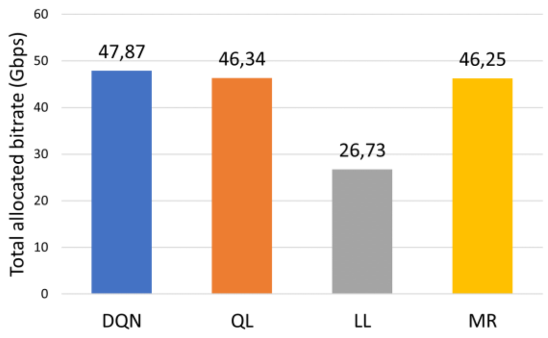

Figure 6: Overall rejection rates

The various controllers have been tested on the same scenarios to obtain fair results among their performances. Moreover, to ensure more balanced experiments, the results shown are the average between ten different scenarios, each one tested by all the different controllers. Finally, both the DQN and the QL controllers have been trained before the execution of the simulations. Several metrics are showed to better understand the performances of controllers with

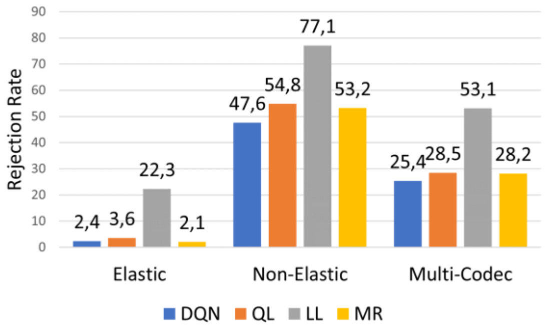

Figure 7: Rejection rates divided by service type

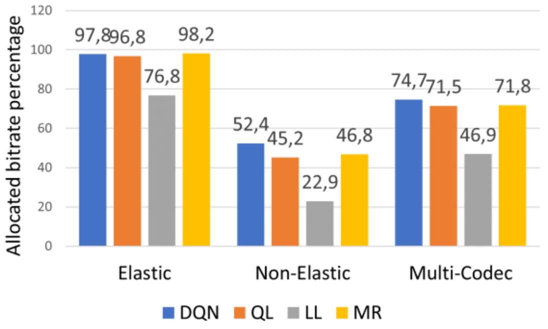

Figure 8: Allocated bitrate percentage divided by service type

respect each other.

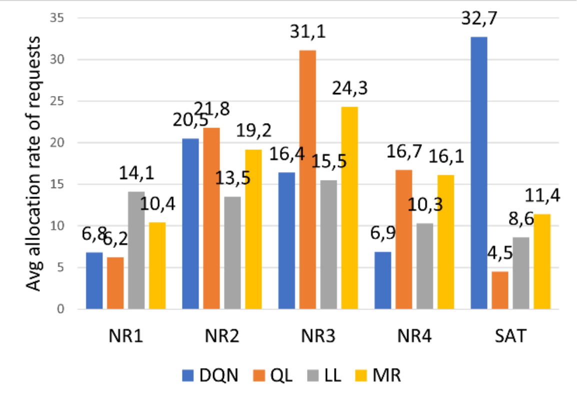

Figure 9: Load distribution among each AP

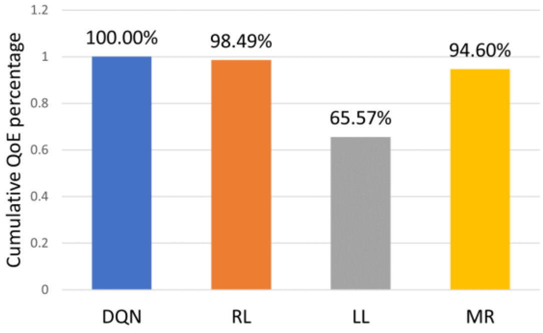

Figure 10: Cumulative QoE gained by each of the controllers with respect to DQN controller

As it emerges from Figure 6, the DQN controller outperforms the other controllers in terms of rejection rate, even if both the Max-RSRP (MR) one and the Q-Learning (RL) one have similar results. This behaviour is not surprising, since the Max-RSRP approach allocates requests to the AP with minimum path-loss, so the number of requested physical resources will be in general lower, and the Q-Learning approach has a similar behaviour w.r.t. the DQN approach, since the only difference is in its finite state space.

Figure 7 reports the rejection rate of each controller divided by service type. In the figure we can note how all controllers allocate a lower percentage of the non-elastic service requests, whereas the LL controller shows a significantly higher rejection rate for the elastic services.

In terms of bitrate, the DQN approach results to be the best one, allocating around 48Gbit over the accepted incoming requests.

Figure 8 details the allocated bitrate percentage with respect to the total requests bitrate divided by service type.

The result demonstrates that DQN behaves almost in the same way of MR for what concerns the elastic services, while it allocates about 6% more than the other approaches for what regards non-elastic traffic and about 3% for what regards the multi-codec requests.

Finally, from Figure 9, that represents the average percentage of successful allocations on each AP over all the requests made by UEs, it is evident that the Least-Loaded controller is the one that better balances the load among the APs: despite its limited performances according to the other metrics presented, due to its definition it allocates requests to the least used AP at the given time instant, resulting into an overall reasonable balance among all the APs.

The other controllers appear to be less balanced when allocating resources, with one or two Base Stations exploited more than the others. In particular, the DQN controller relies heavily on the Satellite Base Station to allocate incoming requests, allocating about 30% of requests to this AP. DQN is hence the only approach that manages to fully exploit Satellite resources, as the others tend to utilise mainly the NR Base Stations.

Figure 10 represents the QoE collected by each of the controllers. The values for each controller are computed summing the QoE gained by each request according to the QoE profiles defined in section 3 and then normalised on the result obtained by the DQN controller. As expected, the Q-Learning controller has similar performances with respect to the DQN one, that reaches the highest level of QoE. The performance gap increases when comparing a learning-based agent against the other approaches.

7. Conclusion

The paper proposed a Network Controller based on Deep Reinforcement Learning to enable the integration of satellite systems into 5G heterogeneous networks. The proposed controller dealt with the problem of Network Selection by formulating it as a Markov Decision Process, and was compared to several standard benchmark algorithms. The proposed solution proved to be able to cope with large scale scenarios involving 100 different UEs.

For validation purposes, the authors developed an open source network simulator [27] that realistically captures the network resource usage of different radio technologies, including satellite connections.

Overall, the proposed controller improved the performances of the network, increasing the connection-flows acceptance rate and providing a better resource management with respect to the other methods tested.

Future works are related to the introduction of other unmodeled complexities in the simulator, such as user and access point mobility. Actor-critic algorithms [32] will also be explored to enable the split of QoS-flows and multi-connectivity, allocating a single flow over different access points at the same time.

Acknowledgements

This work was supported by the 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. The authors acknowledge all their colleagues of the CRAT team working in the project for their fruitful discussions and confrontations.

References

[1] 3GPP, “ETSI TR 38.811 v15.4.0, Study on New Radio (NR) to support non-terrestrial networks,” 2020.

[2] V. Mnih et al., “Playing Atari with Deep Reinforcement Learning.” 2013.

[3] K. S. S. Anupama, S. S. Gowri, and B. P. Rao, “A Comparative Study of Outranking MADM Algorithms in Network Selection,” 2018, doi: 10.1109/iccmc.2018.8487931.

[4] Y. Zhong, H. Wang, and H. Lv, “A cognitive wireless networks access selection algorithm based on MADM,” Ad Hoc Networks, vol. 109, p. 102286, 2020, doi: 10.1016/j.adhoc.2020.102286.

[5] S. Radouche, C. Leghris, and A. Adib, “MADM methods based on utility function and reputation for access network selection in a multi-access mobile network environment,” 2017, doi: 10.1109/wincom.2017.8238177.

[6] 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.

[7] T. Ding, L. Liang, M. Yang, and H. Wu, “Multiple Attribute Decision Making Based on Cross-Evaluation with Uncertain Decision Parameters,” Math. Probl. Eng., vol. 2016, pp. 1–10, 2016, doi: 10.1155/2016/4313247.

[8] R. K. Goyal, S. Kaushal, and A. K. Sangaiah, “The utility based non-linear fuzzy AHP optimization model for network selection in heterogeneous wireless networks,” Appl. Soft Comput., vol. 67, pp. 800–811, 2018, doi: 10.1016/j.asoc.2017.05.026.

[9] X. Yan, P. Dong, T. Zheng, and H. Zhang, “Fuzzy and Utility Based Network Selection for Heterogeneous Networks in High-Speed Railway,” Wirel. Commun. Mob. Comput., vol. 2017, pp. 1–14, 2017, doi: 10.1155/2017/4967438.

[10] M.-M. R. Mou and M. Z. Chowdhury, “Service aware fuzzy logic based handover decision in heterogeneous wireless networks,” 2017, doi: 10.1109/ecace.2017.7912992.

[11] 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.

[12] R. Trestian, O. Ormond, and G.-M. Muntean, “Game Theory-Based Network Selection: Solutions and Challenges,” IEEE Commun. Surv. Tutorials, vol. 14, no. 4, pp. 1212–1231, 2012, doi: 10.1109/surv.2012.010912.00081.

[13] J. Antoniou and A. Pitsillides, “4G Converged Environment: Modeling Network Selection as a Game,” in 2007 16th IST Mobile and Wireless Communications Summit, Jul. 2007, pp. 1–5, doi: 10.1109/ISTMWC.2007.4299242.

[14] M. T. Rahman, M. Z. Chowdhury, and Y. M. Jang, “Radio access network selection mechanism based on hierarchical modelling and game theory,” 2016, doi: 10.1109/ictc.2016.7763451.

[15] L. Rajesh, K. B. Bagan, and B. Ramesh, “User Demand Wireless Network Selection Using Game Theory,” in Lecture Notes in Electrical Engineering, Springer Singapore, 2017, pp. 39–53.

[16] Meenakshi and N. P. Singh, “A comparative study of cooperative and non-cooperative game theory in network selection,” 2016, doi: 10.1109/icctict.2016.7514652.

[17] R. S. Sutton, A. G. Barto, and others, Introduction to reinforcement learning, vol. 135. MIT press Cambridge, 1998.

[18] Z.-H. Zhang, X.-F. Jiang, and H.-S. Xi, “Optimal content placement and request dispatching for cloud-based video distribution services,” Int. J. Autom. Comput., vol. 13, no. 6, pp. 529–540, 2016, doi: 10.1007/s11633-016-1025-z.

[19] F.-S. Lin, B.-Q. Yin, J. Huang, and X.-M. Wu, “Admission control with elastic QoS for video on demand systems,” Int. J. Autom. Comput., vol. 9, no. 5, pp. 467–473, 2012, doi: 10.1007/s11633-012-0668-7.

[20] Z. Du, C. Wang, Y. Sun, and G. Wu, “Context-Aware Indoor VLC/RF Heterogeneous Network Selection: Reinforcement Learning With Knowledge Transfer,” IEEE Access, vol. 6, pp. 33275–33284, 2018, doi: 10.1109/access.2018.2844882.

[21] Y. Yang, Y. Wang, K. Liu, N. Zhang, S. Gu, and Q. Zhang, “Deep Reinforcement Learning Based Online Network Selection in CRNs With Multiple Primary Networks,” IEEE Trans. Ind. Informatics, vol. 16, no. 12, pp. 7691–7699, 2020, doi: 10.1109/tii.2020.2971735.

[22] D. D. Nguyen, H. X. Nguyen, and L. B. White, “Reinforcement Learning With Network-Assisted Feedback for Heterogeneous RAT Selection,” IEEE Trans. Wirel. Commun., vol. 16, no. 9, pp. 6062–6076, 2017, doi: 10.1109/twc.2017.2718526.

[23] F. Liberati et al., “Stochastic and exact methods for service mapping in virtualized network infrastructures,” Int. J. Netw. Manag., vol. 27, no. 6, p. e1985, Nov. 2017, doi: 10.1002/nem.1985.

[24] X. Wang, J. Li, L. Wang, C. Yang, and Z. Han, “Intelligent User-Centric Network Selection: A Model-Driven Reinforcement Learning Framework,” IEEE Access, vol. 7, pp. 21645–21661, 2019, doi: 10.1109/access.2019.2898205.

[25] K.-S. Shin, G.-H. Hwang, and O. Jo, “Distributed reinforcement learning scheme for environmentally adaptive IoT network selection,” Electron. Lett., vol. 56, no. 9, pp. 462-464(2), Apr. 2020, [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/el.2019.3891.

[26] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning.” 2015.

[27] E. De Santis, “trunk96/wireless-network-simulator,” GitHub. [Online]. Available: https://github.com/trunk96/wireless-network-simulator.

[28] F. D. Priscoli, A. Giuseppi, F. Liberati, and A. Pietrabissa, “Traffic Steering and Network Selection in 5G Networks based on Reinforcement Learning,” 2020, doi: 10.23919/ecc51009.2020.9143837.

[29] 3GPP, “ETSI TS 138 211 v15.2.0, 5G NR Physical channels and modulation,” 2018.

[30] “Final report for COST Action 231.”

[31] G. Maral, M. Bousquet, and Z. Sun, Satellite Communications Systems. Wiley, 2020.

[32] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” arXiv Prepr. arXiv1509.02971, 2019.

Emanuele De Santis received his B.Sc. and M.Sc. degree in engineering in computer science respectively in 2017 and 2019, both summa cum laude, from Sapienza University of Rome, Rome, Italy, where he is currently a PhD student in automatic control. He participated in the H2020 projects 5G-ALLSTAR and 5G-Solutions and in the ESA project ARIES. His main research activities are in the field of power and communication network control, artificial intelligence and optimal control. He is a student member of IEEE.

E-mail: edesantis@diag.uniroma1.it (Corresponding author)

ORCID iD: 0000-0003-1011-9737

Alessandro Giuseppi was born in Rome, Italy, in 1992. He received the B.Sc. degree in computer and automation engineering in 2014, the M.Sc. degree in control engineering in 2016 and the Ph.D. in automatica in 2019, all summa cum laude, from the University of Rome La Sapienza Rome, Italy, where he is currently a postdoctoral researcher in automatic control.

Since 2016, he has participated in five EU and national research projects. His main research activities are in the fields of network control and intelligent systems. He is a member of IEEE.

E-mail: giuseppi@diag.uniroma1.it

ORCID iD: 0000-0001-5503-8506

Antonio Pietrabissa is an associate professor at Sapienza University of Rome, Rome, Italy, where he received his degree in electronics engineering and his PhD degree in systems engineering in 2000 and 2004, respectively. Since 2000, he has participated in about 20 EU and national research projects. His research focuses on the application of systems and control theory to the analysis and control of networks. He is a senior member of IEEE.

E-mail: pietrabissa@diag.uniroma1.it

ORCID iD: 0000-0003-0188-3346

Michael Capponi received the M.Sc. degree in engineering in computer science summa cum laude at Sapienza University of Rome, Rome, Italy, in 2020.

His research interests include reinforcement learning applications to communication networks.

E-mail: michaelcapponi96@gmail.com

Francesco Delli Priscoli was born in Rome, Italy, in 1962. He received the Graduate degree in electronics engineering (summa cum laude) and the Ph.D. degree in systems engineering from the University of Rome, La Sapienza, Rome, Italy, in 1986 and 1991, respectively.

From 1986 to 1991, he was with Telespazio, Rome, Italy. Since 1991, he has been with the University of Rome, La Sapienza, where, at present, he is a Full Professor of automatic control, control of autonomous multiagent systems, and control of communication and energy networks. His research interests include closed-loop multiagent learning techniques in advanced communication and energy networks.

Dr. Delli Priscoli is an Associate Editor of Control Engineering Practice and a Member of the IFAC Technical Committee on Networked Systems. He was/is the Scientific Responsible with the University of Rome, La Sapienza, for 40 projects funded by the European Union and by the European Space Agency. He is a member of IEEE.

E-mail: dellipriscoli@diag.uniroma1.it

ORCID iD: 0000-0001-6140-3661