Reinforcement Learning-Based Load Balancing Satellite Handover Using NS-3

Nour Badini[†§], Mona Jaber[§], Mario Marchese[†] and Fabio Patrone[†]
[†]Department of Electrical, Electronics and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Genoa, Italy
[§]School of Electronic Engineering and Computer Science, Queen Marry University of London, London, UK
Email: nour.badini@edu.unige.it, m.jaber@qmul.ac.uk, mario.marchese@unige.it, f.patrone@edu.unige.it

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279521

***Abstract*—The Fifth-Generation of Mobile Communications (5G) is intended to meet users' growing needs for high-quality services at any time and from any location. The unique features of Low Earth Orbit (LEO) satellites in terms of higher coverage, reliability, and availability, can help expand the reach of 5G and beyond technologies to support those needs. However, because of their high speeds, a single LEO satellite is unable to provide continuous service to multiple User Equipments (UEs) spread over a large (potentially worldwide) area, resulting in the need for LEO satellite constellations with a high number of satellites and a consequent high amount of satellite handovers (HOs). Moreover, UEs can only acquire partial information about the satellite system and compete for the limited available communication resources of the satellites, requiring the implementation of a decentralized satellite HO strategy to avoid network congestion. In this paper, we propose a decentralized Load Balancing Satellite HO (LBSH) strategy based on multi-agent reinforcement Q-learning, implemented within the software Network Simulator 3 (NS-3). LBSH aims to reduce the total number of HOs and the blocking rate while balancing the load distribution among satellites. Our results show that the proposed LBSH method outperforms the state-of-the-art methods in terms of a 95% drop in the average number of HOs per user and an 84% reduction in blocking rate.**

***Index Terms*—Reinforcement Learning, Satellite Handover, NS-3, 5G Satellite-Terrestrial Integrated Networks**

I. Introduction

5G and beyond communication technologies are driven by the exploding demand for heterogeneous, reliable, secure, low-latency, broadband, and high-speed services [1]. They are dedicated to connect humans and machines at any time and location. However, terrestrial networks cannot provide Internet access to users on airplanes, ships, high-speed trains, highways, or to very remote areas, such as mountains or islands, or where it is too expensive to deploy terrestrial networks. While on the other hand, Non-Terrestrial Networks (NTN), which include satellites, Unmanned Aerial Vehicles (UAVs), and High Altitude Platforms (HAPs), are the most effective means to connect the world's unconnected, unserved, and underserved areas with high reliability as they are not limited by geography and are not affected by natural disasters or wars [2]. Thus, NTNs can help expand the reach of the next-generation communication technologies and achieve some of their Key Performance Indicators (KPI), especially in terms of increased coverage, reliability, and availability. In particular, Low Earth Orbit (LEO) satellites, whose altitudes range from 500km to 1500km, have attracted the interest of many researchers since they have the advantages of low propagation delay, low energy consumption, reduced transmission power requirements, and suppressed signaling attenuation [3]. However, LEO satellites orbit Earth at high speeds, therefore, they typically operate in multi-satellite constellations to simultaneously cover large areas of the world. As a consequence, each satellite visibility time from a ground user is limited and the implementation of a flexible Handover (HO) strategy is needed in such networks. In addition, without a centralized controller, User Equipment (UE)s can only partially obtain information about the satellite system and compete for the limited satellite communication resources. This requires the implementation of a distributed satellite HO strategy to avoid network congestion.

In this paper, we propose a decentralized Load Balancing Satellite Handover (LBSH) strategy based on multi-agent reinforcement Q-learning, that reduces the total number of HOs and the blocking rate while balancing the load among all the satellites in the network. The proposed LBSH is proven to outperform state-of-the-art methods when these are implemented in a realistic simulation setting using Satellite-Terrestrial Integrated Network (STIN) simulator based on the Network Simulator 3 (NS-3) which makes it close to the real case scenario.

The rest of the paper is organized as follows. Section II reports the main related works for implementing satellite HOs. Section III describes the simulator used to implement the satellite HO strategy and presents the HO optimization problem. In Section IV, the optimization problem is transformed into a reinforcement learning problem, starting with a single-agent Q-learning followed by a multi-agent Q-learning. Simulation results were discussed in Section V. Finally, Section VI provides the conclusions of the presented work.

II. Related Works

Since the position of the base stations in terrestrial networks is fixed, users typically perform HO due to users' movements and based on the measured received signal strength, reference signal received power, or reference signal received quality [4]. However, LEO satellites move at a very high speed and rapidly change their footprints on the Earth's surface, which makes the satellites' movements the main reason for HO and the above measurements not fully applicable. Thus, we must consider

other parameters, such as remaining service time, number of available channels, and received signal strength. For example, the number of available satellite channels was considered as the basic HO criterion in [5] and [6]. The authors in [5] divided the multimedia traffic into two types and the satellite HO requests are addressed based on the queue state of each traffic type to ensure a low drop blocking and forced termination probability. The authors in [6] proposed a dynamic Doppler-based HO prioritization scheme which employs Doppler shift monitoring to estimate the actual number of HO requests and the actual time of occurrence in order to avoid resource reservation. The above HO criteria can achieve a balanced load in the system, but they do not guarantee good communication quality. On the other hand, the maximum elevation angle HO criterion was proposed in [7] and [8]. Authors in [7] presented a hard HO scheme and in [8] presented a hybrid channel adaptive HO scheme for the satellite HO, both of which considered the elevation angle as the satellite HO criterion. However, the current elevation angle does not necessarily reflect the actual performance of the network as it may lead to channel congestion and thus increased blocking rate. A graph theory-based satellite HO framework was proposed in [9], where the authors set different weights for the edges in the satellite-connected graph based on different satellite selection criteria and then use the shortest path algorithm to obtain the user's optimal HO scheme. Authors in [10] considered three HO criteria for the satellite selection (maximum elevation angle, longest visible time, and maximum idle channels) aim to provide users a low forced termination probability while meeting Quality of Service (QoS) limits even in high traffic conditions. Recently, authors in [11] proposed a multi-agent reinforcement learning HO strategy that aims to minimize the number of HOs in a load-aware manner. To this end, the method considers the minimum elevation angle and the available satellite channels in the UE-to-satellite allocation and is shown to reduce the blocking rate compared to load-unaware schemes.

Most of the state-of-the-art studies, either consider one HO criterion for a specific optimization goal or provide a solution that takes several criteria into account from the viewpoint of a single user. Nevertheless, without a central controller, users can only acquire partial information about the satellite system with respect to themselves. Furthermore, because a satellite's channel budget is limited, competition for available channels among users served by the same satellite may result in a highly imbalanced satellite load. This necessitates the implementation of a decentralized satellite HO strategy that takes into account the users' real-time resource competition. The use of multi-agent reinforcement learning in [11] solved the decentralized problem by considering each user as an agent that has a partial view of the system, and therefore take individual actions. This HO approach is load-aware which avoids connecting to a fully loaded satellite, however, it does not prioritize the connection to a satellite with higher available channels. Hence this method does not promote load balancing among the satellites which increases the probability of blocking per UE. In contrast, the LBSH method proposed in this work is a load-balancing HO scheme that successfully reduced the blocking rate, as will be demonstrated in Section V.

III. 5G Satellite Handover

The 5G STIN simulator proposed in [12] is used in this work to evaluate the performance of leading HO schemes, including the proposed LBSH method. This simulator is built on top of NS-3 that simulates packet data networks with user-defined traffic models [13]. In addition, the 5G-LENA module [14] which is employed to model 5G New Radio (NR) cellular networks [15] and the SGP4 mathematical model is used to estimate the speed and position of LEO satellites [16].

A. Reference scenario and Network Assumptions

We implemented our network considering the satellite HO problem over a specific period of time T, as illustrated in Figure 1. The main components of the network are:

- *5G UEs*: terrestrial or aerial nodes that can connect to the Internet to send or receive data through a 5G Radio Access Network (RAN). They are uniformly distributed across the Earth's surface. A set of *K* users is considered and denoted by $\mathcal{K} = \{1, 2, ..., K\}$.
- *LEO Satellites*: satellite nodes that make up the 5G RAN. Each satellite can produce a 5G cell that provides direct access to the 5G UEs. A set of *N* satellites was considered and denoted by $\mathcal{N} = \{1, 2, ..., N\}$.

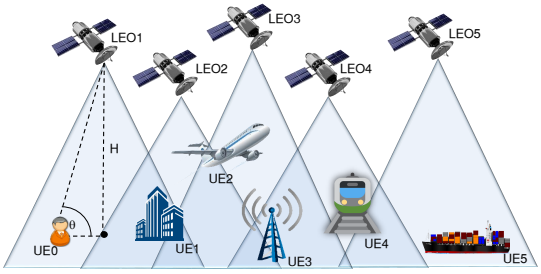

Fig. 1. Satellite Handover Scenario

The elevation angle between user k and satellite n is represented by $\theta_{k,n}$ and can be calculated based on the position information of the user and its covering satellite as follows:

$$\theta_{k,n} = \frac{\arcsin(H_n^2 + (2 \times R_e \times H_n) - D_{k,n}^2)}{2 \times R_e \times D_{k,n}} \tag{1}$$

where H_n is the altitude of satellite n, R_e the Earth radius, and $D_{k,n}$ the distance between user k and satellite n. The minimum elevation angle θ_0 is a design parameter used to ensure threshold link quality. As a result, a satellite n is considered a good candidate for user k only if:

$$\theta_{k,n} \geq \theta_0, \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{2}$$

At time t, $C_{k,n}^t$ is the coverage indicator between satellite n and user k, and it is defined as follows:

$$C_{k,n}^t = \begin{cases} 1 & \text{if user } k \text{ is covered by satellite } n \text{ at time } t, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Moreover, $X_{k,n}^t$ indicates if user k is served by satellite n at time t, and is defined as follows:

$$X_{k,n}^t = \begin{cases} 1 & \text{if user } k \text{ is served by satellite } n \text{ at time } t, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We assume that each satellite's total bandwidth is partitioned into L_n channels of equal bandwidth and that each user can only use one channel to transmit/receive. Following that, the channel budget limitation is given by:

$$\sum_{k \in \mathcal{K}} X_{k,n}^t \leq L_n, \qquad \forall n \in \mathcal{N} \tag{5}$$

B. Problem Formulation

At any time t in T, the selection from candidate satellites ($C_{k,n}^t = 1$) is optimized to minimize the number of HOs. In this case, any decision that results in $X_{k,n}^t \neq X_{k,n}^{t-1}$ at any time t during T causes the HO count to increase by 1. The average number of HOs is thus given by:

$$HO_{avg} = \frac{\sum_{k \in \mathcal{K}} HO_k}{K} \tag{6}$$

Our goal is to decrease the average number of HOs while enhancing channel utilization efficiency in the LEO satellite system during the considered time period T.

The optimization problem is therefore defined as follows:

$$\min HO_{avg} \tag{7a}$$

$$s.t \sum_{k \in \mathcal{K}} X_{k,n}^t \leq L_n, \qquad \forall n \in \mathcal{N}, X_{k,n}^t \in \{0,1\}, \tag{7b}$$

$$\theta_{k,n} \geq \theta_0, \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{7c}$$

where (7b) is the constraint of the total number of available channels of a satellite n given the two possible states of connection between user k and satellite n represented by $X_{k,n}^t$, which ensures channel utilization efficiency, and (7c) is the constraint for the minimum acceptable elevation angle that guarantees a good communication link quality. Problem (7) is an NP-hard combinatorial integer optimization problem in general and we convert it into a Reinforcement Learning (RL) optimization problem based on a stochastic game.

IV. 5G Satellite Handover based on Reinforcement Learning

RL is a computational method for understanding and automating goal-directed learning and decision-making. It differs from other computational approaches as it focuses on an agent learning through direct interaction with its environment, rather than requiring ideal supervision or entire models of the environment. It can learn anything from scratch by pursuing a goal that can be defined as the maximization of the expected value of a cumulative sum of a received scalar signal called reward. RL defines the interaction between a learning agent and its environment in terms of states, actions, and rewards by using the formal framework of Markov decision processes. This framework is intended to be a straightforward way of representing key aspects of the artificial intelligence problem. These characteristics include a sense of cause and effect, uncertainty, non-determinism, and the presence of explicit goals [17]. The main components of our RL framework are:

- *Agent*: The component that makes the decision of what action to take at each step which may cause a transition to a new state. We consider each user to be an agent, i.e., the set of agents $\mathcal{A}$ is equal to the set of users $\mathcal{K}$.
- *Environment*: The world in which the agent lives and interacts by taking some actions, but those actions cannot influence the environment's rules or dynamics In this paper, the environment is the NS-3-based 5G satellite integrated network.
- *State*: The observations that the agent receives from the environment. We define the state of agent k at time t as the 3-tuple, $s_k^t =< \overline{C_k^t}, \overline{l^t}, \overline{V_k^t} >$, where $\overline{C_k^t}$, $\overline{l^t}$, and $\overline{V_k^t}$ are all vectors of size N, such that, $\overline{C_k^t} = [C_{k,0}^t, C_{k,1}^t, ..., C_{k,n}^t, ..., C_{k,N}^t]$ contains the coverage indicators between user k and each satellite $n \in \mathcal{N}$, $\overline{l^t} = [l_0^t, l_1^t, ..., l_n^t, ..., l_N^t]$ indicates the number of loaded channels of each satellite $n \in \mathcal{N}$ at time t, and $\overline{V_k^t} = [V_{k,0}^t, V_{k,1}^t, ..., V_{k,n}^t, ..., V_{k,N}^t]$ includes the Remaining Visibility Time (RVT) between user k and each satellite $n \in \mathcal{N}$ at time t. $\mathcal{S}$ indicates the set of states.
- *Action*: Represents the decision taken by an agent which is to connect to one of the satellites $n \in \mathcal{N}$. In this paper, an action of an agent k at time t is defined as a_k^t where a_k^t is equal to one of the satellites $n \in \mathcal{N}$ such that $C_{k,n}^t = 1$.
- *Reward*: A motivation mechanism that uses reward or penalty. The instantaneous reward of an agent k, after an action a_k^t was taken knowing that it is in state s_k^t, is represented by $r_k^t(s_k^t, a_k^t)$. Considering that agent k chooses to connect to satellite n at time t (ie. $a_k^t = n$):

$$r_k^t(s_k^t, a_k^t) = \begin{cases} -p_1 & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 0, \\ -p_2 & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 1, \\ & l_n^t < \sum_{k \in \mathcal{K}} X_{k,n}^t, \\ f(t,n,k) & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 1, \\ & l_n^t \geq \sum_{k \in \mathcal{K}} X_{k,n}^t \end{cases} \tag{8}$$

  A high penalty is associated with the instantaneous reward function when an action results in a HO, and a lower penalty when the action results in blocking. However, when the action avoids HO and blocking, a positive reward is given such that it is higher when the agent chooses to connect to a satellite with higher RVT and lower load. $p_1 = 300, p_2 = 100$ and $f(t,n,k) = v_{k,n}^t + w_n^t$, where $w_n^t = l_n^t - \sum_{k \in \mathcal{K}} X_{k,n}^t$ represents the number of available channels of satellite n at time t.
- *Policy*: A strategy used by the agent to achieve its goal. The policy directs the agent's actions based on the agent's current state. The goal of agent k is to find an optimal policy π_k^* that maximizes the expected cumulative reward:

$$\pi_k^* = arg \max_{\pi} R^k(s, \pi) \tag{9}$$

  where $R^k(s,\pi) = \sum_{t=0}^{T} \gamma E\{r_k^t | s^0 = s, \pi\}$ is the expected cumulative reward of agent k and $\gamma \in [0,1)$ is

the discount factor used to increase/decrease the weight of new rewards in comparison to previously stored rewards.

The goal of the proposed LEO satellite HO optimization problem, is equivalent to Eq. (9), which aims to find an optimal policy to maximize the expected cumulative reward over time.

A. Single-Agent Q-learning Algorithm

Q-learning is a model-free RL algorithm that can be used to learn the value of an action in a given state. It can handle problems with stochastic transitions and rewards without requiring adaptations [18]. Q can be learned through trial-and-error interactions with the environment by running through a large Number of Episodes (NEP) (training duration in which a sequence of states, actions, and rewards is considered), and thus through as many state/action pairs as possible. "Q" refers to the function that involves a simple updating procedure in which the agent starts with arbitrary initial values of $Q(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, and updates the Q-values as follows:

$$Q_{t+1}(s^t,a^t) = (1-\alpha_t)Q_t(s^t,a^t) + \alpha_t[r^t + \gamma \max_a Q_t(s^{t+1})] \quad (10)$$

where $\alpha_t \in [0,1)$ is the learning rate.

When designing an action selection policy in RL, it is critical to balance exploitation and exploration. Exploitation occurs when agents choose the best action based on the current Q-values, also known as a greedy policy. Exploration entails the agents attempting more actions that have not been exploited yet in order to explore a larger action space. To make better random actions, we combine Boltzman exploration and ϵ-greedy policy. An action's selection probability, denoted by $\pi_t(a^t)$ is weighted by its associated Q-value as follows:

$$\pi_t(a^t) = \frac{\exp \frac{Q^k(s_k^t,a^t)}{\tau}}{\sum_{a^t} \exp \frac{Q^k(s_k^t,a^t)}{\tau}} \quad (11)$$

where τ is called the temperature factor. It controls the amount of exploration, i.e., the probability of executing actions other than the one with the highest Q-value. When τ is high, all actions will be explored equally; when it is low, high-rewarding actions will be chosen with higher probability.

Despite the fact that this policy is viewed as a random action selection policy, the agents have a greater chance of choosing good actions due to the property of the probability function. Thus, given an exploration parameter $\varepsilon \in [0,1)$, we have:

$$a_*^t = \begin{cases} arg\max_{a^t} \pi_t(a^t) & \text{if } \epsilon < \varepsilon, \\ arg\max_{a^t} Q_k^t(s_k^t,a^t) & \text{otherwise} \end{cases} \quad (12)$$

In this subsection, a single agent is considered (only one UE). During each episode, at each time step, the agent observes the state s and selects an action a based on a policy π. The Q-table of this agent is then updated following Eq. (10). Since satellites are continuously moving, the covering set of satellites for the user, along with the corresponding RVT, changes at each time instance, causing a transaction to a new state independently by the taken action, while, at the same time, the action of the agent may alter each satellite load, causing the transition to a new state too. This results in a huge number of possible states that is hard to predict and define at the beginning of the learning. To solve this problem, the procedure shown in Algorithm 1 is used.

At the beginning of the learning ($t = 0$), there is only one state in the set of states ($\mathcal{S} = \{s^0\}$). Then, at each time step, after an action is taken, the load allocation of each satellite, the RVT and the set of covering satellites of the user, changes accordingly, causing the agent to move to a new state s'. If s' is already included in the set of states, its Q-value will be updated, otherwise, s' is added to the set of states ($\mathcal{S} = \{s^0, s_1, ..., s'\}$ with an initial Q-value of zero.

Algorithm 1 Single-Agent Q learning

Initialize:
$t = 0$, $\mathcal{S} = \{s^0\} =< \overline{C^0}, \overline{l^0}, \overline{V^0} >$;
$Q_0(s^0, a^0) = 0$
while $ep < NEP$ **do**
  while $t < T$ **do**
    observe $s^t =< \overline{C^t}, \overline{l^t}, \overline{V^t} >$;
    choose action a^t based on policy $\pi(s^t)$;
    move to a new state $s^{t+1} =< \overline{C^{t+1}}, \overline{l^{t+1}}, \overline{V^{t+1}} >$;
    get the reward r^t;
    if $s^{t+1} \in \mathcal{S}$ **then**
      update the Q-value $Q_t(s^t, a^t)$ by Eq. (10);
    else
      add the new state s^{t+1} to $\mathcal{S}$;
      initialize the Q-value of the new state to zero;
    end if
  end while
end while

B. Multi-Agent Q-learning

In this subsection, we will define our multi-agent LBSH method considering six UEs. Each UE is a RL agent that interacts with the environment based on its partial view of its current state. Considering Eq. (12), instead of having a constant ϵ throughout the learning process, a variable ϵ_t that increases linearly with time is considered to encourage the agents to explore more at the beginning of the learning and then, as ϵ_t increases, the agents start to explore with lower probability. Note that ϵ_t stops increasing when it reaches a value of 0.8 to let the agents still explore with a certain probability. As shown in Algorithm 2, at each time step, the agents observe their current state s_k^t and select an action a_k^t based on a policy π, acquire a corresponding reward r_k^t, and update their own Q-table following Eq. (10). The agents are considered to take actions successively one after another, where the sequence of learning for the agents is randomly chosen at each instant. After agent k chooses an action, the load of the satellites will change accordingly, which can be obtained by the other agents before the latter takes its action. It is assumed that each agent has no knowledge about the reward function of other agents, however, they can get each other actions.

Algorithm 2 Multi-Agent Q learning

Initialize:
$t = 0$, $\mathcal{S}_k = \{s_k^0\} =< \overline{C_k^0}, \overline{l^0}, \overline{V_k^0} > \forall k \in \mathcal{K}$;
Satellites-Load= l^0;
$Q_k^0(s^0, a^0) = 0$
while $ep < NEP$ **do**
  while $t < T$ **do**
    for $k \in \mathcal{K}$ **do**
      choose random agent k;
      observe $s_k^t =< \overline{C_k^t}, \overline{l^t}, \overline{V_k^t} >$;
      choose action a_k^t based on policy $\pi(s_k^t)$;
      move to a new state $s_k^{t+1} =< \overline{C_k^{t+1}}, \overline{l^{t+1}}, \overline{V_k^{t+1}} >$;
      get the reward r_k^t;
      update Satellites-Load to l^{t+1};
      if $s_k^{t+1} \in \mathcal{S}_k$ **then**
        update the Q-value $Q_k^t(s_k^t, a_k^t)$ by Eq. (10);
      else
        add the new state s_k^{t+1} to $\mathcal{S}$;
        initialize the Q-value of the new state to zero;
      endif
    endfor
  end while
end while

V. Simulation Results

A. Single-Agent Q-Learning

We test the single-agent learning by considering one UE whose position is fixed on the Earth's surface and 48 LEO satellites continuously moving around the Earth following the SGP4-module [16]. Table I summarizes all the simulation parameters considered in our approach for both single-agent and multi-agent tests.

TABLE I
Simulated scenario parameters

Parameter	Single-Agent	Multi-Agent
Number of satellites	48	48
Number of UE	1	6
Satellite altitude	600 km	600 km
Orbital planes eccentricity	0 (circular)	0 (circular)
Orbital planes inclination i	88°	88°
Orbital planes argument of perigee	90°	90°
Minimum elevation angle between UE and gNB for transmissions	20°	20°
Number of satellite available channels	1	5
α	0.1	0.1
γ	0.95	0.95
ϵ	0.82	0.1-0.8
τ	10	10
NEP	5000	2500
Duration of an episode (T)	600 s	600 s

Figure 2 shows the number of HOs as a function of NEP. As NEP increases from 0 to 5000, the number of HOs significantly decreases from about 190 to about only 3 HOs. This proves the optimization of the HO problem when considering only one agent.

B. Multi-Agent Q-Learning

We test the proposed multi-agent LBSH optimization approach and compare it to different approaches in a real case scenario within the developed NS-3 based 5G STIN simulator

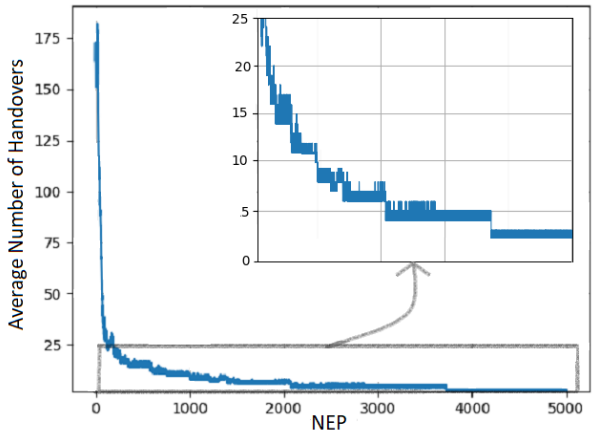

Fig. 2. Single-Agent: Average number of handovers as a function of NEP

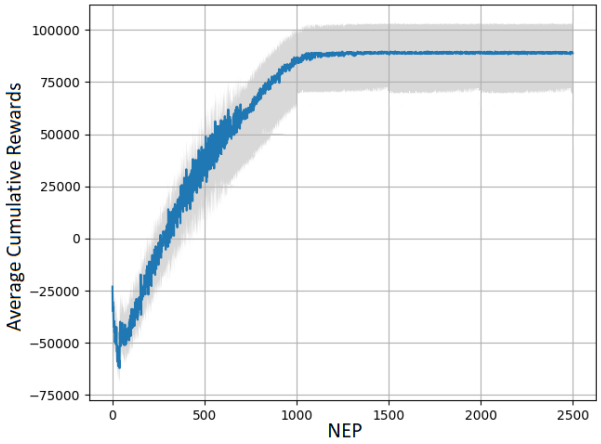

Fig. 3. Multi-Agent: Average Cumulative Reward as a function of NEP

[12]. For the LBSH approach proposed in this paper, ϵ is assumed to increase linearly from 0.1 to 0.8 while NEP increases. The reward function defined in Eq. (8) is adopted. Figure 3 shows the average cumulative rewards as a function of the NEP. As NEP increases from 0 to 2500, the cumulative reward increases from -60000 to about 950000, starting to converge after 1000 episodes. The reason is that in the first episodes the agents explore more, and then start to exploit more following Eq. (12). The grey shade in Figure 3 represents the range of the cumulative reward for all the six agents during learning. We first compare our work with a non-smart HO strategy where the HO decision is taken based only on the minimum distance between the UE and its associated satellite and the RVT between them. The minimum distance and the RVT between each UE-satellite pair are continuously traced. Accordingly, at each instant, each UE chooses to be connected to the satellite closest to it with higher RVT. The second approach is a Load Aware Satellite Handover (LASH) proposed in [11]. The reward function in LASH follows the same structure as in Eq.8, with $p1 = 20$, $p2 = 10$, and $f(t, n, k) = v_{k,n}^t$ which is limited to the remaining visibility time $v_{k,n}^t$, thus rendering LASH a load-aware HO strategy. In

contrast, in the LBSH, $f(t,n,k) = v^t_{k,n} + w^t_n$ accounts for the actual load of the satellite, and therefore, is a load-balancing scheme.

Moreover, at each instant, the action in [11] is chosen based on a fixed exploration parameter ϵ rather than an adaptable ϵ_t. Figure 4 shows the average number of HOs as a function of the NEP for the three different approaches.

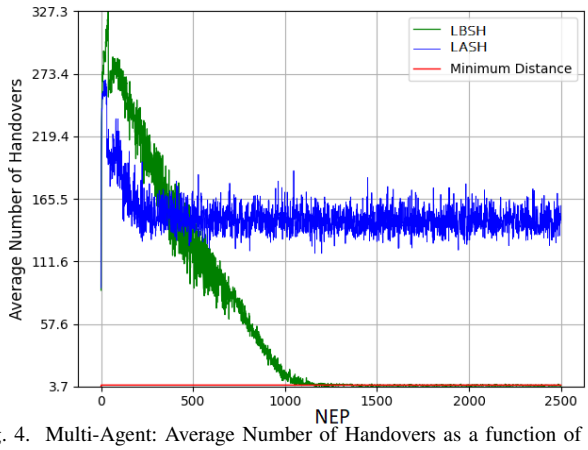

Fig. 4. Multi-Agent: Average Number of Handovers as a function of NEP

The LBSH approach proposed in this paper outperforms the LASH approach implemented in [11] when implemented in a realistic simulator, as the final number of HOs per user is 95% lower. Besides, the proposed LBSH converges to the same average HOs value of the Minimum distance approach which is around 3.7 HOs in the 600 s episode duration.

The blocking rate of the user k at time t is denoted by Br^t_k and defined as follows:

$$BR^k = \frac{\sum_t BN^k_t}{T} \tag{13}$$

where BN^k_t indicates whether user k was dropped at time t:

$$BN^k_t = \begin{cases} 1 & \text{if user } k \text{ chooses to connect to satellite } n \text{ and} \\ & l^t_n < \sum_{k \in \mathcal{K}} X^t_{k,n}, \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

As shown in Figure 5, the minimum distance approach results in the maximum blocking rate since it does not take the load constraint of each satellite into consideration. Moreover, the proposed LBSH approach results in the minimum blocking rate which converges to a value of 0.0042 (0.42%) and hence outperforms the LASH approach by around 84%. These results suggest that a load balancing scheme avoids allocating UEs to loaded satellites, and therefore, mitigates the risk of blocking.

VI. Conclusion

We presented a novel load balancing satellite HO scheme, LBSH, that is suitable for 5G STIN. The strategy has been implemented and tested within a simulation environment based on the software NS-3 and the obtained results show that it outperforms state-of-the-art solutions by reducing the average number of HOs by 95% and the blocking rate by 84%. This implies that LBSH is a promising technique to manage 5G STIN HOs whilst still effectively exploiting the available channels of satellite-generated 5G cells.

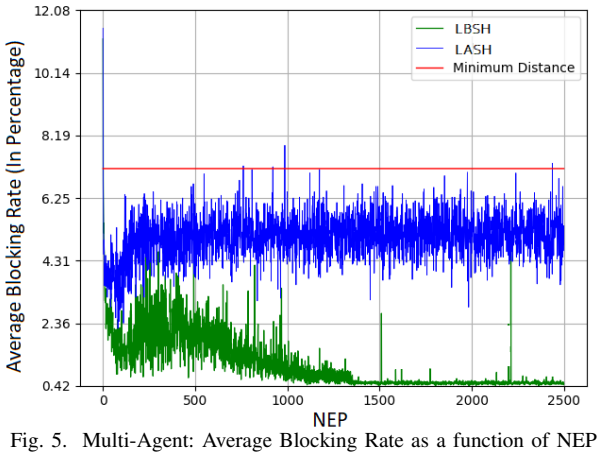

Fig. 5. Multi-Agent: Average Blocking Rate as a function of NEP

References

[1] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli, "Integration of Satellites in 5G through LEO Constellations," in *Global Communications Conference (GLOBECOM)*. IEEE, 2017, pp. 1–6.

[2] G. Giambene, E. O. Addo, and S. Kota, "5G aerial component for IoT support in remote rural areas," in *5G World Forum (5GWF)*. IEEE, 2019, pp. 572–577.

[3] I. Leyva-Mayorga, B. Soret, M. Röper, D. Wübben, B. Matthiesen, A. Dekorsy, and P. Popovski, "LEO small-satellite constellations for 5G and beyond-5G communications," *IEEE Access*, vol. 8, pp. 184 955–184 964, 2020.

[4] Y. I. Demir, M. S. J. Solaija, and H. Arslan, "On the Performance of Handover Mechanisms for Non-Terrestrial Networks," in *Vehicular Technology Conference: (VTC2022-Spring)*. IEEE, 2020, pp. 1–6.

[5] S. Karapantazis and F.-N. Pavlidou, "QoS handover management for multimedia LEO satellite networks," *Telecommunication Systems*, vol. 32, no. 4, pp. 225–245, 2006.

[6] E. Papapetrou and F.-N. Pavlidou, "Analytic study of Doppler-based handover management in LEO satellite systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 830–839, 2005.

[7] M. Gkizeli, R. Tafazolli, and B. Evans, "Modeling handover in mobile satellite diversity based systems," in *Vehicular Technology Conference (VTC) Proceedings*. IEEE, 2001, pp. 131–135.

[8] M. Gkizeli, R. Tafazolli, and B. G. Evans, "Hybrid channel adaptive handover scheme for non-GEO satellite diversity based systems," *IEEE Communications Letters*, vol. 5, no. 7, pp. 284–286, 2001.

[9] Z. Wu, F. Jin, J. Luo, Y. Fu, J. Shan, and G. Hu, "A graph-based satellite handover framework for LEO satellite communication networks," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1547–1550, 2016.

[10] E. Papapetrou, S. Karapantazis, G. Dimitriadis, and F.-N. Pavlidou, "Satellite handover techniques for LEO networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 2, pp. 231–245, 2004.

[11] S. He, T. Wang, and S. Wang, "Load-aware satellite handover strategy based on multi-agent reinforcement learning," in *Global Communications Conference (GLOBECOM)*. IEEE, 2020, pp. 1–6.

[12] N. Badini, M. Marchese, and F. Patrone, "NS-3-based 5G Satellite-Terrestrial Integrated Network Simulator," in *21st Mediterranean Electrotechnical Conference (MELECON)*. IEEE, 2022, pp. 1–6.

[13] [Online]. Available: https://www.nsnam.org/

[14] [Online]. Available: https://5g-lena.cttc.es/

[15] N. Patriciello, S. Lagén, B. Bojović, and L. Giupponi, "NR-U and IEEE 802.11 Technologies Coexistence in Unlicensed mmWave Spectrum: Models and Evaluation," *IEEE Access*, vol. 8, pp. 71 254–71 271, 2020.

[16] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, "Revisiting spacetrack report# 3," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[18] F. S. Melo, "Convergence of Q-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.