DQN-based Handover Optimization for LEO Satellites in NTN

Shadab Mahboob
The Bradley Department of Electrical and Computer Engineering
Virginia Tech
mshadab@vt.edu

Abstract

In this project, we aim to optimize the handover procedure involving Low Earth Orbit (LEO) satellites in Non-Terrestrial Networks (NTN). We introduce a Deep Q-Network (DQN) model to solve this problem considering different handover criteria such as propagation loss, potential service time, and quality. We show that our approach meets the convergence demonstrating the learning process of the agent. We also show the effect of different hyperparameters on the performance of this model in the end. **The GitHub repository link for the project: https://github.com/Shadab442/dqn-leo-handover-python.git**

1 Introduction

Non-Terrestrial Networks (NTN) have emerged as a critical element of the 6th Generation (6G) networks, promising uninterrupted, widespread, and scalable services [1]. With their proximity to Earth and lower cost of launch and maintenance, Low Earth Orbit (LEO) satellites, operating at an altitude of approximately 550-600 km, have gained significant attention in recent years as a potential component of NTN. However, these satellites must travel at extremely high speeds, often reaching up to 7.8 km/s to maintain their low height. The rapid orbital motion of LEO satellites, with an average orbital period of around two hours, presents a significant obstacle to integrating them into wireless communication systems. Due to the shorter orbital period, terrestrial User Equipment (UE) is only visible to an LEO satellite for a brief period of several minutes, requiring frequent handovers regardless of mobility [2]. These frequent handovers result in service interruptions, increased overheads, and higher energy consumption, leading to a degradation in network performance and efficiency. In this project, We aim to optimize the handover procedure involving these LEO satellites in a Deep Reinforcement Learning (DRL) framework considering the experienced path loss, potential visibility/ service time, and potential service quality as handover criteria [2].

2 Related works

In the mega constellation of LEO satellites, each non-terrestrial User Equipment (UE) is often covered by multiple satellites at any given time. The UEs can choose the most optimal connection from the available candidate LEO satellites based on different handover criteria like received signal strength, network load, potential service time, service quality, etc. An RL framework can be naturally adopted for solving this problem considering the handover criteria as states and UEs as agents who act by selecting a suitable LEO satellite and collecting a reward based on the network performance metrics. In [3], only the overall signal quality of the network is maximized using the RL approach without considering any other criteria. In [4, 5], a multi-objective optimization problem considering satellite load and signal quality constraints is solved using a DRL approach. In real networks, we have a large number of UEs; the handover decision for one UE can affect another UE, so the handover

problem needs to be solved in a cooperative manner. In [6], a MARL framework is considered where multiple UEs cooperatively optimize the number of handovers in the whole network considering different handover criteria. In [7], using graph matching, a database of optimum handover decisions in satellite networks is produced and later it is used to predict handover decisions using a CNN model. Advanced DL architectures like Auction based DL [8], DDQN [9], Successive DQN [10], etc. are also considered to provide optimal handover decisions.

3 Environment

We consider an environment which is a typical LEO satellite network consisting of a set of N satellites denoted by $\mathbb{N} = \{1, 2, ..., j, ..., N\}$ and a single UE which connected to one of the satellites at the starting time instant based on received signal strength. There is a Handover Controller (HC) for this UE located in a Radio Access Network (RAN) Intelligent Controller (RIC) connected to its serving satellite. This HC takes care of necessary handover control signaling to transfer the UE from one satellite to another. We divide the total time into T time slots indexed by $\{1, 2, 3, ..., t, ..., T\}$. At the end of each time slot t, the user is handed over by its HC by choosing a suitable satellite from the set of candidate satellites, $\mathbb{N}$. This handover decision is defined by a binary decision variable, $x_j^t \in \{0, 1\}$. Here x_j^t is 1 if the UE is handed over to the satellite j at the end of time instant t, otherwise, it is 0.

We assume the HC controller has access to three different network parameters related to the user, e.g., experienced wireless propagation loss, potential service time, and expected service quality. The propagation loss can be modeled as a combination of free space path loss, shadow fading, and Rayleigh fading and is denoted by p_j^t where j and t denote the candidate satellite and the time slot as before. The potential service time can be calculated based on the UE location and the UE trajectory information which is denoted by v_j^t. The service quality can be represented by the average elevation angle on that trajectory experienced by the user with respect to a candidate satellite j and is denoted by l_j^t. These parameters will be used as deciding criteria for handover decisions for the user.

4 Learning Algorithm

In this section, we formulate a Reinforcement Learning (RL) framework for optimizing handover decisions for the user and then discuss the considered DQN framework for solving this problem.

4.1 Agents

The agent is the HC dedicated to the user connected to a satellite in the network.

4.2 States

The set of path measurements for different satellites for the user at the time slot t can be denoted by $\mathbb{P}^t = \{p_1^t, p_2^t, ..., p_j^t, ..., p_M^t\}$. The set of potential service times for different satellites for the user at the time slot t can be denoted by $\mathbb{V}^t = \{v_1^t, v_2^t, ..., v_j^t, ..., v_M^t\}$. The set of service quality measurements for different satellites for the user at the time slot t can be denoted by $\mathbb{L}^t = \{l_1^t, l_2^t, ..., l_j^t, ..., l_M^t\}$. The state of an agent at the time slot t can be defined as $s^t =< \mathbb{P}^t, \mathbb{V}^t, \mathbb{L}^t >$.

4.3 Actions

The handover decision vector for an agent for the user at the time slot t can be represented by $\mathbb{X}^t = \{x_1^t, x_2^t, ..., x_j^t, ..., x_N^t\}$. The action for the agent is to select a new satellite or keep the user at the current satellite, so the action of an agent for the user at the time slot t can be defined as $a^t = \mathbb{X}^t$.

4.4 Rewards

The reward function that an agent receives at the end of time slot t is due to the action taken at the end of time t and moves from state s^t to another state s^{t+1}.

The reward function for an agent at the time slot t is given below:

$$R^t(s^t, a^t) = \begin{cases} -25, & \text{for } \sum_j a^t v_j^t \leq 1 \text{ or } \sum_j a^t p_j^t \geq 185 \\ 25, & \text{for } a^t = a^{t-1} \\ \sum_j a^t (10 * p_j^t + 10 * v_j^t - 10 * l_j^t), & \text{otherwise} \end{cases}$$

4.5 Policy

The policy is a function that maps states to actions for the agent and is denoted by $pi(s^t)$ for the agent. The final objective is to maximize the expected cumulative reward or return value over K time slots by finding the optimal policy π^* for the agent,

$$\pi^* = \max_{\pi} \mathbb{E}[\sum_{t=1}^{t=K} \gamma^t R^t(s^t, a^t)] \tag{1}$$

4.6 Deep Q-Network (DQN) Model

We consider a DQN model to map the state action values from the given input states using a Fully Connected Neural Network. Then we use this value to find out the optimal policy in a Q-learning framework. We summarize the algorithm below:

Algorithm 1 Deep Q-Network (DQN) Algorithm

Initialize replay memory D with capacity N.
Initialize C as the synchronizing interval between the target and policy network.
Initialize action-value function Q with random weights θ.
Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$.
for each episode **do**
  Initialize state s_1.
  for each time step t **do**
    With probability ϵ select a random action a_t, otherwise select $a_t = \text{argmax}_a Q(s_t, a; \theta)$.
    Execute action a_t in the environment and observe reward r_t and next state s_{t+1}.
    Store transition (s_t, a_t, r_t, s_{t+1}) in D.
    Sample a random minibatch of transitions (s_i, a_i, r_i, s_{i+1}) from D.
    Set target value $y_i = r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a'; \theta^-)$.
    Calculate the Mean Squared Error (MSE) loss between the target value and the value through the Bellman equation.

$$Q_t(s_t, a_t) = \sum_{s_{t+1}, r_t} p(s_{t+1}, r_t | s_t, a_t) \left[r_t + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) \right] \tag{2}$$

    Calculate gradients through backpropagation
    Update weights θ using ADAM optimizer
    if t mod C = 0 **then**
      Update the target weights $\theta^- = \theta$.
    end if
  end for
end for

5 Experimental Setup

In this section, we talk about the experimental setup in detail listing the set of environment, RL framework and DQN model parameters along with their chosen values for the training. In our environment, we consider a single user covered by multiple different satellites at different timestamps. We put the UE at (-62 °, 50 °) geographical coordinates on the ground level. We consider a simulation

time of 5 minutes divided into 30 equal time slots starting on April 29, 2023, 09:30 am UTC time. Within this period, we consider 17 different satellites to provide coverage to the user with a typical height varying from $400 - 600$ km. The UE is considered to be under the coverage of a particular satellite when the elevation angle of that satellite with respect to the UE is greater than or equal to $10°$. In Table 1, we give a summary of the LEO satellite mobility environment parameters:

Table 1: Summary of LEO satellite mobility environment parameters

Parameter	**Value**
UE position (Latitude, Longitude, Altitude)	(-62 °, 50 °, 0m)
Simulation time (minutes)	5
Number of total time slots	30
Number of total satellites providing coverage	17
Typical satellite height (km)	400-600
Minimum coverage elevation angle	10°
Simulation starting time	04-29-2023 09:30 am (UTC)

In the Q-learning framework, we consider the 10 best satellites, each with 3 features, in total 30 distinct states. For the action, we consider a one-hot encoded vector of dimension 10 denoting the connectivity status of these satellites. We consider 500 episodes for training and typical values for other exploration parameters as listed below in the table 2. Then in Table 2, we give a summary of the system model parameters:

Table 2: Summary of RL framework parameters

Parameter	**Value**
Size of state space	30
Size of action space	10
Number of episodes	500
γ	0.6
ϵ_{start}	1.0
ϵ_{end}	0.05
$\epsilon_{decayrate}$	0.95, 0.99, 0.9995

We set the learning rate as 0.001, memory buffer size as 64, and target network synchronization interval 100 episodes. We consider a 4-layered Fully Connected Neural Network with $64, 48$, and 24 neurons respectively for the first, second, and third hidden layer. In each hidden layer, we consider the ReLU activation function. Finally in Table 3, we give a summary of the DQN model parameters:

Table 3: Summary of DQN framework parameters

Parameter	**Value**
Learning rate	0.001
Memory buffer size	64
Target network synchronization interval	100 episodes
Number of Fully Connected Layers	4
Number of neurons in hidden layer 1	64
Number of neurons in hidden layer 2	48
Number of neurons in hidden layer 3	24
Optimizer	Adam
Loss Function	Mean-Squared Error (MSE)

6 Evaluation Results

In this section, we discuss the evaluation results. We use the mean reward function value at each episode as the evaluation metric. We vary different hyperparameters to find out the best suitable set to tackle this problem.

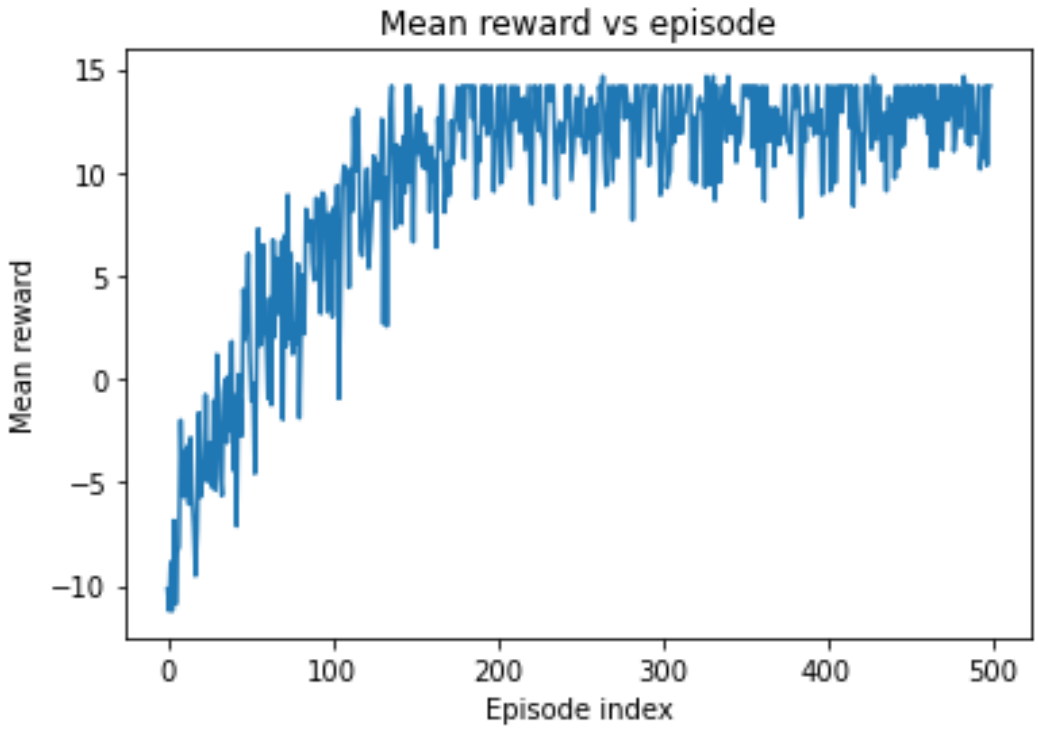
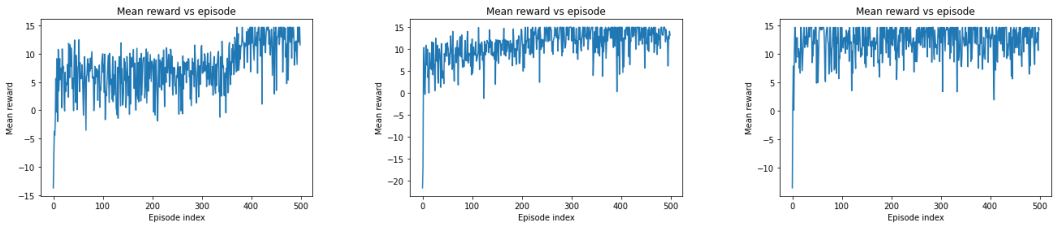


Figure 1: Convergence of mean rewards with episodes.

6.1 Learning Convergence Analysis

In an RL framework, the agent learns how to behave in the environment by learning the optimal policy for mapping the states to the best action at a specific time instant. Learning convergence is achieved when the agent's policy or value function stabilizes and no longer changes significantly over time, indicating that the agent has learned the optimal strategy for the given task. This is one of the fundamental curves to verify the learning process of an RL framework. In Figure 1, which shows the mean reward variation with episode progression, we can see that the mean reward received by the agent gradually increases as the training progresses. After around 200 episodes, we see the mean reward stabilizes which means the learning procedure converges to an optimal (near-optimal more accurately) solution for the specific problem. This verifies with this setup, our learning model converges.

6.2 Effect of Discounting Factor



(a) $\gamma = 0.1$ (b) $\gamma = 0.6$ (c) $\gamma = 0.95$

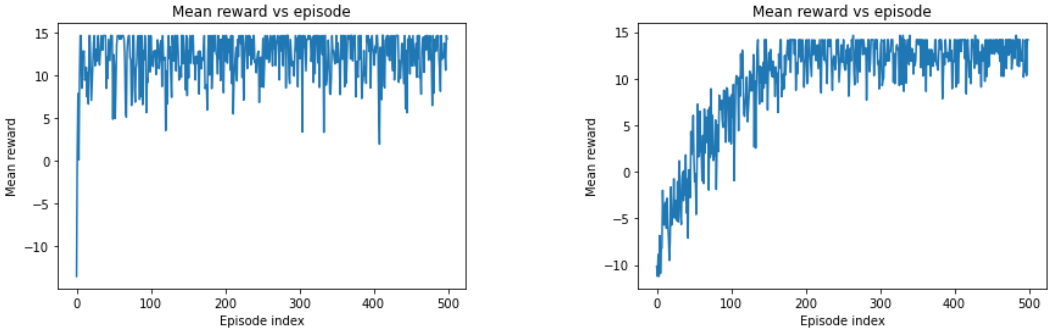
Figure 2: Effect of γ on the learning curve.

We demonstrate the effect of the discounting factor, γ in Figure 2. In Figure 2a, we see $\gamma = 0.1$, so the agent decides its action mostly based on the current reward. So it cannot capture the future cumulative reward. Again, when $\gamma = 0.95$, which is very close to 1, the agent puts a lot more

emphasis on future rewards. So the agent puts excessive importance on future benefits. So the agent goes to the maximum value of mean reward momentarily after the first episode and oscillates a lot (Figure 2c). None of these approaches can provide stable results as expected. On the contrary, when $\gamma = 0.6$, the training hits a balance, and we can see less instability in the mean reward variation as shown in Figure 2b.

6.3 Effect of Exploration Factor Decay Rate

This ϵ decay rate controls the exploration rate of the model. It can change the value of exploration probability (ϵ) rapidly or slowly depending on its value. As exploration can control the learning direction of the agent, it can shape the convergence curve too. This effect is illustrated in Figure 3. In Figure 3a, the value of this parameter is 0.95 whereas in Figure 3b, it is 0.9995. So the second approach can explore the environment more and learns the model steadily with time compared to the first approach.



(a) $\epsilon_{decaryrate} = 0.95$ (b) $\epsilon_{decaryrate} = 0.9995$

Figure 3: Effect of $\epsilon_{decaryrate}$ on the learning curve.

7 Conclusion and discussion

In this project, we test a real problem, handover optimization for LEO satellites using a DQN model considering a single UE with simplified channel model assumptions. Our algorithm shows expected convergence in the mean reward and the effect of different hyperparameters is also analyzed. We see a balanced value of discounting factor and a small exploration factor decay rate can provide us with better reward accumulation. In the future, I want to test it in an environment consisting of multiple users along with more realistic channel assumptions and interference considerations.

References

[1] X. Lin, S. Rommer, S. Euler, E. A. Yavuz, and R. S. Karlsson, "5G from space: An overview of 3GPP non-terrestrial networks," *IEEE Commun. Stand. Mag.*, vol. 5, no. 4, pp. 147–153, 2021.

[2] T. 38.811, "Study on New Radio (NR) to support non-terrestrial networks," *V15.4.0*, October 2020.

[3] M. Chen, Y. Zhang, Y. Teng, B. Liu, and L. Zhang, "Reinforcement learning based signal quality aware handover scheme for LEO satellite communication networks," in *Human Centered Comput.: 5th Int. Conf. , HCC 2019, Čačak, Serbia, August 5–7, 2019, Revised Selected Papers 5*. Springer, 2019, pp. 44–55.

[4] J. Wang, W. Mu, Y. Liu, L. Guo, S. Zhang, and G. Gui, "Deep reinforcement learning-based satellite handover scheme for satellite communications," in *2021 13th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2021, pp. 1–6.

[5] H. Xu, D. Li, M. Liu, G. Han, W. Huang, and C. Xu, "QoE-driven intelligent handover for user-centric mobile satellite networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 127–10 139, 2020.

[6] S. He, T. Wang, and S. Wang, "Load-Aware Satellite Handover Strategy Based on Multi-Agent Reinforcement Learning," in *GLOBECOM 2020 - 2020 IEEE Global Commun. Conf.*, 2020, pp. 1–6.

[7] C. Zhang, N. Zhang, W. Cao, K. Tian, and Z. Yang, "An AI-Based Optimization of Handover Strategy in Non-Terrestrial Networks," in *2020 ITU Kaleidoscope: Industry-Driven Digit. Transformation (ITU K)*, 2020, pp. 1–6.

[8] S. Jung, M.-S. Lee, J. Kim, M.-Y. Yun, J. Kim, and J.-H. Kim, "Trustworthy handover in LEO satellite mobile networks," *ICT Express*, vol. 8, no. 3, pp. 432–437, 2022.

[9] D.-F. Wu, C. Huang, Y. Yin, S. Huang, M. W. A. Ashraf, Q. Guo, and L. Zhang, "LB-DDQN for handover decision in satellite-terrestrial integrated networks," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–11, 2021.

[10] H. Liu, Y. Wang, and Y. Wang, "A successive deep Q-learning based distributed handover scheme for large-scale LEO satellite networks," in *2022 IEEE 95th Veh. Tech. Conf. :(VTC2022-Spring)*. IEEE, 2022, pp. 1–6.