remote sensing

MDPI

Article

LEARNING-BASED TRAFFIC SCHEDULING IN NON-STATIONARY MULTIPATH 5G NON-TERRESTRIAL NETWORKS

Achilles Machumilane [1,3,*], Alberto Gotta [1,2], Pietro Cassará [1,2], Giuseppe Amato [1] and Claudio Gennaro [1]

[1] Institute of Information Science and Technologies (ISTI), CNR, 56124 Pisa, Italy; alberto.gotta@isti.cnr.it (A.G.); pietro.cassara@isti.cnr.it (P.C.); giuseppe.amato@isti.cnr.it (G.A.); claudio.gennaro@isti.cnr.it (C.G.)
[2] CNIT—National Inter-University Consortium for Telecommunications, 43124 Parma, Italy
[3] Department of Information Engineering, University of Pisa, 56126 Pisa, Italy
* Correspondence: achilles.machumilane@phd.unipi.it

Abstract: In non-terrestrial networks, where low Earth orbit satellites and user equipment move relative to each other, line-of-sight tracking and adapting to channel state variations due to endpoint movements are a major challenge. Therefore, continuous line-of-sight estimation and channel impairment compensation are crucial for user equipment to access a satellite and maintain connectivity. In this paper, we propose a framework based on actor-critic reinforcement learning for traffic scheduling in non-terrestrial networks scenario where the channel state is non-stationary due to the variability of the line of sight, which depends on the current satellite elevation. We deploy the framework as an agent in a multipath routing scheme where the user equipment can access more than one satellite simultaneously to improve link reliability and throughput. We investigate how the agent schedules traffic in multiple satellite links by adopting policies that are evaluated by an actor-critic reinforcement learning approach. The agent continuously trains its model based on variations in satellite elevation angles, handovers, and relative line-of-sight probabilities. We compare the agent's retraining time with the satellite visibility intervals to investigate the effectiveness of the agent's learning rate. We carry out performance analysis while considering the dense urban area of Paris, where high-rise buildings significantly affect the line of sight. The simulation results show how the learning agent selects the scheduling policy when it is connected to a pair of satellites. The results also show that the retraining time of the learning agent is up to 0.1 *times* the satellite visibility time at given elevations, which guarantees efficient use of satellite visibility.

Keywords: non-terrestrial networks; satellites; link prediction; reinforcement learning; actor-critic; multipath

Citation: Machumilane, A.; Gotta, A.; Cassará, P.; Amato, G.; Gennaro, C. Learning-Based Traffic Scheduling in Non-Stationary Multipath 5G Non-Terrestrial Networks. *Remote Sens.* **2023**, *1*, 0. https://doi.org/

Academic Editor:

Received:
Revised:
Accepted:
Published:

1. Introduction

Non-Terrestrial Networks (NTNs), including Low Earth Orbit (LEO) satellite constellations, Unmanned Aerial Systems (UASs), and High Altitude Platforms (HAPs), have been identified as promising technologies to provide ubiquitous connectivity [1] in the future generation Internet. For this reason, the Third-Generation Partnership Project (3GPP) [2] has included NTNs among the supporting technologies for the extension of the terrestrial fifth generation (5G) into the sixth-generation (6G) mobile networks. NTNs can be exploited to meet the requirements of emerging technologies, such as ubiquitous artificial intelligence (AI) and the Industrial IoT (IIoT), for application use cases such as remote monitoring, goods delivery, connected autonomous vehicles (CAVs), and high-speed transportation (e.g., trains or aircraft). However, the main challenge in New Radio NTN integration is the communication between the User Equipment (UE) and the satellite, because it requires the Line-of-Sight (LOS). In dense urban scenarios, high-rise buildings or tall infrastructures

can severely affect LOS communication due to signal blockage and reflection phenomena. Communication in the LOS between satellites and the UE becomes even more challenging in scenarios where the satellite and the UE are moving relative to each other because, in these scenarios, the LOS probability changes with the satellite elevation angle. Therefore, continuous LOS estimation techniques are paramount for the UE to access the satellite and maintain connectivity.

This paper proposes a Reinforcement Learning (RL)-based network function (NF) to self-learn the selection of non-terrestrial links with Multi-Path Routing (MPR) in dense urban scenarios. In an MPR transmission system, an original data stream is split into sub-streams, each of which is transmitted over its own path [3]. This means that the transmission system is characterized by multiple spatially or logically separated paths that are aggregated for transmission [2]. This differs from using only one path at a time while keeping the others as backups in case of link failures. The paths are distinguished by attributes such as the source and destination Internet Protocol (IP) addresses and relative ports. A path is identified by a series of connections from the sender to the receiver. Since this work considers a direct link or connection between the Unmanned Aerial Vehicle (UAV) and the satellite, the terms path and link can be used interchangeably. Transmission over multiple paths can improve channel availability and reliability, data recovery, throughput, path aggregation and load balancing. It can therefore improve the Quality of Service (QoS) and Quality of Experience (QoE). Multipath transmission can support UAVs deployed in satellite networks when there is no LOS [4]. In our scenario, the MPR allows UE with multiple radios to set up multiple satellite links to improve reliability and data rates [3,4] even when the performance of a single link is degraded due to LOS variations. Despite all these advantages of multipath transmission, dynamic path selection and estimating the required replicas for traffic protection are major challenges and require the Channel State Information (CSI). In this work, we assume a non-stationary LOS probability due to the continuous variation in the satellite elevation angle, as provided in [5]. In such scenarios, a reliable LOS estimation model allows the UE to select a link or more links to maximize an objective, such as limiting the End-to-End (E2E) loss while using minimal bandwidth. To this end, we adopt the Actor-Critic (AC) version of RL, which guarantees better performance with continual learning for the non-stationary LOS probability that underlies our system. We analyze the latency of the NF agent in recovering from an abrupt change in the LOS of one or more links. The changes in the LOS are due to the satellite visibility period, which depends on both the satellite's elevation angle and the latitude of the UE.

The main contributions of our work can be summarized as follows:

- We provide a learning-based method for selecting an optimal MPR-based policy according to the time-varying satellite elevation angle. We also provide a mechanism for reliable estimation of the non-stationary LOS probability.
- By including MPR capabilities in our transmission policy, we allow the UE to transmit on multiple satellite links to improve link availability and data rates and minimize end-to-end (E2E) loss.
- The novelty in this work is that we provide a self-learning-based LOS tracking mechanism that can work in scenarios with non-stationary link state transition probabilities, which is a challenge in NTN mobile systems.

In the following part of the paper, we review the literature related to our research and describe our system model and the architecture of the proposed AC agent. We then present and discuss the simulation results and finally conclude the paper by setting the direction for future research.

1.1. Related Work

In this section, we provide various techniques that have been proposed in the literature for traffic scheduling and protection in multipath transmission systems, LOS prediction and tracking and the application of RL in multipath traffic scheduling.

1.1.1. Multipath Traffic Scheduling and Protection

Multipath traffic scheduling means allocating network traffic on multiple paths by selecting the appropriate path(s) for data transmission to meet specific constraints or service requirements. A multipath scheduling technique must take into account both traffic protection and bandwidth preservation. However, most of the scheduling techniques proposed in the literature do not take either into consideration. For example, the conventional round-robin (RR) scheduling strategy sends data sequentially over multiple paths and neglects path conditions. The RR scheduler has been found to perform poorly in the Multipath Transmission Control Protocol (MP-TCP) [6]. Although the Weighted Round Robin (WRR) scheduler is improved compared with the (RR) scheduler, the weights it assigns to paths are usually static, which makes it impractical under time-varying conditions such as those in NTN networks. Similarly, the deficit round-robin (DRR) and weighted fair queuing (WRQ) schedulers [6] do not adapt easily to dynamic channel conditions. Path-Aware Networking (PAN) scheduling strategies consider path conditions such as the Round-Trip Time (RTT), Packet Loss Rate (PLR) and bandwidth [7]. The RTT as used in schedulers such as the round-trip time threshold and the lowest-RTT-first schedulers [6] enables traffic to arrive before the expiration time [8,9], but the *head-of-line blocking* can affect connections that differ greatly in latency. When PLR [9,10] is used for scheduling, undelivered and delayed traffic is taken as lost traffic. In contrast, our proposed learning-based scheduling system can adapt to dynamic link states and make proper path selections and redundancy estimations.

As for traffic protection, several schemes have been proposed, such as Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ). FEC can waste bandwidth because of the fixed redundancy rate that does not take into account the dynamism of the network. ARQ, on the other hand, uses retransmissions to compensate for lost traffic but can cause network congestion [11,12] and adversely affect multimedia quality [13]. Thus, it is not preferred for real-time traffic, especially in satellite transmission, which is characterized by long delays. In such scenarios, FEC could provide a solution, but it introduces a computational load on constrained devices such as UAVs. Our learning-based model, on the other hand, provides traffic protection by using the required redundancy, depending on the dynamic network conditions, while avoiding excessive overhead.

1.1.2. RL-Based Traffic Scheduling

Recently, there has been great interest in applying RL in transmission networks. As a result, various RL-based models have been proposed for traffic control and scheduling. In [14], the authors presented a scheduling framework based on RL for satisfying the bandwidth requirements of Wi-Fi users. In [15], another scheduler using the RL model was proposed for multipath QUIC in Wi-Fi and cellular transmissions. An AC framework was proposed in [16] for dynamic single-user and multi-user access to multiple links in wireless networks which avoids collisions by selecting suitable links. An RL framework was proposed in [17] to improve data rates and mitigate E2E delay in IoT networks . The work in [18] proposed an AC-based scheduling framework for UAV cellular integrated networks.

Inspired by this development, but different from these works, we propose a learning-based framework that performs path selection and traffic protection by repeating traffic over multiple links to provide redundancy. Using an AC-based algorithm, our agent searches for a policy to select suitable satellite links in terms of LOS availability and determine how much redundancy is required to protect traffic against channel losses due to a varying LOS probability, which depends on the changing satellite elevation angle. Redundancy estimation is carried out in such a way as to provide enough protection without wasting bandwidth. Moreover, the AC algorithm used by our agent is a model-free RL algorithm that does not require knowing in advance the model's underlying transmission channels.

1.1.3. LOS Prediction and Tracking

Various LOS prediction and tracking methods have also been proposed. For example, in [19], the authors proposed a theoretical model that estimates the probability of cloud-free LOS (CFLOS) for satellite links based on the satellite elevation and the altitude of the ground station. Sun et al. [20] proposed a method for detecting Non-Line-of-Sight (NLOS) using data from the Global Navigation Satellite System. In [21], the authors proposed an empirical model to estimate the LOS probability for satellite and HAP communications. In contrast to these physical and empirical methods, we propose an RL-based model for non-stationary scenarios in which LEO satellites move continuously, causing their elevation angles and consequently the LOS probability to change. Moreover, our model allows UE to estimate the traffic scheduling policy for multiple satellites, allowing for multiple parallel transmissions.

2. Materials and Methods

2.1. Channel Model

Figure 1 shows the reference scenario considered in this paper. We studied LOS estimation and link selection in the presence of dual connectivity in NTNs with simultaneous use of two radios, as envisioned by the 3GPP [2]. In this architecture, the LEO satellites are equipped with the gNB Distributed Unit (DU) [22], while the Centralized Unit (CU) is located on the ground. We considered a scenario in which the satellites and a UAV equipped with two pieces of UEs were moving relative to each other. We made use of the StarLink satellite system with a mass constellation of 3000 LEO satellites. The UE could connect to two satellites simultaneously. To allow our framework to overcome energy constraints, our AC agent ran on the Ground Control Station (GCS) of the UAV which had enough computational resources. After computations, the GCS sent the traffic scheduling policy to the energy-constrained UE (UAV), which only performed traffic scheduling over the satellite links. (see Figure 1).

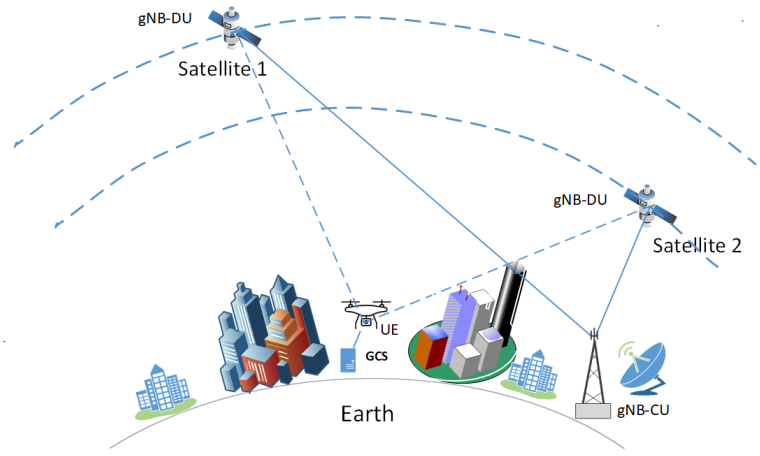

Figure 1. Reference scenario: a UE (UAV) accessing two satellites in an NTN in a dense urban environment.

According to [23], a satellite in the constellation moves in a circular orbit with inclination ι at an altitude h and an orbit radius $r_S = r_E + h$, and the satellites move independently of each other. The same authors in [23] defined

$$\gamma(\theta) = cos^{-1}((r_E/r_S) \cdot cos(\theta) - \theta) \quad (1)$$

as the central angle between the Earth station (the UE in this case) and the locus of the trajectory points of the satellite corresponding to an elevation angle θ, with $\theta_{min} \leq \theta \leq \theta_{max}$. For any single point of the satellite's locus, the maximum elevation angle θ_{max} determines the visibility time of the satellite and the distribution of the elevation angles in the visibility

region [23]. The visibility region is defined as the smallest angle $\gamma(\theta_{max})$ for which the satellite is visible from the UE along its whole trajectory. Therefore, given the UE latitude ϕ_0, the probability for a satellite in its trajectory to be visible from the UE can be determined from the Probability Density Function (PDF) of θ_{max}, denoted by

$$f_{\Theta_{max}}(\theta_{max}) = \frac{G(\theta_{max})}{K} \cdot \cdot \left(\frac{cos(\phi_0 + \gamma(\theta_{max}))}{\pi\sqrt{sin^2(\iota) - sin^2(\phi_0 + \gamma(\theta_{max}))}} + \frac{cos(\phi_0 - \gamma(\theta_{max}))}{\pi\sqrt{sin^2(\iota) - sin^2(\phi_0 - \gamma(\theta_{max}))}} \right) \quad (2)$$

where $\theta_{min} \leq \theta_{max} \leq \frac{\pi}{2}$ and

$$G(\theta_{max}) = \frac{1 + (r_E/r_S)^2 - 2(r_E/r_S)cos(\gamma(\theta_{max}))}{1 - (r_E/r_S)cos(\gamma(\theta_{max}))}$$

$$K = \frac{1}{\pi} sin^{-1}\left(\frac{sin(\phi_0 + \gamma(\theta_{min}))}{sin(\iota)} \right) sin^{-1}\left(\frac{sin(\phi_0 - \gamma(\theta_{min}))}{sin(\iota)} \right) \quad (3)$$

The PDF in Equation (2) may assume different shapes according to ι, ϕ_0 and $\gamma(\theta_{max})$, as detailed in [23]. For space limitations, we will only account for the PDF of elevation angles considering the points of the satellite's trajectory in the visibility region. The authors in [23] derived the PDF $f_\Theta(\theta)$ as the marginalization of the joint probability $f_{\Theta,\Theta_{max}}(\theta, \theta_{max})$, defined as in the following equation:

$$f_\Theta(\theta) = \int_\theta^{\theta_{max}} f_{\Theta,\Theta_{max}}(\theta, \theta_{max}) d\theta_{max} \quad (4)$$

where $\theta_{min} \leq \theta \leq \theta_{max}$ and

$$f_{\Theta,\Theta_{max}}(\theta, \theta_{max}) = \frac{G(\theta)sin(\gamma(\theta))}{\sqrt{cos^2(\gamma(\theta_{max})) - cos^2(\gamma(\theta))}} \cdot \cdot \frac{f_{\Theta_{max}}(\theta_{max})}{\int_{\theta_{min}}^{\theta_{max}} f_{\Theta_{max}}(x) \cdot cos^{-1}\left(\frac{cos(\gamma(\theta_{min}))}{cos(\gamma(x))} \right) dx}.$$

Therefore, the satellite visibility interval from UE at a given latitude as the elevation angle varies from θ_i to θ_j was given in [23] as

$$T_{\theta_i,\theta_j} = \int_{\theta_i}^{\theta_j} \frac{2}{\omega_S - \omega_E cos(\iota)} cos^{-1}\left(\frac{cos(\gamma(\theta_i))}{cos(\gamma(x))} \right) f_{\Theta_j}(x) dx. \quad (5)$$

The satellites move in different orbits at different speeds. According to 3GPP [4], the LOS probability changes with the changing satellite elevation angle. In general, the LOS probability increases with the elevation, reaching a maximum at *Nadir* (90°) when the satellite is above the UE if it is in the orbital plane of the satellite. In dense urban areas, the LOS probability is lower, especially at low altitudes, because the signal is obstructed by and reflected off of high-rise buildings. Consequently, the AC agent must learn whether to schedule traffic transmission on any one link or on both links simultaneously (for redundancy), according to a given QoS requirement and according to the estimate of the LOS probability model of the two links as the satellites change their elevation angles.

In this work, for the channel model, we adopted the statistical model for mixed propagation conditions provided by the International Telecommunication Union (ITU) for the design of Earth-space land mobile telecommunications systems [5]. In this ITU recommendation, a communication channel between a satellite and a UAV or any land mobile terminal is characterized by variations in the received signal power due to shadowing from buildings and vegetation, as well as multipath fading as a result of reflections from

obstacles and from the ground. The ITU recommendations [5] provide a three-state Markov chain model to characterize the behavior of the land mobile satellite channel: (1) The first state is characterized by the presence of the LOS. This state is modeled by a Ricean fading for unshadowed areas with high received signal power. (2) The second is the state with no LOS due to strong shadowing and blocking from obstacles. This state is modeled with Rayleigh fading. (3) Between these two states, there is a third state known as the transition state, in which the multipath component power increases or decreases linearly [5]. For the purpose of this work, we followed a more simplified Markov chain model known as the Lutz model [24,25], which approximates the three states into two states: The first is the lossless good state (G) with LOS, and the second is the bad state (B) with no LOS which is characterized by shadowing, blocking and erroneous traffic reception.

We now derive the state transition probabilities according to the Lutz model. Following the work in [26], we let

$$\tau_s = \begin{pmatrix} 1-b & b \\ g & 1-g \end{pmatrix}$$

be the switching matrix of the Lutz model. In this model, the time required to transmit a bit is taken as the channel state transition unit, and b and g are the transition probabilities from G to B and from B to G, respectively, with G denoting the good state and B denoting the bad state.

If we let L_g and L_b denote the mean length (in meters) of the G and B states, respectively, as derived in [24], and let ν be the speed in meters per second (m/s) of a moving vehicle, and we then assume that the packets transmitted by the vehicle have a length of l bits, with R being the bit rate in bits per second (bps), then the state time durations D_b and D_g are given by $D_g = 1/b$ and $D_b = 1/g$, respectively, equal to

$$D_b = \frac{R}{\nu \cdot l} L_g; \; D_g = \frac{R}{\nu \cdot l} L_b. \tag{6}$$

Then, it follows that

$$b = (\frac{R}{\nu \cdot l} L_g)^{-1}; \; g = (\frac{R}{\nu \cdot l} L_b)^{-1} \tag{7}$$

According to [5], L_b and L_g can be computed as follows:

$$L_{G,B} = exp\left(\mu_{G,B} + \frac{\sigma^2_{G,B}}{2}\right) \frac{erfc\left(\frac{logdur_{min,G,B} - (\mu_{G,B} + \sigma^2_{G,B})}{\sigma\sqrt{2}}\right)}{erfc\left(\frac{logdur_{min,G,B} - \mu_{G,B}}{\sigma\sqrt{2}}\right)} \tag{8}$$

where $(\mu, \sigma)_{G,B}$ and dur_{min} are the mean, standard deviation and minimum state lengths in meters, respectively, of the channel states.

These parameters ($(\mu, \sigma)_{G,B}$ and dur_{min}) are provided in [5] for urban, suburban and rural environments at different elevation angles and transmission frequencies as reported in Table 1.

Table 1. Satellite link parameters for the dense urban area in France at 2.2 GHz [5].

Elevation	$\mu_{G,B}$	$\sigma_{G,B}$	$dur_{minG,B}$
20°	2.0042, 3.6890	1.2049, 0.9796	3.9889, 10.3114
30°	2.7332, 2.7582	1.1030, 1.2210	7.3174, 5.7276
45°	3.0639, 2.9108	1.6980, 1.2602	10.0, 6.0
60°	2.8135, 2.0211	1.9595, 0.6568	10.0, 1.9126
70°	4.2919, 2.1012	2.4703, 1.0341	118.3312, 4.8569

For the purpose of this work, we used the parameters for an urban environment at 2.2 GHz to compute the mean lengths of the channel states L_g and L_b using Equation (7). We then used Equation (6) to calculate the corresponding transition probabilities g and b for our channel model as reported in Table 2. We then used these transition matrices to create Markov link state traces for training our model. We assumed successful traffic reception only if there was an LOS (i.e, it was in a good state). We also assumed that the UAV received some feedback reports as described in [3], indicating the traffic reception status and the link state.

Table 2. Satellite link state transition probabilities for the dense urban area in France at 2.2 GHz.

Elevation	$P(B \rightarrow G)$ (g)	$P(G \rightarrow B)$ (b)
20°	0.00014310	0.00047466
30°	0.00024460	0.00027570
45°	0.00020318	0.00007556
60°	0.00105161	0.00010797
70°	0.00052923	2.76683×10^{-6}

2.2. Problem Formulation

LOS estimation on multiple links can be formulated as a Markov decision process (MDP). Specifically, it is modeled as a Partially Observable Markov Decision Process (POMDP) [27] because, at any observation time, the RL agent can only observe the link(s) it has selected. A POMDP is defined by the tuple $\{\mathcal{S}, \mathcal{A}, P(s_{t+\Delta t}|s_t, a_t), r_t\}$, where $\mathcal{S}$ is the state space of the system and $\mathcal{A}$ denotes the action space for achieving the optimal choice. $P(s_{t+\Delta t}|s_t, a_t)$ is the probability of being in state $s_{t+\Delta t} \in \mathcal{S}$ after a time interval Δt conditioned by the action $a_t \in \mathcal{A}$ and the state $s_t \in \mathcal{S}$, and r_t is the immediate reward for the action a_t that leads to the state transition from s_t to $s_{t+\Delta t}$. In the following, we describe the POMDP for our problem, where we assume that the UE can select a subset of the N available satellite links to which it is connected.

1. *State space*: We assumed the state of each selected channel to be a binary variable with values in $\{\text{LOS}, \text{NLOS}\}$ so that we could formally define the state of the link $n = 1 \dots N$ at time t as follows:

$$s_{nt} = \begin{cases} +1 & \text{if } s_{nt} = \text{LOS} \\ -1 & \text{otherwise.} \end{cases}$$

   Each of the N links dynamically changes its state between LOS and NLOS according to its own transition matrix T_n as defined in [28]. In our use case, we assumed that the number of available links was $N = 2$ so that we could define the link state space as the set of vectors $\mathcal{S} = \{\mathbf{s}_t \mid \mathbf{s}_t = [s_{1t}, \dots, s_{Nt}]\}$, which generates a state space $\mathcal{S} = \{[LOS, LOS], [LOS, NLOS], [NLOS, LOS], [NLOS, NLOS]\}$.
2. *Actions*: An action constitutes the choice of the appropriate transmission pattern (i.e., a subset of the N links). The action space is the set of vectors $\mathcal{A} = \{\mathbf{a}_t \mid \mathbf{a}_t = [\rho_{1t}, \dots, \rho_{Nt}]\}$, where $\rho_{nt} = 1$ indicates that the nth link is selected and it is $\rho_{nt} = 0$ otherwise for $n = 1 \dots N$. In this case study, we assumed that we had a pair of radio interfaces (i.e., $N = 2$), which led to having an action space $\mathcal{A} = \{[0, 1], [1, 0], [1, 1]\}$.
3. *Reward*: The immediate reward r_t is defined as a penalty to the agent and is proportional to the E2E loss rate above the threshold calculated over an episode, as in the following equation:

$$r_t = \begin{cases} \frac{-(\psi - \varphi)}{\eta} & \text{if } \psi >= \varphi \\ \frac{1}{\eta} & \text{otherwise.} \end{cases} \quad (9)$$

where ψ is the E2E loss, φ is the E2E loss threshold and η is the number of transmission links used in the previous episode. The first term in the equation encourages the use of double transmissions when there are losses, while the second term conserves bandwidth in favorable link conditions.

2.3. Actor-Critic RL Architecture

In this part, we describe the architecture of the actor-critic algorithm. As shown in Figure 2, the architecture of our AC learning agent model consists of three networks: the actor $\pi_{\theta_a}(\mathbf{s}_t)$, the critic network $Q_{\theta_c}(\mathbf{s}_t, \mathbf{a}_t)$ and the target critic network $Q_{\theta_{tc}}(\mathbf{s}_t, \mathbf{a}_t)$, parameterized by θ_a, θ_c and θ_{tc} respectively:

1. Actor: The actor explores a policy π that maps the agent's observation of the state to the action space $\mathcal{A}$: using the mapping policy which is the function of the state: $\pi_{\theta_a}(\mathbf{s}_t) \quad : \quad S \rightarrow \mathcal{A}$. To explore the optimal policy π^*, the actor selects actions $\mathbf{a}_t$ from the action space $\mathcal{A}$ and optimizes its selection policy in order to maximize the long-term rewards. The selected action is given by

$$\mathbf{a}_t \sim \pi_{\theta_a}(\mathbf{s}_t) \tag{10}$$

The agent's optimization goal of the long-term rewards can be represented by

$$\pi^*(\mathbf{a}_t|\mathbf{s}_t) = \arg\max_{\mathbf{a}_t} E\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big] \tag{11}$$

where γ is the discounting factor.

2. Critic: The critic is used to estimate the state-action value $Q_{\theta_c}(\mathbf{s}_t, \mathbf{a}_t)$, which gives the goodness of the action selected by the actor at time t and state $\mathbf{s}_t$ and is used to optimize the agent's selection policy in the direction of maximizing future rewards.
3. Target-Critic: To overcome the instability problem of the critic due to frequent updates, we use a third network, the target-critic network, to perform the Bellman's estimation of the future state-action values. When the action $\mathbf{a}_t$ is taken at time t and executed by the agent by transmitting the traffic over the network, the received feedback from the environment is sent to the target critic to estimate the future state-action value. The feedback includes the instant reward r_t and the next state of the environment (transmission paths). The future state-action values are estimated as follows:

$$Q_{\theta_{tc}}(\mathbf{s}_{t+\Delta t}, \mathbf{a}_{t+\Delta t}) = r_t + \gamma Q_{\theta_c}(\mathbf{s}_{t+\Delta t}, \mathbf{a}_{t+\Delta t}) \tag{12}$$

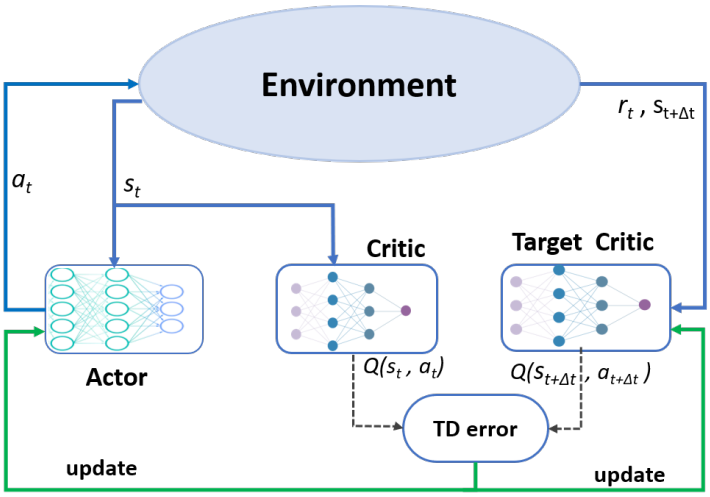

Figure 2. Architecture of the actor-critic learning agent.

Updating the Networks

The estimates of the critic and target-critic networks were used to compute the Time Difference (TD) error [29]:

$$\delta_t = r_t + \gamma Q_{\theta_c}(\mathbf{s}_{t+\Delta t}, \mathbf{a}_{t+\Delta t}) - Q_{\theta_c}(\mathbf{s}_t, \mathbf{a}_t) \tag{13}$$

Then, the critic was updated as given below:

$$Q^*_{\theta_c} = \arg\min_{Q_{\theta_c}} (\delta)^2. \tag{14}$$

The actor was updated by using the policy gradient. The policy gradient was computed using the TD error as follows:

$$\nabla\theta_a J(\theta_a) = E_{\pi_{\theta_a}}[\nabla_{\theta_{at}} \ln \pi_{\theta_{at}}(\mathbf{s}_t, \mathbf{a}_t)\delta_t] \tag{15}$$

where $\nabla\theta_a J(\theta_a)$ is the policy gradient and $J(\theta_a)$ is the policy objective function.

The actor was then updated using the gradient descent method as follows:

$$\theta_a = \theta_a + \beta \nabla_{\theta_a} \ln \pi_{\theta_a}(\mathbf{s}_t, \mathbf{a}_t)\delta_t \tag{16}$$

where β is the learning rate of the actor. Finally, we updated the target critic network by using a soft update method:

$$\theta_{tc} = \alpha\, \theta_{tc} + (1 - \alpha)\theta_c. \tag{17}$$

We designed the actor and critic networks using the TensorFlow-2 and Keras libraries with an ADAM optimizer in a fully connected multilayer perceptron Neural Network (NN) with the parameters given in Table 3.

Table 3. Simulation parameters.

Name	Value
Number of hidden layers	3
Number of neurons for hidden layer	64
Discount factor (ξ)	0.96
Learning rate for the actor (β)	0.001
Learning rate for the critic (α)	0.005
Optimizer	ADAM
UAV velocity (v)	10 m/s
Packet length (l)	1000 bits

2.4. The Learning Process of the AC Agent

At the beginning of each transmission window, the agent using the actor network will select the path (s) to use for transmission according to the observed channel state. It will then transmit the traffic over the selected paths. After several rounds of transmissions, it will receive feedback from the receiver on the loss rate over that window and the path states determined by the reception status. The reward is then computed and sent to the target-critic network to estimate the future state-action values. The critic network estimates the action-state value. The TD error is then computed as the difference between the critic and the target-critic estimates. Then, the actor and critic networks are updated using the TD error. Finally, after several transmission windows, the target-critic network is updated, copying the weights of the critic network. The full learning procedure is detailed in Algorithm 1.

Algorithm 1 The training procedure for the AC-DRL agent for traffic scheduling.

Set to T the total length of the video to be transmitted, the number of iterations per episode to m, and the learning rates for the actor and the critic networks to β and α, respectively. Set the initial state to $\mathbf{s}_0$, the target-critic network update interval to $n = km$ with k positive integer, and the counter for the target-critic update $j = 0$. Finally, set the parameters of the actor network, the critic network, and the target-critic network to θ_a^0, θ_c^0 and θ_{tc}^0, respectively.
while $t \leq T$ **do**
    - Select the action $\mathbf{a}_t \sim \pi_{\theta_a}(\mathbf{s}_t)$ according to the available policy;
    - Set $i = 0$;
    while $i \leq m$ **do**
    - Transmit the video traffic in bits in each iteration by using the selected action $\mathbf{a}_t$;
        if $i = m - 1$ **then**
    - Receive feedback from the receiver on the loss rate and the future
        states of the paths determined by the reception status in the last iteration;
    - Compute the reward r_t according to (9);
    - Compute the state-action value $Q_{\theta_c}(\mathbf{s}_t, \mathbf{a}_t)$;
    - Calculate the future state-action values according to Equation (12);
    - Calculate the TD error according to Equation (13);
    - Update the critic parameters by minimizing: δ_t^2;
    - Update the actor policy according to Equation (16);
    - Update the agent's state observation: $\mathbf{s}_t$;
        end if
    - Set $i = i + 1$;
    - Set $j = j + 1$;
    end while
    if $j = n$ **then**
    - Update the target-critic network according to Equation (18);
    - Set $j = 0$;
    end if
end while

For the purpose of comparison and performance evaluation of the RL method, an optimal traffic allocation policy was defined. This was considered optimal because we assumed that the channel state model was known in advance. Thus, the steady state probabilities were known accurately in each context and for all available paths.

2.5. Simulation Set-up

We simulated UAV satellite transmission with dual connectivity, in which one UAV could use two pieces of UEs to connect to two different satellites. Our goal was to train an AC learning agent to estimate the LOS model of the two satellites and the optimal policy for selecting appropriate links (transmission policy) while tracking the changes in the elevation angles of the satellites. As pointed out earlier, the LOS probability changes according to the elevation angle. Therefore, the learning agent must continuously track the variation of the LOS of the two satellites as a function of the elevation angles. To this end, using a satellite tracker (https://satellitemap.space, accessed on 11 January 2023), pairs of Starlink satellites visible from Paris, France at a given time were selected. A satellite pair was selected that provided clear handoff events that forced the learning agent to retrain its model. This means that as the elevation angle of one satellite decreased and, consequently, the LOS probability decreased, the corresponding UAV interface connected to a new visible satellite with a higher elevation angle. Note, however, that the handoffs of the two interfaces did not happen simultaneously, since the two radio interfaces were independent of each other.

Since the new connections had channel models different from the previous ones, the learning agent was forced to retrain to adapt to the new environment. The selected pairs of angles and the two handoff events are shown in Figure 3.

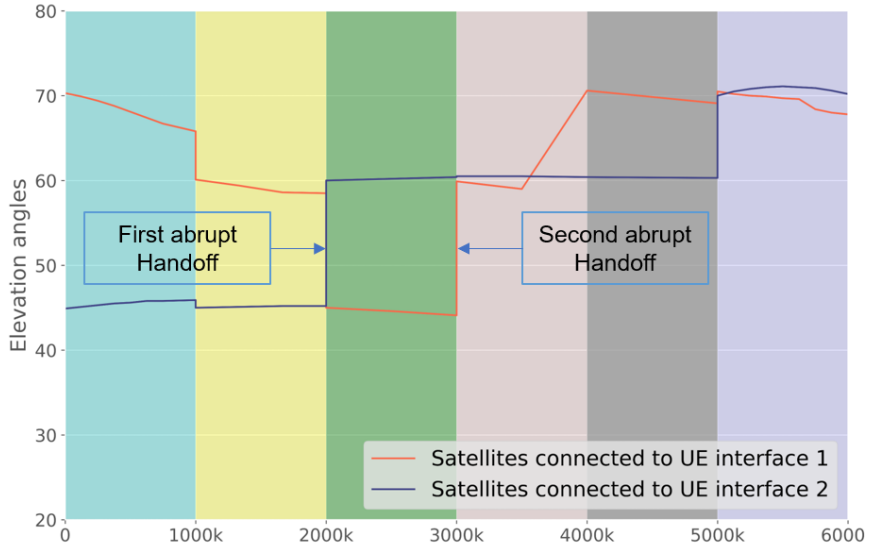

Figure 3. Elevation angles of the pair of satellites connected to the UE in different contexts.

Figure 4 shows the probability mass function of the satellite visibility at given elevation angles, evaluated using Equation (4), compared with the empirical values achieved from the real dataset collected by the satellite tracker during a window of 15 min (the maximum allowed).

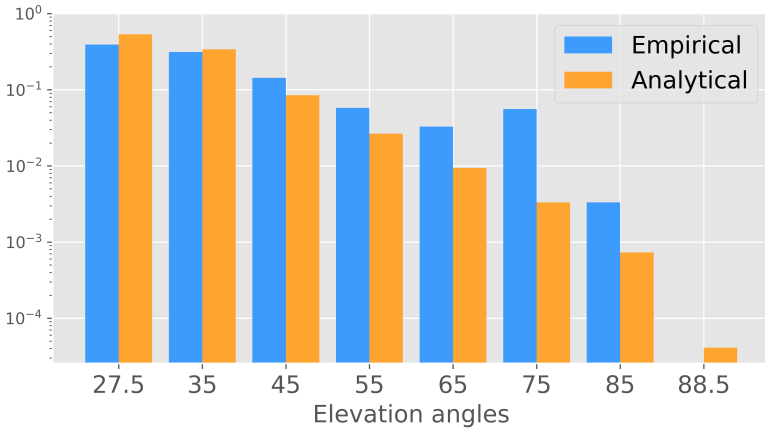

Figure 4. Probability that a satellite is visible from UE at given satellite elevation angles in Paris.

The ITU recommendations in [5] provide the link parameters required to compute the state transition probabilities at different elevation angles, propagation frequencies and environments. These parameters are reported in Table 1 for each state: the good state (G) and bad state (B). These include the mean (μ), variance (σ) and minimum duration of each state in the given propagation environment. For the purpose of this work, we used the parameters for the dense urban scenario at 2.2 GHz and elevation angles of 45°, 60° and 70°, at which the satellites were visible from Paris, France, which was our reference scenario. By applying these parameters in Equations (6–8), we computed the corresponding state transition probabilities, as reported in Table 2. Other parameters used for the computation included the velocity of the mobile UE (v = 10 m/s) and the packet length (l = 1000 bits).

Using the selected pairs of angles, shown in Table 4, and the computed transition probabilities, we constructed state transition matrices for each satellite and for each pair of elevation angles, obtaining a total of six transitions or *contexts* (i.e., in each context or range of elevation angles, we transitioned to a different channel model). The duration of the context was approximated to the minimum satellite visibility time. It is important to

note that we did not draw any assumptions about the physical layer schemes or channel loss models because it was not mandatory for the training of the AC agent, according to the objective function that was designed. We then trained our AC agent using the channel traces of LOS and NLOS created above. The simulated AC networks consisted of 3 fully connected layers with 64 neurons on each layer. The output layer of the actor had a softmax activation function because it had to give the probabilites of selecting each channel, while the output layer of the critic networks had no activation functions because they gave only a single value: the state-action value. The other simulation parameters are reported in Table 3. We ran the simulation for 6 million iterations, with 1 million iterations for each context. On each iteration or transmission event, we considered it to have good reception only if the reported channel state was good or in the LOS. The E2E loss was computed after an episode of 1000 iterations, and the results are reported in the following sections.

Table 4. Satellite elevation angles in each context.

Context	Satellite 1	Satellite 2
1	70°	45°
2	60°	45°
3	45°	60° (handoff)
4	60°(handoff)	60°
5	70°	60°
6	70°	70°

3. Results

We now provide the simulation results in terms of the following aspects: learning rate, path selection, E2E loss rate and bandwidth utilization.

3.1. The AC Agent Learning Performance

Figure 5 shows the total discounted rewards obtained by the agent in a set of contexts represented by different colors. In this figure, we present the median and the 25th and 75th percentiles of rewards. The smooth semi-plateaus within the colored stripes show the steady states that the model achieved at convergence within each context.

Figure 5. Total discounted rewards achieved at different elevation angles.

3.2. Path Selection

Figure 6 shows the categorical distribution achieved by the RL agent for the transmission with satellite 1 (sat_1), satellite 2 (sat_2) and both satellites ($sat_{1,2}$) in the different contexts, corresponding to the probabilities $P(1)$, $P(2)$ and $P(1,2)$, respectively. These are

the path selection probabilities achieved by both the AC agent and the optimal policy after convergence.

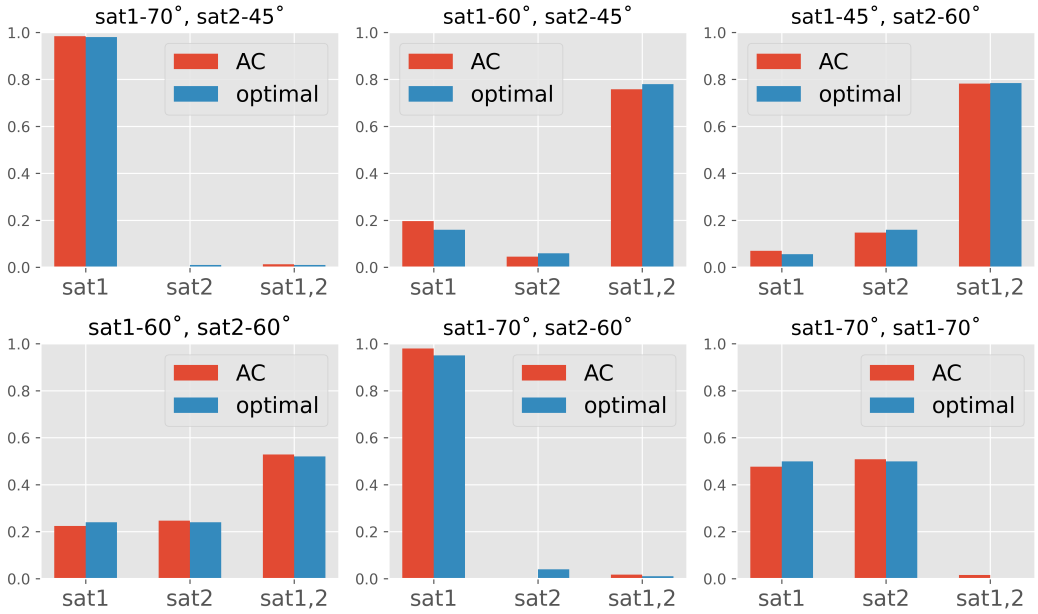

Figure 6. Categorical distribution for multi-link scheduling with AC vs. optimal scheduling at different elevation angle **PLRs!** (**PLRs!**) (good = 0, bad = 1).

3.3. E2E Loss Rate

Figure 7 reports the E2E loss rate achieved by our AC agent, the optimal policy and a round-robin scheduling scheme that does not use redundancy and transmits using only a single path in a round-robin fashion without considering the channel conditions of each path. In our simulations, the E2E loss event κ occurred when all the transmitted bit(s) were lost; that is, they could not be recovered on any of the paths. The E2E loss rate in terms of the Bit Error Rate (BER) was calculated as follows:

$$\psi = \frac{\sum_{i=1}^{m} \kappa}{\sum_{i=1}^{m} \eta} \tag{18}$$

where ψ is the E2E loss rate in terms of the BER, κ is the loss event at a given iteration ($\kappa = 1$ if all sent bits are lost and $\kappa = 0$ otherwise), m is the number of iterations in a window (episode) and η is the number of bits transmitted in each iteration, which is equal to the number of links used in each iteration because each bit was transmitted over a unique link. The numerical values of the average E2E loss rates are reported in Table 5.

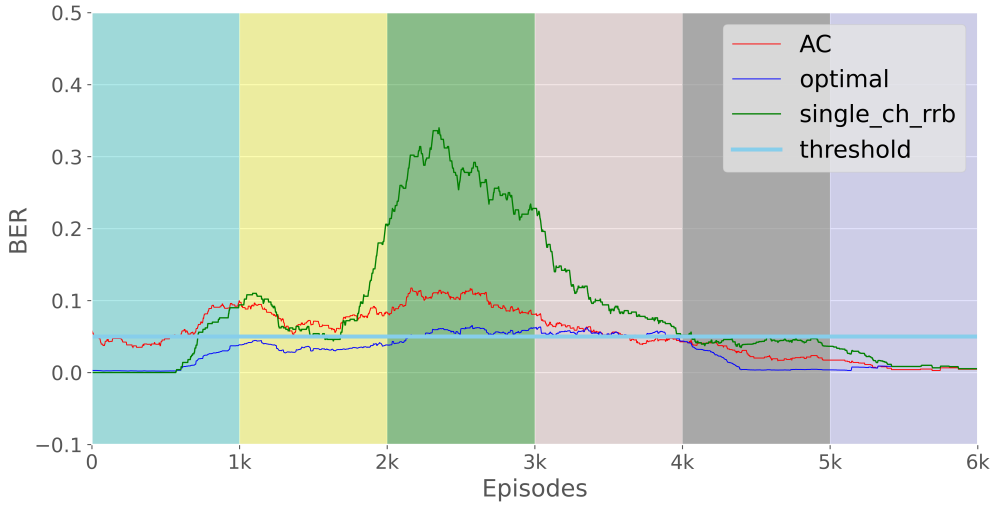

Figure 7. E2E Loss rates.

Table 5. E2E loss rate and bandwidth utilization.

Model	E2E Loss Rate (BER)	Bandwidth (Mbps)
Actor-critic	5.12%	2.1
Optimal policy	4%	2.1
Round-robin	8%	1.5

3.4. Bandwidth Utilization

Figure 8 shows the bandwidth used by our AC agent and the optimal policy in different contexts. On average, both had a repetition factor of 1.4, corresponding to a 2.1 Mbps average throughput. In our simulation, we used 1.5 Mbps as the source rate. On the other hand, the RR system consumed the least bandwidth, with an average bandwidth of 1.5 Mbps (omitted in the figure because it always transmitted with one bit without repetitions). The numerical results are presented in Table 5.

Figure 8. Bandwidth utilization.

4. Discussion

We now discuss in detail the simulation results presented above and provide a proof of concept for how our AC agent works and how it can handle a non-stationary channel model by retraining its model either after moderate changes in the elevation angle or after abrupt

changes (e.g., due to handoffs). We do not wish to elaborate on how the satellite sequences were selected to perform the handoffs. This could instead be the subject of future studies to determine an optimal strategy for the handoffs that moderates the abrupt changes and shortens the learning periods. We evaluated the performance of our AC agent in terms of the following aspects: retraining performance, path selection, E2E loss reduction and bandwidth utilization. We also compared the performance of our agent with the optimal policy and an RR system that transmitted with only a single path in a round-robin fashion with no regard to the path conditions.

4.1. Learning Performance

Figure 5 shows the total discounted rewards obtained by the agent in a set of contexts represented by different colors. As explained above, in each context, we had different pairs of satellite elevation angles and, consequently, different channel models characterized by different channel state transition probabilities. As a result, in each context, the agent had to retrain to track the change in the channel model, as shown by the different discounted rewards that the agent obtained in each context. In the figure, the smooth semi-plateaus within the colored stripes show the steady states the model reached within the context. In contexts 2, 3 and 4, the rewards were lower in relation to the other contexts because the satellites were at low elevation angles, forcing the agent to struggle when compensating for the loss due to low LOS probabilities and high loss probabilities. Note that the reward was proportional to the lost traffic in each episode. On the other hand, in contexts 5 and 6, the satellite elevation angles were relatively high. Thus, the channel conditions were favorable, and the agent could easily reach convergence with high rewards. These results show that our AC agent can dynamically detect the change in a satellite's elevation angle that triggers the change in LOS probability and schedule traffic appropriately to overcome the loss. The relative motions of each satellite are shown in Figure 3, which similar to Figure 5 shows the sequence of contexts, the relative elevation angles and two abrupt handoff events. Figure 5 also shows the average duration it took the agent to retrain the NN model after a change in elevation angle, juxtaposed with the relative context duration. As mentioned earlier, this is of the utmost importance to optimally utilize the satellite visibility time. It was found that the RL agent required on average 2000 iterations to update the NN parameters and reach a local steady state, which is equivalent to $0.1\times$ the satellite visibility time at a given elevation angle, guaranteeing efficient use of satellite visibility.

4.2. Path Selection Performance

Figure 6 shows the categorical distribution achieved by the RL agent at convergence in each context. These are the probabilities $P(1)$, $P(2)$ and $P(1,2)$ for transmitting with satellite 1 (sat_1), satellite 2 (sat_2) and both satellites ($sat_{1,2}$), respectively. According to [**?**] and [28], the higher the elevation angle, the higher the LOS probability, since there are fewer obstacles such as buildings at high elevation angles than at low elevation angles. It can be seen that our agent can recognize this pattern in each context and transmit more on a link with a higher elevation angle (i.e., with a higher LOS probability). In the second context (60° and 45°), the agent transmitted more on satellite link 1, which was at 60°, than on satellite link 2, which was at a lower elevation of 45°. In the scenario where both satellites had the same elevation angle, as in contexts 4 (60° and 60°) and 7 (70° and 70°), the agent distributed traffic equally between the two links. Another observation is that the agent prefers double transmissions when both satellites are at lower altitudes (i.e., it uses replicas to compensate for the loss due to low LOS probabilities at low elevations). This was evident in context 2 (60° and 45°) and context 3 (45° and 60°). On the other hand, when all links were favorable, as in context 6 (70° and 70°), the model transmitted with single links (without replicas) to save bandwidth. The same happened when the two satellites differed significantly in their elevation angles, as in context 1 and context 5. Figure 6 also shows the comparison between the categorical distributions achieved by the AC and those achieved by the optimal policy, where the system knew the channel model in advance and

therefore had a relatively optimal prediction. It can be seen that our AC agent achieved a quasi-optimal scheduling policy in all contexts, even under non-stationary conditions without a priori knowledge of the channel model.

4.3. Tracking the E2E Loss Threshold

The agent's main task is to select an appropriate subset of the available connections to avoid E2E losses. Since we transmitted the information traffic together with replicas, the data were considered lost if both the information bits and their replicas were lost. The E2E loss threshold was set to 5%. As explained earlier, we set the loss probability for LOS to 0% and that for noLOS to 100%. The results presented in Figure 7 show the E2E loss rate achieved by our AC agent, the optimal policy and an RR scheduling scheme that did not use redundancy and transmitted only over a single path in a round-robin fashion without considering the channel conditions of each path. It is shown that our AC agent was able to track the predefined E2E loss threshold of 5% and achieve an average loss rate of 5.12%, which was very close to the loss rate of 4% achieved by the optimal policy. It outperformed the RR system, which had a loss rate of 8%. In each context, the agent exhibited the typical learning behavior of an RL model. At the beginning of a context, the loss was high because the model was still learning and searching for the optimal scheduling pattern according to the channel conditions of that context. However, after convergence, the loss decreased toward the end of the context until a new context began. As can be seen from Figure 7, in contexts 2, 3 and 4 with satellite elevation angles of 45° and 60°, the agent experienced higher losses compared with contexts 1, 5 and 6 with elevation angles of 60° and 70°, despite the small difference in elevation angles. This was because the LOS probabilities were quite low at 45° and 60° compared with the probabilities at an elevation angle of 70°. For example, while the minimum LOS duration at 70° was 118, it was 10 at 60° and 45° (see Table 1). According to Equation (8), this affects both the LOS probability and the loss probability. For this reason, the losses in contexts 2, 3 and 4 were high, while the losses in contexts 5 and 6 were well below the threshold in all models.

4.4. Bandwidth Utilization

Figure 8 shows the bandwidth used by our AC agent compared to the optimal policy in different contexts. To make good use of the bandwidth, the agent must determine appropriately when to use replicas to compensate for losses without wasting bandwidth. As mentioned earlier, too much redundancy can waste bandwidth, while too little redundancy means less protection. The results show that the performance of the AC agent was similar to that of the optimal policy in terms of bandwidth utilization. Both policies traded the bandwidth to overcome high loss rates in high-loss contexts (contexts 2 and 3) and use little bandwidth in contexts with relatively favorable conditions in terms of loss rate (contexts 1, 4, 5 and 6). On average, both had a repetition factor of 1.4, which corresponded to an average redundancy of 40% (i.e., 40% of the traffic was used to protect information traffic). The repetition factor of 1.4 corresponded to an average throughput of 2.1 Mbps, with 1.5 Mbps as the average source rate. On the other hand, the RR system consumed the least bandwidth, with an average bandwidth of 1.5 Mbps (omitted in the figure because it always transmits with one bit without repetitions), but it did not reach the E2E loss threshold (see Figure 7) because it did not use redundancy.

5. Conclusions

In this work, we proposed an actor-critic RL agent for LOS estimation in non-stationary conditions deployed in multi-link NTNs in dense urban environments. The simulation results showed that the learning agent had a performance similar to an optimal policy which had total knowledge of the channel model in estimating the LOS probabilities of multiple satellite links, selecting the suitable scheduling policy for the selection of the links and tracking the predefined E2E loss threshold and bandwidth utilization. Multiple links were used to increase the resilience to E2E loss, reliability, data rate and throughput and thus

improve the QoS. In this work, we outlined the handoffs between LEO satellites with real traces from the Starlink constellation that lead to an abrupt change in the elevation angles with respect to the user equipment. In future research, we plan to deepen the analysis of the handoff policies and investigate the integration of both ground and terrestrial segments.

Author Contributions: Conceptualization, A.M., A.G. and P.C.; methodology, A.G. and P.C.; software, A.M.; supervision, G.A. and C.G.; validation, A.M., A.G. and P.C.; visualization, A.M.; writing—original draft, A.M.; writing—review and editing, A.G. and P.C. All authors have read and agreed to the published manuscript.

Funding: This work has been funded by The HORIZON-CL4-2021-SPACE-01 project "5G+ evoluTion to mutioRbitAl multibaNd neTwORks" (TRANTOR) No. 101081983

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bacco, M.; Davoli, F.; Giambene, G.; Gotta, A.; Luglio, M.; Marchese, M.; Patrone, F.; Roseti, C. Networking Challenges for Non-Terrestrial Networks Exploitation in 5G. In Proceedings of the IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019; pp. 623–628.
2. 3GPP. Technical Specification Group Radio Access Network; Solutions for NR to support non-terrestrial networks (NTN): TR 38.821 V16.1.0 (2021-05), (Release 16). Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3525 (Accessed on 05 January 2023).
3. Machumilane, A.; Gotta, A.; Cassarà, P.; Bacco, M. A Path-Aware Scheduler for Air-to-Ground Multipath Multimedia Delivery in Real Time. *IEEE Commun. Mag.* **2022**, *60*, 54–58.
4. Bacco, M.; Cassará, P.; Gotta, A.; Pellegrini, V. Real-Time Multipath Multimedia Traffic in Cellular Networks for Command and Control Applications. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
5. Recommendation, I. *Propagation Data Required for The design of Earth-Space Land Mobile Telecommunication Systems*; International Telecommunication Union: Geneva, Switzerland, 2017; pp. 681–710.
6. Paasch, C.; Ferlin, S.; Alay, O.; Bonaventure, O. Experimental Evaluation of Multipath TCP Schedulers. In Proceedings of the ACM SIGCOMM Workshop on Capacity Sharing Workshop, Chicago, IL, USA, 18 August 2014; pp. 27–32.
7. Wu, J.; Yuen, C.; Cheng, B.; Shang, Y.; Chen, J. Goodput-Aware Load Distribution for Real-Time Traffic over Multipath Networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 2286–2299.
8. Houze, P.; Mory, E.; Texier, G.; Simon, G. Applicative-Layer Multipath for Low-Latency Adaptive Live Streaming. In Proceedings of the International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–7.
9. Afzal, S.; Rothenberg, C.E.; Testoni, V.; Kolan, P.; Bouazizi, I. Multipath MMT-based Approach for Streaming High Quality Video over Multiple Wireless Access Networks. *Comput. Netw.* **2021**, *185*, 1–18.
10. Mao, S.; Bushmitch, D.; Narayanan, S.; Panwar, S.S. MRTP: A Multiflow Real-Time Transport Protocol for Ad Hoc Networks. *IEEE Trans. Multimed.* **2006**, *8*, 356–369.
11. Hodroj, A.; Ibrahim, M.; Hadjadj-Aoul, Y. A Survey on Video Streaming in Multipath and Multihomed Overlay Networks. *IEEE Access* **2021**, *9*, 66816–66828.
12. Bacco, M.; Gotta, A.; Roseti, C.; Zampognaro, F. A study on TCP error recovery interaction with Random Access satellite schemes. In Proceedings of the 2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), Livorno, Italy, 8–10 September 2014; pp. 405–410. https://doi.org/10.1109/ASMS-SPSC.2014.6934574.
13. Kazemi, M.; Shirmohammadi, S.; Sadeghi, K.H. A Review of Multiple Description Coding Techniques for Error-Resilient Video Delivery. *Multimed. Syst.* **2014**, *20*, 283–309.
14. Wang, Q.; Nguyen, T.; Bose, B. Towards Adaptive Packet Scheduler with Deep-Q Reinforcement Learning. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020.; pp. 118–123.
15. Wu, H.; Alay, Ö.; Brunstrom, A.; Ferlin, S.; Caso, G. Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2295–2310.
16. Zhong, C.; Lu, Z.; Gursoy, M.C.; Velipasalar, S. A deep actor-critic reinforcement learning framework for dynamic multichannel access. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 1125–1139.
17. Yang, H.; Xie, X. An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems. *IEEE Syst. J.* **2019**, *14*, 51–60.

18. Machumilane, A.; Gotta, A.; Cassará, P.; Gennaro, C.; Amato, G. Actor-Critic Scheduling for Path-Aware Air-to-Ground Multipath Multimedia Delivery. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–5.
19. Badr, A.; Khisti, A.; Tan, W.T.; Apostolopoulos, J. Perfecting Protection for Interactive Multimedia: A survey of forward error correction for low-delay interactive applications. *IEEE Signal Process. Mag.* **2017**, *34*, 95–113. https://doi.org/10.1109/MSP.2016.2639062.
20. Sun, Y.; Fu, L. Stacking Ensemble Learning for Non-Line-of-Sight Detection of Global Navigation Satellite System. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10.
21. Aydın, V.; Çavdar, İ.H.; Hasirci, Z. Line of sight (los) probability prediction for satellite and haps communication in trabzon, turkey. *Int. J. Appl. Math. Electron. Comput.* **2016**, *1*, 155–160.
22. Granelli, F. Network slicing. In *Computing in Communication Networks*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 63–76.
23. Li, S.Y.; Liu, C. An analytical model to predict the probability density function of elevation angles for LEO satellite systems. *IEEE Commun. Lett.* **2002**, *6*, 138–140. https://doi.org/10.1109/4234.996035.
24. Lutz, E.; Cygan, D.; Dippold, M.; Dolainsky, F.; Papke, W. The land mobile satellite communication channel-recording, statistics, and channel model. *IEEE Trans. Veh. Technol.* **1991**, *40*, 375–386.
25. Bischel, H.; Werner, M.; Lutz, E. Elevation-dependent channel model and satellite diversity for NGSO S-PCNs. In Proceedings of the Proceedings of Vehicular Technology Conference-VTC, Atlanta, GA, USA, 28 April–1 May 1996; Volume 2, pp. 1038–1042.
26. Celandroni, N.; Gotta, A. Performance analysis of systematic upper layer FEC codes and interleaving in land mobile satellite channels. *IEEE Trans. Veh. Technol.* **2011**, *60*, 1887–1894.
27. Monahan, G.E. State of the art—A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Manag. Sci.* **1982**, *28*, 1–16.
28. Juan, E.; Rodriguez, I.; Lauridsen, M.; Wigard, J.; Mogensen, P. Time-correlated Geometrical Radio Propagation Model for LEO-to-Ground Satellite Systems. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 27–30 September 2021; pp. 1–5.
29. Grondman, I.; Busoniu, L.; Lopes, G.A.D.; Babuska, R. A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 1291–1307.