Deep Learning-Aided Optimization of Multi-Interface Allocation for Short-Packet Communications

Hugo De Oliveira*†, Megumi Kaneko*, Lila Boukhatem†, Ellen Hidemi Fukuda ‡

*National Institute of Informatics, Tokyo, Japan

†University Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190, Gif-sur-Yvette, France

‡Graduate School of Informatics, Kyoto University, Kyoto, Japan

e-mail: hugo.de-oliveira@universite-paris-saclay.fr, megkaneko@nii.ac.jp, lila.boukhatem@universite-paris-saclay.fr, ellen@i.kyoto-u.ac.jp

Abstract

The severe spectrum scarcity and the stringent requirements of Beyond 5G applications call for an integrated use of low frequency Sub-6 GHz and high frequency millimeter Wave bands. Focusing on future Internet of Things (IoT) Short-Packet Communications (SPC), this paper investigates the optimized usage of such diverse wireless interfaces. We propose an unifying framework devoted to SPC that jointly optimizes the user partitioning over each band, and the radio resource scheduling within each band. Leveraging Deep Reinforcement Learning (DRL) tools, the proposed method enables to better tackle the challenges imposed by dynamically varying mobile environments such as the Line-of-Sight situations of each link, and the heterogeneity of individual Quality of Service (QoS) requirements, such as rate, delay and reliability. Regarding the DRL-based user partitioning to each band, we have investigated three different types of partitioning actions to obtain a high network performance as well as a rapid convergence. Regarding the proposed sub-schedulers within each band, we designed two optimization methods, i.e., one that leverages Difference of Convex Programming (DCP) technique, and the second that accelerates convergence to a local optimum. Numerical evaluations show that the proposed methods outperform conventional approaches in terms of sum-rate and QoS outage probabilities.

Index Terms

Beyond 5G, Sub-6 GHz, millimeter Wave, Short-Packet Communications, Deep Reinforcement Learning, Resource Allocation Optimization

I. Introduction

A. Background and Problem Definition

While 5G networks introduced use cases such as enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC) and Ultra-Reliable Low-Latency Communications (URLLC), Beyond 5G (B5G) and 6G networks should cater for extreme Quality of Service (QoS) demands including massive URLLC users or Mobile Broadband Reliable Low Latency Communications (MBRLLC). Future applications will ask for ever more stringent QoS levels, jointly in terms of rate, latency and reliability, including the Terabits-level data rates for Extreme Reality or the acute reliability for remote surgery [1], [2]. Meeting such requirements will be immensely challenging under the unprecedented increase of wireless devices and the lack of available spectrum.

To mitigate the severe spectrum scarcity issue, many efforts have been devoted towards taming the high frequency millimeter Wave (mmWave) band. However, the high path loss and sensitivity to obstacles of mmWaves created a consensus on the need for an integrated network exploiting both features of conventional Sub-6 GHz and mmWave bands [3]. In such B5G integrated systems, each network entity would be equipped by multiple interfaces, thereby enabling the seamless use of Sub-6 GHz and mmWave bands, and even Terahertz bands as we head towards 6G. To meet the demands of B5G applications, coordinated radio resource allocation optimization and interference management over these multiple interfaces will be of paramount importance.

Furthermore, given the explosion of the number of IoT devices, more and more mMTC and URLLC types of applications need to be accommodated. Unlike conventional mobile broadband applications, such devices generate a large amount of small packets, entailing Short-Packet Communications (SPC) [4]. The conventional achievable rate expression given by Shannon's theorem assumes infinite codebooks, making it unsuitable to characterize the rates of SPC. However, recent advances in the field of finite blocklength information theory provided an accurate achievable rate approximation for SPC, opening the road towards designing SPC-specific resource allocation and inference management methods [5]. Although some solutions have been proposed, many open research issues remain.

B. Related Work

To overcome the uncertainties of the wireless environment while satisfying different types of requirements, many research works have focused on applying Deep Reinforcement Learning

(DRL) methods for limited size problems [6]. A Model-free Reinforcement Learning method was designed in [7] for Resource Block (RB) and power allocation. More recently, [8] and [9] proposed a DRL-based algorithm to schedule eMBB and URLLC users dynamically. These methods assign resources to URLLC users as soon as they try to send delay sensitive packets, even if the resources are used by eMBB users. Similarly, [10] used DRL-based network slicing methods to meet heterogeneous QoS requirements. In order to reduce the state-space complexity, the authors proposed to integrate an action elimination technique to the DRL algorithm to remove undesirable actions.

Leveraging finite blocklength information theory, the joint resource allocation problem for eMBB and URLLC users has been addressed in several papers. In [11], the authors studied energy efficiency optimization under a delay outage constraint. To optimize resource allocation between URLLC and eMBB users, [12] and [13] used puncturing techniques so as to meet the stringent QoS requirements of URLLC users while minimizing the throughput degradation of eMBB users. However, these works only considered conventional Sub-6 GHz bands, and hence are not applicable to integrated networks with multiple wireless interfaces.

Many recent works have focused on the joint use of the mmWave and the Sub-6 GHz bands. Reference [14] proposed to assign the users with the tightest delay requirements to the Sub-6 GHz band, based on their QoS requirements under perfect Channel State Information (CSI). The Line-of-Sight (LoS) of the remaining users is estimated by Q-Learning in order to optimize their scheduling. In [15], centralized and distributed algorithms based on DRL were proposed. The two approaches aimed at maximizing the number of satisfied users in terms of data rate, by using both Sub-6 GHz and mmWave bands. Unlike these previous works, we target SPC applications and aim at jointly exploiting the benefits of both Sub-6 GHz and mmWave bands to maximize the global sum-rate while satisfying heterogeneous QoS requirements of rate, delay, and reliability, by fully integrating users' varying channel conditions and LoS situations.

C. Contributions

In this work, we propose a unified architecture for user partitioning and scheduling over both Sub-6 GHz and mmWave bands specifically for SPC, with the goal of optimizing the global sum-rate while satisfying stringent and heterogeneous requirements of rate, delay and reliability across devices. Given the intractability of the optimization problem at hand, firstly, our proposed method optimizes user partitioning over the two bands by using a DRL-based Partitioner

which fully integrates users' LoS situations as well as their delay and rate QoS requirements. Secondly, resource allocation is solved within each band through dedicated sub-schedulers, given the specificities of SPC. Unlike a preliminary work [16], we investigate different action spaces for the DRL Partitioner and propose a new optimization method to reduce convergence time. Our main contributions are detailed as follows:

1) We design a framework based on DRL and mathematical optimization for user partitioning and scheduling of SPC which, unlike previous approaches, jointly learns, predicts, and optimizes the interface and RB allocation by fully taking into account the varying users' LoS conditions and heterogeneous QoS requirements.
2) We propose three different partitioning methods for the DRL-based Partitioner. The first proposed method achieves lower outage at the cost of computation time, while the second one is faster but results in higher outage. Finally, we investigate a third partitioning approach that strikes a trade-off between the low outage of the first approach and the computation efficiency of the second one.
3) To reduce the time complexity of the resource allocation problem solved by each sub-scheduler, we design a method leveraging both Difference of Convex Programming (DCP) and regularization while guaranteeing convergence to a local optimum.
4) Extensive numerical evaluations show that our partitioning and scheduling methods outperform baseline algorithms in terms of QoS outage probabilities, while ensuring a high global sum-rate. We also investigate and discuss the involved trade-offs between global network performance, individual QoS satisfaction, and computation efficiency.

The remainder of this paper is organized as follows. Section II introduces the system model and Section III formulates our problem. Section IV presents our proposed solution framework, then the DRL-based Partitioner is described in Section V and the sub-schedulers in Section VI. Numerical evaluations are conducted in Section VII. Finally, the conclusion and future works are given in Section VIII.

II. System Model

We consider a set of Base Stations (BS) distributed over a network area and the downlink transmissions of K uniformly distributed users and associated to each BS through Voronoi partitioning. All BSs and users can operate over both Sub-6 GHz and mmWave bands. Fig. 1 presents the network environment.

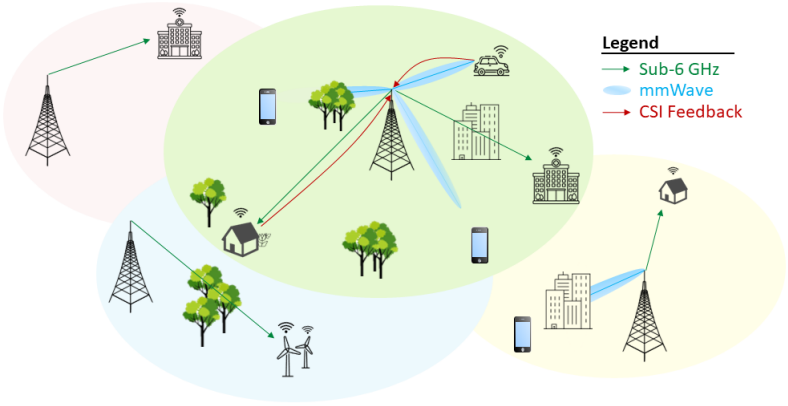


Fig. 1: Integrated mmWave-Sub-6 GHz Network

We denote the set of users as $\mathcal{K} = \{1, ..., K\}$. Let $\mathcal{K}^{\text{s6}} \subset \mathcal{K}$ be the set of users partitioned into the Sub-6 GHz band and $\mathcal{K}^{\text{mW}} \subset \mathcal{K}$ the set of users partitioned into the mmWave band. Fig. 2 illustrates the general episode and frame structure. The scheduling frame has a duration of T_f and is divided into S time slots with duration T_s. Users assigned to the Sub-6 GHz frequency are allocated over N time/frequency RBs in each time slot while users assigned to the mmWave band are allocated over M beams in each time slot. Thus, during a scheduling frame composed of S time slots, we can allocate $N \times S$ resources in the Sub-6 GHz band and $M \times S$ resources in the mmWave band, where a user assigned to a beam can make use of the whole mmWave bandwidth. BSs transmit power P^{s6} and P^{mW} for each band, equally split among RBs or beams, respectively. CSI feedback is performed per frame, hence each sub-scheduler makes use of instantaneous SINR values per user and per RB/beam.

In the Sub-6 GHz band, the SINR of user k served by BS b on RB n is given as,

$$\Gamma^{\text{s6}}_{bkn} = \frac{p^{\text{s6}}_{bkn} g^{\text{s6}}_{bkn}}{I^{\text{s6}}_{bkn} + N^{\text{s6}}_{0}}, \tag{1}$$

where p^{s6}_{bkn} is the transmit power and g^{s6}_{bkn} is the channel power (including small-scale fading and path loss) from BS b to user k on RB n. N^{s6}_{0} is the noise power and I^{s6}_{bkn} is the interference power towards user k served by BS b on RB n. Denoting $\mathcal{B}$ the set of operating BSs, the interference is given as,

$$I^{\text{s6}}_{bkn} = \sum_{b' \in \mathcal{B} \setminus \{b\}} \sum_{k' \in \mathcal{K}'} p^{\text{s6}}_{b'k'n} g^{\text{s6}}_{b'kn}, \tag{2}$$

where $p^{\text{s6}}_{b'k'n}$ is the power allocated by BS b' for user $k' \in \mathcal{K}'$ in surrounding cells, on RB n.

On the mmWave interface, we denote by β_{bkm} and θ_{bkm} the beam direction and the beamwidth from BS b to user k on beam m, respectively. β_{bkm} takes continuous values in $[0, 2\pi]$, whereas

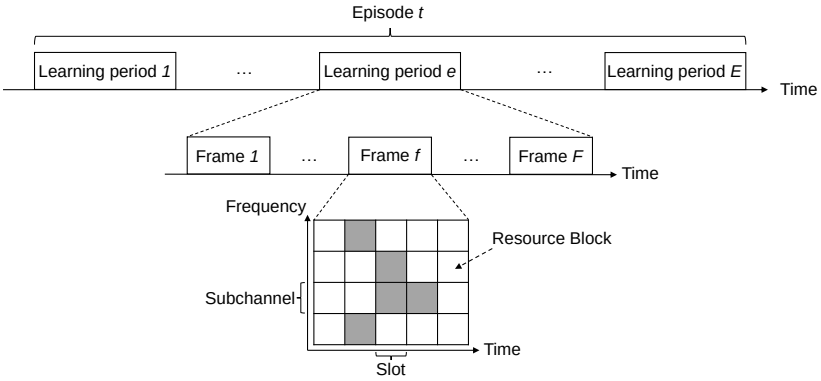


Fig. 2: Episode and Frame Structure

θ_{bkm} takes discrete values from the set of beamwidths Θ. With the rapid evolution of beamforming techniques, we can assume that the time required to change the beamdirection from a user k to another user k' is negligible, i.e., much lower than a time slot. This assumption allows us to change the beamdirection at the time slot level instead of frame level, so as to increase the resource sharing efficiency among users. The SINR from BS b to user k on beam m is given as,

$$\Gamma_{bkm}^{\mathrm{mW}} = \frac{p_{bkm}^{\mathrm{mW}}\, h_{bkm}^{\mathrm{mW}}(\beta_{bkm}, \theta_{bkm})}{I_{bkm}^{\mathrm{mW}} + N_0^{\mathrm{mW}}}, \tag{3}$$

where p_{bkm}^{mW} is the transmit power and N_0^{mW} is the noise power on mmWave band. $h_{bkm}^{\mathrm{mW}}(\beta_{bkm}, \theta_{bkm})$ is the channel power from BS b to user k on beam m, given as,

$$h_{bkm}^{\mathrm{mW}}(\beta_{bkm}, \theta_{bkm}) = G_{bkm}^{\mathrm{Tx}}(\beta_{bkm}, \theta_{bkm}) G_{bk}^{\mathrm{Rx}} h_{bkm} PL_{bk}, \tag{4}$$

where h_{bkm} is the small-scale fading parameter, PL_{bk} denotes the path loss between BS b and user k. $G_{bkm}^{\mathrm{Tx}}(\beta_{bkm}, \theta_{bkm})$ is the transmit beam gain of BS b to user k on beam m, defined as,

$$G_{bkm}^{\mathrm{Tx}}(\beta_{bkm}, \theta_{bkm}) = \begin{cases} G^{\mathrm{main}}, & \text{if } 0 < |\beta_{bk}^{\mathrm{LoS}} - \beta_{bkm}| < \frac{\theta_{bkm}}{2} \\ \mu, & \text{otherwise} \end{cases}, \tag{5}$$

where $\beta_{bk}^{\mathrm{LoS}}$ is the LoS angle between BS b and user k, G^{main} and μ are the gains of the mainlobe and sidelobe with $\mu \ll G^{\mathrm{main}}$, given by $G^{\mathrm{main}} = \frac{2\pi - (2\pi - \theta)\mu}{\theta}$. In (4), the receive beam gain G_{bk}^{Rx} for BS b to user k is assumed fixed to G^{main} for simplicity, as in [17]. Finally, in (3), I_{bkm}^{mW} denotes the interference power received by user k served by BS b on beam m expressed as,

$$I_{bkm}^{\mathrm{mW}} = \sum_{b' \in \mathcal{B} \setminus \{b\}} \sum_{k' \in \mathcal{K}'} \sum_{m' \in \mathcal{M}} p_{b'k'm'}^{\mathrm{mW}} h_{b'km'}^{\mathrm{mW}}(\beta_{b'km'}, \theta_{b'km'}) \tag{6}$$

where, unlike in Eq. (2) for orthogonal RBs, the interference may come from any beam m' allocated by any BS $b' \neq b$ to a neighboring user $k' \in \mathcal{K}'$.

Because the proposed framework is designed for SPC, the conventional Shannon capacity expression is no longer valid as it considers infinite blocklengths. Instead, we adopt the approximation for finite blocklength codes over N-parallel channels, defined in [5], Theorem (4.3.2). Denoting by Γ_n the SNR of channel n, and given $\Gamma = (\Gamma_1, ..., \Gamma_N)$, the maximum number of bits that can be sent with a packet of length l over N-parallel AWGN channels and a target error probability ϵ is given as,

$$\log M^*(l, \epsilon, \Gamma) = lC_N(\Gamma) - \sqrt{lV_N(\Gamma)}Q^{-1}(\epsilon) + O(log(l)), \tag{7}$$

where $C_N(\Gamma)$ and $V_N(\Gamma)$ are the channel capacity and channel dispersion, respectively, and are expressed as,

$$C_N(\Gamma) = \sum_{n=1}^{N} C(\Gamma_n) = \sum_{n=1}^{N} \log(1 + \Gamma_n), \tag{8}$$

$$V_N(\Gamma) = \sum_{n=1}^{N} V(\Gamma_n) = \sum_{n=1}^{N} \log^2(e) \frac{\Gamma_n(\Gamma_n + 2)}{2(\Gamma_n + 1)^2}. \tag{9}$$

For the Sub-6 GHz band, we define the binary RB allocation multidimensional array $\mathbf{x}^{\mathrm{s6}}$ of size $K \times N \times S$ with element x^{s6}_{bkns}. Similarly to [18], the achievable rate for a user k served by BS b is given as

$$R^{\mathrm{s6}}_{bk}(\mathbf{x}^{\mathrm{s6}}) = \sum_{s=1}^{S}\sum_{n=1}^{N} x^{\mathrm{s6}}_{bkns} \log(1+\Gamma^{\mathrm{s6}}_{bkn}) - \frac{Q^{-1}(\epsilon_k)}{\sqrt{l^{\mathrm{s6}}}} \sqrt{\sum_{s=1}^{S}\sum_{n=1}^{N} x^{\mathrm{s6}}_{bkns} \log^2(e) \left(1 - \frac{1}{(1+\Gamma^{\mathrm{s6}}_{bkn})^2}\right)}, \tag{10}$$

where l^{s6} defines the packet blocklengths for the Sub-6 GHz band, ϵ_k the target error probability for user k, namely the reliability metric, and Q^{-1} the inverse of the Q function.

In the mmWave band, we define the binary beam allocation multidimensional array $\mathbf{x}^{\mathrm{mW}}$ of size $K \times M \times S$ with element x^{mW}_{bkms}, the beam direction vector $\boldsymbol{\beta_{bk}}$ of size M and the beamwidth vector $\boldsymbol{\theta_{bk}}$ of size M. For a user k served by BS b, the achievable rate is:

$$\begin{aligned} R^{\mathrm{mW}}_{bk}(\mathbf{x}^{\mathrm{mW}}, \boldsymbol{\beta_{bk}}, \boldsymbol{\theta_{bk}}) = &\sum_{s=1}^{S}\sum_{m=1}^{M} x^{\mathrm{mW}}_{bkms} \log(1 + \Gamma_{bkm}(\beta_{bkm}, \theta_{bkm})) \\ &- \frac{Q^{-1}(\epsilon_k)}{\sqrt{l^{\mathrm{mW}}}} \sqrt{\sum_{s=1}^{S}\sum_{m=1}^{M} x^{\mathrm{mW}}_{bkms} \log^2(e) \left(1 - \frac{1}{(1 + \Gamma_{bkm}(\beta_{bkm}, \theta_{bkm})^2}\right)}, \end{aligned} \tag{11}$$

where l^{mW} defines the packet blocklengths for the mmWave band.

Moreover, the delay D_k experienced by user k served by BS b during a scheduling frame f is given as the last slot assigned to user k during this scheduling frame.

Finally, we adopt the LoS probability model detailed in [19] based on the distance d_{bk} between BS b and user k. If the distance d_{bk} between user k and its serving BS is smaller than a threshold Δ, we consider that $P_{\mathrm{LoS}}(d_{bk}) = 1$, i.e., user k is in LoS during his beam assignment, otherwise, $P_{\mathrm{LoS}}(d_{bk}) = Ae^{Bd_{bk}}$, i.e., it decreases exponentially when d_{bk} increases. Landscape parameters Δ, A and B are fixed depending on urban, rural or industrial scenarios.

III. Problem Formulation

In this section, we formulate the considered optimization problem. The goal is to optimize the partitioning of users among the Sub-6 GHz and mmWave bands, as well as the RB and beam allocation within each band. In this paper, we focus on maximizing the average sum-rate over time at BS b, under its associated user QoS constraints pertaining to mission-critical or URLLC types of SPC, namely a maximum delay requirement $D_k^{\max}$ expressed in terms of number of time slots, a data rate requirement b_k^{req} and finally a reliability requirement expressed in terms of target Packet Error Rate (PER) ϵ_k. We define the set of possible interfaces, namely Sub-6 GHz and mmWave, as $\mathcal{I} = \{\mathrm{s6}, \mathrm{mW}\}$. The general optimization problem is over the binary allocation matrix $\mathbf{x}^\sigma(t)$ of size $K \times (N+M) \times S$, for $\sigma \in \mathcal{I}$. The beamdirection vector $\boldsymbol{\beta}(t)$ is of size $K \times M$ and is defined in $[0; 2\pi]^M$ and the beamwidth vector $\boldsymbol{\theta}(t)$ of size $K \times M$ is defined in Θ^M, with $\Theta = \{i \times 5^o, i \in [1, ..., 10]\}$, namely,

$$\max_{\mathbf{x}^\sigma(t), \boldsymbol{\beta}(t), \boldsymbol{\theta}(t)} \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{\sigma \in \{\mathrm{s6}, \mathrm{mW}\}} R_{bk}^{\sigma}(\mathbf{x}^\sigma(t), \boldsymbol{\beta}(t), \boldsymbol{\theta}(t)) \tag{12}$$

$$\text{s.t.} \quad x_{bkps}^{\sigma} = 0, \quad s > D_k^{\max}, \forall b \in \mathcal{B},\ k \in \mathcal{K},\ p \in \mathcal{P}^\sigma, \sigma \in \mathcal{I} \tag{12a}$$

$$\sum_{\sigma \in \{\mathrm{s6}, \mathrm{mW}\}} R_{bk}^{\sigma}(\mathbf{x}^\sigma(t), \boldsymbol{\beta}(t), \boldsymbol{\theta}(t)) \geq b_k^{\mathrm{req}}, \quad \forall k \in \mathcal{K}, b \in \mathcal{B}, \sigma \in \mathcal{I} \tag{12b}$$

$$\sum_{k=1}^{K} x_{bkps}^{\sigma}(t) \leq 1, \quad \forall p \in \mathcal{P}^\sigma, \sigma \in \mathcal{I}, b \in \mathcal{B}, s \in \mathcal{S}_t \tag{12c}$$

$$\sum_{k=1}^{K} \sum_{p=1}^{P^\sigma} x_{bkps}^{\sigma}(t) \leq P^\sigma, \quad \forall \sigma \in \mathcal{I}, b \in \mathcal{B}, s \in \mathcal{S}_t \tag{12d}$$

$$x_{bkns}^{\mathrm{s6}}(t) x_{bkms}^{\mathrm{mW}}(t) = 0, \quad \forall b, k, n, m, s, \tag{12e}$$

where P^{σ} is such that $P^{\text{s6}} = N$ and $P^{\text{mW}} = M$. In Problem (12), constraint (12a) ensures that user k is assigned within $D_k^{\max}$ slots while constraint (12b) sets the rate requirements of user k. Eq. (12c) ensures that at most one user is allocated to each RB or beam during a time slot s, and (12d) that at most N RBs (M beams) are allocated in each frame t and time slot s on Sub-6 GHz (mmWave) band. Constraint (12e) prevents a given user to be allocated on both bands during the same time frame. From (10)-(11), Problem (12) is a non-linear non-convex mixed integer optimization problem which cannot be solved optimally as such. Even under fixed continuous variables $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, it remains an intricate optimization problem. To solve this problem, we propose the solution framework presented in the next sections.

IV. Proposed Framework

The proposed framework is composed of two distinct parts: the DRL-based Partitioner used to partition users among one of the two bands and the sub-schedulers used to allocate resources to each user. Fig. 3 illustrates the proposed solution.

The first entity of the proposed framework is the DRL-based Partitioner. Its role is to distribute the users among one of the two bands: the Sub-6 GHz band or the mmWave band. Due to the complexity of the band assignment problem, we used a DRL method by exploiting a Deep Q-network (DQN) that observes users' mobile environments and predicts a band assignment action, as detailed in Section V [20].

The second part of the proposed framework is composed of the two sub-schedulers: one for the Sub-6 GHz band and one for the mmWave band. Each sub-scheduler receives the user partitioning solution from the DRL-based Partitioner for their corresponding band and solves the sum-rate maximization problem within its band, subject to user individual QoS constraints.

As users have slow mobility, the partitioning solution issued by the DRL Partitioner is fixed during each learning episode of F frames, while the sub-schedulers will operate every frame to cope with the rapid wireless channel fluctuations. Hence, the metrics fed back from users to the Partitioner every F frames are averaged values over those F frames.

We next detail the mechanisms of each of the proposed entities.

V. Deep Learning-Based Partitioner

The considered DQN algorithm is defined by an action space $\mathcal{A}$, a state-space $\mathcal{S}$ and a reward function R. At every iteration, given state $s(t) \in \mathcal{S}$, the DRL-based Partitioner takes an action $a(t) \in \mathcal{A}$ that maximizes its approximated Q-value function [20]. After performing $a(t)$, the

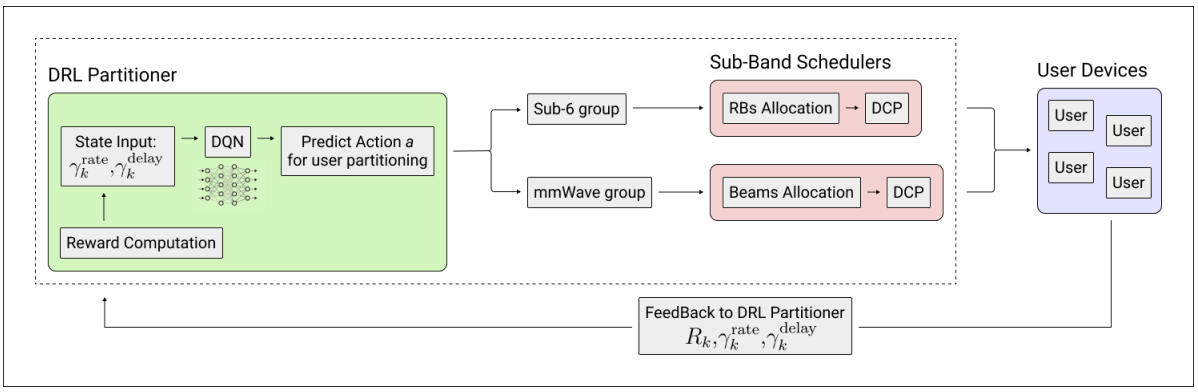


Fig. 3: Proposed General Framework

partitioner computes its corresponding reward and transitions from state $s(t)$ to state $s(t+1) \in \mathcal{S}$. In the following, we present the general structure of the considered DQN and propose three different approaches: the Binary action space, the Ternary action space and the Repartition approaches.[1]

A. General Structure

As shown in Fig. 2, each learning period lasts for F frames before a new feedback from each user is received. At the beginning of a learning period, the DQN chooses an action and assigns the users to one of the two bands. This assignment is then used by each sub-scheduler to allocate the resources. At the end of a learning period, the DQN receives the users feedbacks: the rate and delay sample outage probabilities for each user. These information are used to update the model and to optimize the chosen partition at the next iteration to find an optimal band assignment for each user. The ϵ-greedy DQN approach of [20] is taken, whereby the DQN performs exploration with probability ϵ and exploitation with probability $1-\epsilon$ and where ϵ decreases with time. Next, we propose and discuss three action space proposals, each with their own pros and cons.

B. Binary Partitioner

The Binary Partitioner is designed to take binary actions of user partitioning over the two bands and is defined as follows.

State Space: The state-space is composed of the parameters obtained from the users' feedbacks at the end of the scheduling, namely:

- The Rate reliability: $\gamma_k^{\text{rate}}(t) = 1 - \Pr\{R_k^{\text{ach}} < b_k^{\text{req}}\}$
- The Delay reliability: $\gamma_k^{\text{delay}}(t) = 1 - \Pr\{D_k^{\text{ach}} > D_k^{\max}\}$

[1]It is worth noting that, although the DQN approach is taken here, the proposed DRL-based partitioner is applicable to other DRL methods such as Double DQN.

These two measures are known at the Partitioner by tracking the long-term average of R_k^{ach} and D_k^{ach} via user feedback, the achieved rate and achieved delay of user k after scheduling, respectively. The outage probability measures in terms of rate γ_k^{rate} and delay γ_k^{delay} indicate the current level of satisfaction of the rate and delay QoS constraints, for each user.

The state-space $\mathcal{S}$ is given as:

$$\mathcal{S} = \{(\gamma_k^{\text{rate}}, \gamma_k^{\text{delay}}),\ \forall k \in \mathcal{K}\}. \tag{13}$$

Unlike our previous work [16], we only use the parameters which are modified after each learning period to reduce the state-space complexity while ensuring the efficiency of the DQN. The state-space is thus a multi-dimensional continuous state of size $2 \times K$.

Action Space: At every iteration, the DQN tries to find the best band assignment for each user in order to meet their QoS requirements. The action space is expressed as:

$$\mathcal{A}_{\text{bin}} \ = \ \{a_k, \forall k \in \mathcal{K},\ a_k \in \{1,2\}\}\text{, where } a_k \ = \ \begin{cases} 1 \text{ if Sub-6 GHz assignment} \\ 2 \text{ if mmWave assignment} \end{cases}. \tag{14}$$

At the beginning of a scheduling period, users are assigned either to the Sub-6 GHz band with action 1 or the mmWave band with action 2. Therefore, at each iteration all the users can possibly meet their QoS constraints if the partition given by the DQN allows it, i.e., if the available resources can be shared fairly and efficiently between all the users.

Every action of this Binary action space is a vector of size K composed of the assignment for each user $k \in K$, there are thus 2^K possible actions at each iteration.

Reward: Extending the reward model from [7] and given the two outage probability measures γ_k^{rate} and γ_k^{delay} for each user k, the instantaneous reward function is defined as:

$$R(a(t), s(t)) \ = \ -\sum_{k \in \mathcal{K}} \Big(\omega_k^{\text{rate}}(t)(1 \ - \ \gamma_k^{\text{rate}}(t)) \ + \ \omega_k^{\text{delay}}(t)(1 \ - \ \gamma_k^{\text{delay}}(t)) \Big) \ + \ \Lambda l_s, \tag{15}$$

with

$$w_k^{\text{rate}}(t+1) = \max\{w_k^{\text{rate}}(t) + \gamma^{\text{rate},*} - \gamma_k^{\text{rate}}(t), 0\}, \tag{16}$$

$$w_k^{\text{delay}}(t+1) = \max\{w_k^{\text{delay}}(t) + \gamma^{\text{delay},*} - \gamma_k^{\text{delay}}(t), 0\}, \tag{17}$$

where $\gamma^{\text{rate},*}$ and $\gamma^{\text{delay},*}$ are the rate and delay outage targets for all users. As the DQN strives to maximize the reward, the time-varying weights (16), (17) will be increased if the achieved outage probability measure is below the target, thereby ensuring that the system meets the outage

target. Namely, after convergence, the achieved rate and delay of each user k is guaranteed to fulfill $R_k^{\text{ach}} \geq b_k^{\text{req}}$ and $D_k \leq D_k^{\max}$ under feasible conditions, which can be proven similarly to [7, Th.1]. The last term in (15) is added to speed up the DQN to find a QoS-achieving action. In the previous work, this last term was only added if the QoS constraints for all the users were achieved. We propose a modification of this last term to accelerate the DQN convergence even if there are users with unreachable QoS constraints, namely

$$l_s = \begin{cases} 1, & \text{if } \gamma^{\text{rate}} \geq \gamma^{\text{rate},*} \text{ or } \gamma^{\text{delay}} \leq \gamma^{\text{delay},*} \ \ \forall k \in \mathcal{K} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

In Eq. (15), Λ is a scalar parameter which increases with the number of users. γ^{rate} and γ^{delay} are the average over the rate and delay outage measures achieved by all users during a learning period. This last term has the effect of rewarding the system for ending an epoch (i.e, learning episode) earlier, as soon as the average of the delay or rate outage values are better than the parameters $\gamma^{\text{rate},*}$ or $\gamma^{\text{delay},*}$. To improve the trade-off between QoS fulfillment levels and DQN convergence speed, these outage target measures may be tuned manually according to user QoS requirements and amount of available resources. This would enable to accelerate DQN convergence without stopping an epoch with an insufficient reliability. When the condition in (18) is not reached, an epoch will go on until its maximum number of iterations (frames) is reached.

The Binary Partitioner assigns all users to one of the two bands, regardless of their current channel states, QoS requirements levels and available amount of resources. Such users may lead to infeasible conditions for the sub-schedulers, thereby increasing the delays experienced by users. Indeed, the complexity of the scheduling problem increases with the number of users assigned to each band and, as we will show in the numerical evaluations part, the scheduling computation is the most time consuming part in the proposed framework.

C. Ternary Partitioner

To tackle the above issues, we propose a Ternary Partitioner which strives to balance the complexity-performance trade-off.

Using the same state space as for the Binary Partitioner, we propose to add a new action where the users are assigned to neither the Sub-6 GHz band nor the mmWave band. The new Ternary action space $\mathcal{A}_{\text{ter}}$ is defined as:

$$\mathcal{A}_{\text{ter}} = \{a_k, \forall k \in \mathcal{K},\ a_k \in \{0,1,2\}\} \text{ where } a_k = \begin{cases} 0 \text{ if no band assignment} \\ 1 \text{ if Sub-6 GHz assignment} \\ 2 \text{ if mmWave assignment} \end{cases}. \quad (19)$$

The unpartitioned users would not be able to reach their QoS requirements during the period where they are unassigned to one of the two bands, however, this action may benefit to the overall system by reducing the average outage probabilities across users.

With this method, similarly to the binary case, every action is a vector of size K but the number of possible actions is now $|A_{\text{ter}}| = 3^K$. While the size of the action space is increased compared to the Binary Partitioner, the scheduling problem becomes less complex thereby reducing its required computation time. The impact of this partitioning choice on the performance-complexity trade-off will be further discussed through the numerical evaluations.

Moreover, the same reward model as for the Binary Partitioner (see Section V-B) applies for this case. By considering the average over users of the outage probability measures as a stopping criteria for an epoch, condition (18) can be achieved even if there are unpartitioned users while maintaining low levels of individual outage probabilities, as will be shown in the numerical results.

D. Repartition Method

After convergence, the proposed DQN method predicts the action with the highest Q-value, which fulfills the stopping criteria in (18). When the DQN reaches this state, the same action will be predicted until a change of environment. With the Ternary Partitioner, this means that some users may be unpartitioned for a long period of time, leading to an unequal situation.

To improve fairness among users, we propose a simple yet effective method that forces long-term unpartitioned users to be assigned to a band. Algorithm 1 describes the main steps of this Repartition Method. Namely, a random band assignment is applied to users being assigned action 0 (unpartitioned) during τ consecutive periods.

The same state space is used as the one in the Ternary and the Binary partitioners. Since this method is a variant of the Ternary Partitioner, the same action space $\mathcal{A}_{\text{ter}}$ is considered. Hence, there are three possible actions for each user during each period and the number of possible actions is 3^K. By using this action space, the DQN can converge faster to an optimal action than the Binary Partitioner due to the reduced number of users assigned to a band during each period. After convergence, the long-term unpartitioned users are randomly reassigned to a

Algorithm 1 Proposed General Framework (with Repartition Method)

Initialize DQN Q with random weights;
Set $s_1 = s_{\text{init}}$;
Choose λ;
$\epsilon = 1$;
for $t = 1, 2, ..., T$ **do**
    $\epsilon \leftarrow \epsilon \times \lambda$;
    if random number $p < \epsilon$ **then**
        Select action a_t randomly;
    else
        Select action a_t with max $Q(s_t, a_t)$;
        if $a_t = a_{t-i}$, $\forall i \in [1, ..., \tau]$ **then**
            Select random partitioning for unpartitioned users;
    Send partitioning information of action a_t to the sub-schedulers;
    for each $\sigma \in \{\text{sub6GHz}, \text{mmWave}\}$ **do**
        for $p = 1, 2, ...\rho$ **do**
            Schedule user assigned to σ's band following Alg. 2;
            Apply optimized schedule and receive feedback from each user k;
        Aggregate ρ past feedbacks into outage probability measures for (15);
    Calculate reward of action a_t by (15);
    Update Q's weights;
    Update new state $s_{t+1} \leftarrow s_t$;

band as described in Algorithm 1. This behavior can be regarded as adaptive switching from ternary actions to binary actions which can better meet the QoS constraints of all the users, while reducing time complexity as will be shown through the numerical results. Finally, the same reward model is used as in the Binary and Ternary Partitioners.

VI. Sub-Schedulers Optimization

In this section, we present the mathematical optimization methods used by each sub-scheduler. To solve the intricate optimization problem formulated in Section III, we propose two different optimization approaches: the first one leveraging DCP (referred to as *Optim-DCP*), and the second one based on DCP with Regularization (referred to as *Optim-DCP-Reg*).

At a given scheduling time frame t (index hereafter omitted for sake of clarity), Problem (12) becomes:

$$\max_{\mathbf{x}^\sigma} \quad \sum_{k=1}^{K^\sigma} R_{bk}^\sigma(\mathbf{x}^\sigma) \tag{20}$$

$$\text{s.t} \quad x_{bpks}^\sigma = 0, \quad s > D_k^{\max}, \forall b \in \mathcal{B}, p \in \mathcal{P}, k \in \mathcal{K} \tag{20a}$$

$$R_{bk}^\sigma(\mathbf{x}^\sigma) \geq b_k^{\text{req}}, \quad \forall k \in \mathcal{K}^\sigma, b \in \mathcal{B} \tag{20b}$$

$$\sum_{k=1}^{K^\sigma} x_{bkps}^\sigma \leq 1, \quad \forall b \in \mathcal{B}, p \in \mathcal{P}^\sigma, s \in \mathcal{S}_t \tag{20c}$$

$$\sum_{k=1}^{K^\sigma} \sum_{p=1}^{P^\sigma} x_{bkps}^\sigma \leq P^\sigma, \quad \forall b \in \mathcal{B}, s \in \mathcal{S}_t, \tag{20d}$$

with $\sigma \in \{\text{s6}, \text{mW}\}$ and $P^{\text{sub6}} = N$ and $P^{\text{mW}} = M$. Thus, for the Sub-6 GHz band, the allocation matrix $\mathbf{x}^{\text{s6}}$ is of size $K \times N \times S$ and for the mmWave band the allocation matrix $\mathbf{x}^{\text{mW}}$ is of size $K \times M \times S$.

It is worth mentioning that, given the difficulty of the initial problem (12) on the mmWave band, we consider the beamforming parameters β and θ fixed given the assigned user. Namely, the beamdirection is fixed towards the users' LoS direction, $\beta_{bkm} = \beta_{bk}^{\text{LoS}}$, while the narrowest beamwidth will be considered so as to maximize user rate, i.e., $\theta_{bkm} = \theta_{\min}$. The Sub-6 GHz and the mmWave sub-problems both boil down to the binary user assignment problem of (20).

We next transform Problem (20) similarly to [18] by introducing

$$F(\mathbf{x}^\sigma) = \sum_{k=1}^{\text{K}^\sigma} \sum_{p=1}^{P^\sigma} \sum_{s=1}^{S} x_{bkps}^\sigma \log(1 + \Gamma_{bkp}^\sigma), \tag{21}$$

$$V(\mathbf{x}^\sigma) = \frac{Q^{-1}(\epsilon_k)}{\sqrt{l^\sigma}} \sqrt{\sum_{k=1}^{\text{K}^\sigma} \sum_{p=1}^{P^\sigma} \sum_{s=1}^{S} x_{bkps}^\sigma \log^2(e) \left(1 - \frac{1}{(1 + \Gamma_{bkp}^\sigma)^2}\right)}. \tag{22}$$

Finally, the general problem can be written as,

$$\max_{\mathbf{x}^\sigma} \quad F(\mathbf{x}^\sigma) - V(\mathbf{x}^\sigma) \tag{23}$$

$$\text{s.t } (20a), (20b), (20c), (20d). \tag{23a}$$

Furthermore, we define the capacity and dispersion achieved by a user k during a scheduling time frame as,

$$F_k(\mathbf{x}^\sigma) = \sum_{p=1}^{P^\sigma} \sum_{s=1}^{S} x_{bkps}^\sigma \log(1 + \Gamma_{bkp}^\sigma) \tag{24}$$

$$V_k(\mathbf{x}^\sigma) = \frac{Q^{-1}(\epsilon_k)}{\sqrt{l^\sigma}} \sqrt{\sum_{p=1}^{P^\sigma} \sum_{s=1}^{S} x_{bkps}^\sigma \log^2(e) \left(1 - \frac{1}{(1 + \Gamma_{bkp}^\sigma)^2}\right)}. \tag{25}$$

Hereafter, we detail our proposed approaches for solving Problem (23).

A. Optimization based on Difference of Convex Programming (Optim-DCP)

The first proposed method is similar to that of [16]. Problem (20) can now be resolved through the following steps:

Step 1: Integer Relaxation

The binary constraint $x_{bkps}^\sigma \in \{0, 1\}$ is first relaxed into an equivalent convex form as follows:

$$W(\mathbf{x}^\sigma) - E(\mathbf{x}^\sigma) \leq 0 \quad \& \quad 0 \leq x_{bkps}^\sigma \leq 1, \quad \forall b \in \mathcal{B}, k \in \mathcal{K}, p \in \mathcal{P}, s \in \mathcal{S}_t, \tag{26}$$

where $W(\mathbf{x}^\sigma) = \sum_{s=1}^{S} \sum_{k=1}^{K^\sigma} \sum_{p=1}^{P^\sigma} x_{bkps}^\sigma$ and $E(\mathbf{x}^\sigma) = \sum_{s=1}^{S} \sum_{k=1}^{K^\sigma} \sum_{p=1}^{P^\sigma} (x_{bkps}^\sigma)^2$.

Step 2: Rewriting into a Difference of Convex (DC) problem

Rewriting the objective function as

$$U(\mathbf{x}^\sigma) = U_1(\mathbf{x}^\sigma) - U_2(\mathbf{x}^\sigma), \tag{27}$$

where $U_1(\mathbf{x}^\sigma)$ and $U_2(\mathbf{x}^\sigma)$ are defined as

$$U_1(\mathbf{x}^\sigma) = -F(\mathbf{x}^\sigma) + \beta W(\mathbf{x}^\sigma) \tag{28}$$

$$U_2(\mathbf{x}^\sigma) = -V(\mathbf{x}^\sigma) + \beta E(\mathbf{x}^\sigma), \tag{29}$$

we can show, similarly to [18], that for a large value of $\beta > 1$, Problem (20) is equivalent to Problem (30) below,

$$\min_{\mathbf{x}^\sigma} U(\mathbf{x}^\sigma) \tag{30}$$

$$\text{s.t } (20a), (20c), (20d) \tag{30a}$$

$$F_k(\mathbf{x}^\sigma) - V_k(\mathbf{x}^\sigma) \geq b_k^{\text{req}} \quad , \forall k \in \mathcal{K}^\sigma \tag{30b}$$

$$W(\mathbf{x}^\sigma) - E(\mathbf{x}^\sigma) \leq 0 \quad \& \quad 0 \leq x_{bkps}^\sigma \leq 1, \quad \forall b \in \mathcal{B}, k \in \mathcal{K}, p \in \mathcal{P}, s \in \mathcal{S}_t. \tag{30c}$$

Note that U is a DC function, as F and V are both concave and that both W and E are convex (straightforward by definition). Indeed, F is by definition a weighted sum of concave functions (logarithms), which conserves concavity so F is concave. Concavity of V can be shown by the non-negativity of its second derivative.

Step 3: Transformation into a convex problem

Using the first order approximations for convex function E and concave function V, we can have an upper bound for U as follows,

$$U(\mathbf{x}^{\sigma}) \leq U_1(\mathbf{x}^{\sigma}) - (U_2(\mathbf{x}^{\sigma,(j)}) - \nabla_x U_2(\mathbf{x}^{\sigma,(j)})^T(\mathbf{x}^{\sigma} - \mathbf{x}^{\sigma,(j)})) = \bar{U}(\mathbf{x}^{\sigma}, \mathbf{x}^{\sigma,(j)}), \tag{31}$$

where $\mathbf{x}^{\sigma,(j)}$ is some iterate of the algorithm. Thus, using this bound, and once again the approximation of V, we obtain the following problem,

$$\min_{\mathbf{x}^{\sigma}} \bar{U}(\mathbf{x}^{\sigma}, \mathbf{x}^{\sigma,(j)}) \tag{32}$$

$$\text{s.t } (20a), (20c), (20d) \tag{32a}$$

$$F_k(\mathbf{x}^{\sigma}) - V_k(\mathbf{x}^{\sigma,(j)}) - \nabla_x V_k(\mathbf{x}^{\sigma,(j)})^T(\mathbf{x}^{\sigma} - \mathbf{x}^{\sigma,(j)})) \geq b_k^{\text{req}}, \forall k \in \mathcal{K}^{\sigma} \tag{32b}$$

$$W(\mathbf{x}^{\sigma}) - E(\mathbf{x}^{\sigma}) \leq 0 \quad \& \quad 0 \leq x^{\sigma}_{bkps} \leq 1, \quad \forall b \in \mathcal{B}, k \in \mathcal{K}, p \in \mathcal{P}, s \in \mathcal{S}_t. \tag{32c}$$

Problem (32) is now convex and can be easily solved by a standard convex optimization toolbox. The overall scheduling problem may thus be solved using Algorithm 2 based on [18], [21].

Algorithm 2 Sub-Schedulers Optimization, *Optim-DCP*

Result: a local optimum solution for Problem (32)

Initialization: Set $j = 1$ the iteration index, $J_{\max}$ the maximum number of iterations, $\beta > 1$ the penalty factor, the initial schedule $\mathbf{x}^{\sigma,(1)}$, $\delta > 0$ the optimal tolerance, $V_k(\mathbf{x}^{\sigma,(1)})$, $F_k(\mathbf{x}^{\sigma,(1)})$, $\forall k \in \mathcal{K}^{\sigma}$.

while $j < J_{\max}$ **do**

- Solve optimization problem (32) for a given point $\mathbf{x}^{\sigma,(j)}$ using a convex solver. Get solution $\mathbf{x}^{\sigma,*}$
- Update $V_k(\mathbf{x}^{\sigma,(j+1)}) = V_k(\mathbf{x}^{\sigma,*})$, $F_k(\mathbf{x}^{\sigma,(j+1)}) = F_k(\mathbf{x}^{\sigma,*})$
- **if** $||\mathbf{x}^{\sigma,*} - \mathbf{x}^{\sigma,(j)}|| \leq \delta$ **then**
  - **Return:** $\mathbf{x}^{\sigma,*}$
- **end**
- update $\mathbf{x}^{\sigma,(j+1)} = \mathbf{x}^{\sigma,*}$
- Set $j = j + 1$

end

Return: $\mathbf{x}^{\sigma,(j)}$

Each iteration solves Problem (32) and then uses the previous iteration's solution to solve it again. If the gap between two consecutive solutions is smaller than a given δ, we consider that

we are close enough to a local minimum and we stop the algorithm. Otherwise, it stops when the maximum number of iterations is reached. This methodology is shown in [22] to converge globally with linear rate.

B. Optimization based on DCP with Regularization (Optim-DCP-Reg)

To reduce the computation time of the sub-schedulers, which will be shown in Section VII to be much higher than that of the DRL Partitioners, we have investigated the regularized DCP approach, by adding a regularization term to the objective function of Problem (32), thereby giving:

$$\min_{\mathbf{x}^\sigma} \bar{U}(\mathbf{x}^\sigma, \mathbf{x}^{\sigma,(j)}) + \frac{\rho_j}{2}||A_j(\mathbf{x}^\sigma - \mathbf{x}^{\sigma,(j)})||^2 \tag{33}$$

$$\text{s.t } (20a), (20c), (20d), (32b), (32c). \tag{33a}$$

Here, it is assumed that A_j is a given full rank matrix, $\rho_j > 0$, $\lim_{j\to\infty} \rho_j = 0$, $\sum_{j=1}^{\infty} \rho_j = \infty$. Note that the constraints remain unchanged as to Problem (32). Problem (33) is proved to have global convergence and can be solved using a generic convex solver. Similarly to Algorithm 2, Algorithm 3 describes the main steps for the resolution of Problem (33). In particular, Algorithm 3 uses the previous iteration's solution to find a local minimum. The main difference with Algorithm 2 is the addition of the regularization term and the updates of ρ_j and A_j. At each iteration j, Algorithm 3 can ensure that $\lim_{j\to\infty} \rho_j = 0$ and $\sum_{j=0}^{\infty} \rho_j = \infty$ by taking $\rho_j = \frac{1}{j+1}$.

Algorithm 3 Sub-Schedulers Optimization, *Optim-DCP-Reg*

Result: a local optimum solution for Problem (33)

Initialization: Set $j = 1$ the iteration index, $J_{\max}$ the maximum number of iterations, $\beta > 1$ the penalty factor, the initial schedule $\mathbf{x}^{\sigma,(1)}$, $\delta > 0$ the optimal tolerance, $\rho_0 > 0$, A_0 the identity matrix, $V_k(\mathbf{x}^{\sigma,(1)})$, $F_k(\mathbf{x}^{\sigma,(1)})$, $\forall k \in \mathcal{K}^\sigma$.

while $j < J_{\max}$ **do**

- Solve optimization problem (33) for a given point $\mathbf{x}^{\sigma,(j)}$ using a convex solver. Get solution $\mathbf{x}^{\sigma,*}$
- Update $V_k(\mathbf{x}^{\sigma,(j+1)}) = V_k(\mathbf{x}^{\sigma,*})$, $F_k(\mathbf{x}^{\sigma,(j+1)}) = F_k(\mathbf{x}^{\sigma,*})$
- **if** $||\mathbf{x}^{\sigma,*} - \mathbf{x}^{\sigma,(j)}|| \leq \delta$ **then**
  - **Return:** $\mathbf{x}^{\sigma,*}$
- **end**
- update $\mathbf{x}^{\sigma,(j+1)} = \mathbf{x}^{\sigma,*}$
- update ρ_j
- update A_j
- Set $j = j + 1$

end

Return: $\mathbf{x}^{\sigma,(j)}$

Notation	Parameter	Value	Notation	Parameter	Value
T_f	Scheduling time frame	1 ms	N_0	Noise power spectral density	-174 dBm/Hz
T_s	Time slot duration	0.2 ms	θ	Beamwidth	15°
P_1, P_2	Sub-6 and mmW BS transmit power	30 dBm, 30 dBm	μ	Sidelobe gain	0.1
Ω_1, Ω_2	Sub-6 and mmW available Bandwidth	100 MHz, 1 GHz	Δ	Distance threshold for LoS prob.	25.5m [23]
Σ	Standard deviation of mmW path loss	5.8	A, B	Scenario parameters for LoS prob.	1.283, -0.009808 [23]
α_1, α_2	Sub-6 and mmW path loss exponent	3.0, 2.0	Λ	Success reward scalar	1000
S	Sched. frame size (in time slots)	5	E	Sched. period (in scheduling frames)	10
T	Number of epochs	200	λ	Exploration Rate Decay factor	0.99, 0.995, ...

TABLE I: Simulation parameters based on [14], [23]

VII. Numerical Evaluations

A. Simulation Settings

We consider a 300×300 m^2 network area with 9 cells containing a central BS each and uniformly distributed users. The evaluations will focus on the central BS, while the surrounding BSs will generate interference. The parameters used for our simulations are described in Table I.

For each proposed approach, we compared two different scenarios as follows,

- *Scenario* 1: Some users require a high data rate but a looser delay while other users demand a stringent delay but a small data rate.
- *Scenario* 2: Most users do not require a high data rate but a stringent delay.

Each scenario is evaluated for parameter values $(K, N, M) = (8, 3, 3)$ and $(K, N, M) = (10, 4, 4)$. The QoS requirements for $K = 8$ and $K = 10$ users are presented in Tables II and III, respectively. For each scenario, we used a maximum error probability $\epsilon_k = 10^{-5}$ to compute the achievable rate (Eqs. (10), (11)) in a scheduling period. The outage probabilities are computed during each frame, whereby the delay outage probability uses the instantaneous delay reached during frame i while the rate outage probability uses the average achieved rates since the start of a learning period.

The learning parameters are also given in Table I. The simulations are run over 200 episodes as shown in Fig. 2, where each episode is composed of a maximum of 40 learning periods, each consisting of 40 actions. Each learning period lasts for 10 scheduling frames.

As an epoch stops when the rate outage probability averaged over users is lower than $\gamma^{\mathrm{rate},*}$ or the delay outage probability averaged over users is lower than $\gamma^{\mathrm{delay},*}$, these terms are chosen smaller for the Ternary Partitioner than for the Binary Partitioner. This allows the Ternary

	User1		User2		User3		User4		User5		User6		User7		User8	
Scenario	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Rate Required (Mbit/s)	500	250	50	30	1	50	150	30	25	1	1	20	250	300	250	50
Max Delay	5	5	5	4	3	5	5	3	4	2	4	4	4	5	5	4
LoS probability	1		0.885		0.856		0.844		0.787		0.875		1		0.917	

TABLE II: QoS requirements for $K = 8$, $N = 3$ and $M = 3$, Scenarios 1 and 2

Partitioner to stop an epoch with unpartitioned users while the Binary Partitioner must fulfill the users requirements before stopping an epoch. Namely, we chose $\gamma_{\text{ter}}^{\text{rate},*} = \gamma_{\text{ter}}^{\text{delay},*} = 0.2$ for the Ternary Partitioner and $\gamma_{\text{bin}}^{\text{rate},*} = \gamma_{\text{bin}}^{\text{delay},*} = 0.15$ for the Binary Partitioner, for all scenarios.

B. Benchmark Methods

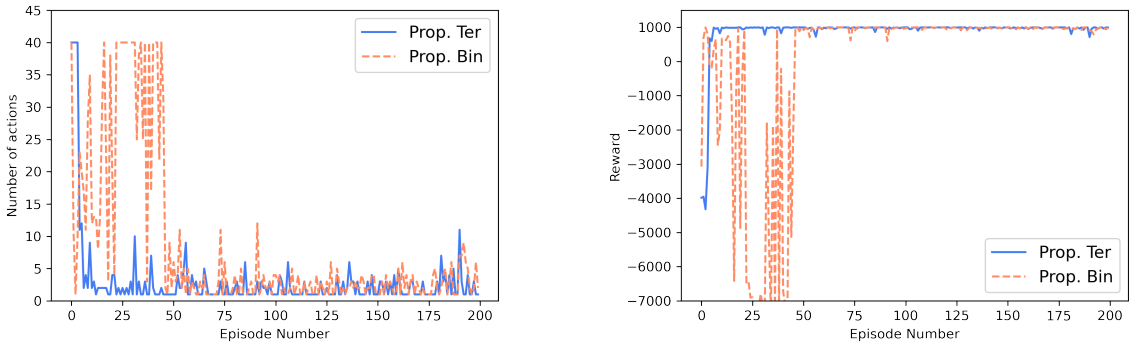
The performance of our proposed framework is compared to three reference partitioning methods:

- Conventional Highest LoS partitioning (*Conv. HLoS*): Users are split in half, those with the highest LoS probability are assigned to mmWave while the other half is assigned to Sub-6 GHz.
- Conventional Highest Requirement partitioning (*Conv. HReq*): Users are split in half, those with the highest required rate are assigned to mmWave while the other half is assigned to Sub-6 GHz.
- Conventional Threshold LoS partitioning (*Conv. ThresLoS*): Given a LoS probability threshold ζ_{LoS}, users that satisfy $\text{LoS}_\text{k} > \zeta_{\text{LoS}}$ are assigned to mmWave while the rest is assigned to Sub-6 GHz. As mmWaves require sufficient LoS, the LoS threshold is chosen close to one, with $\zeta_{\text{LoS}} = 0.9$.

The sub-schedulers of the reference methods operate similarly as our proposed one, as existing heuristic sub-schedulers would perform worse. The results of the baseline methods are averaged over 10000 scheduling frames, using the same randomly generated set of channels as for the proposed methods.

	User1	User2	User3	User4	User5	User6	User7	User8	User9	User10
Rate Required (Mbit/s)	500	50	1	25	150	100	250	1	25	20
Max Delay	5	4	5	3	3	4	5	4	5	5
LoS Probability	1	0.885	0.856	0.844	0.868	1	0.933	0.876	0.816	0.918

TABLE III: QoS requirements for K=10, N=4 and M=4, Scenario 1



(a) Number of Actions per Episode, Scenario 1, $K = 8$

(b) Reward per Episode, Scenario 1, $K = 8$

Fig. 4: Learning Behavior for Scenario 1 with $K = 8$ users

We first compare the proposed Binary and Ternary Partitioners with the baseline methods, and then we evaluate the proposed Repartition Method. Finally, we provide a detailed analysis of the performance of the proposed sub-schedulers' optimization methods.

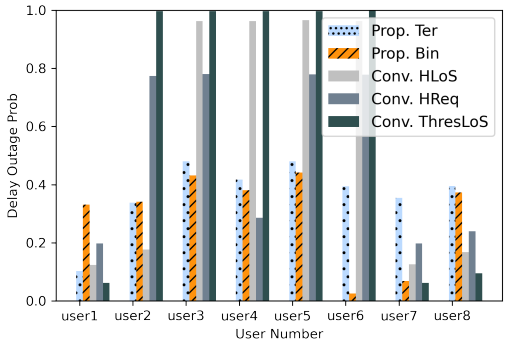
C. Binary and Ternary Partitioning

In this section, we compare the Binary and Ternary Partitioners against the baselines by using the first optimization method.

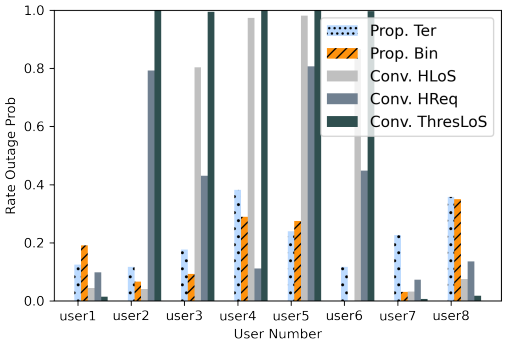
Learning Behavior: In Fig. 4, we show the learning behavior of the proposed DRL-based Partitioner in terms of the number of actions and immediate reward per episode, for Scenario 1 with $K = 8$. As we can see, the Binary approach requires more episodes than the Ternary approach to converge to a feasible action, given by the stopping criteria defined in Eq. (18).

Performance Evaluation: We compare the rate and delay outage probabilities of the Ternary and Binary proposed methods as opposed to the benchmarks algorithms, for both scenarios and for $K = 8$ users.

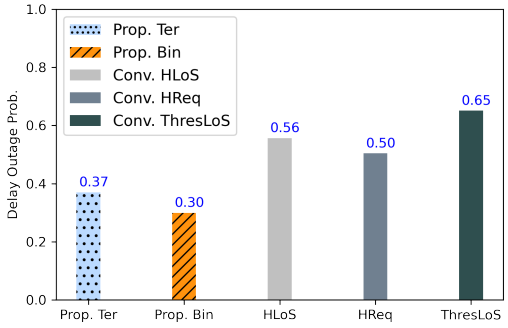
In Figs. 5 and 6, we observe the rate and delay outage probabilities after convergence for all algorithms. We can see that, in both scenarios, the proposed approaches outperform clearly the baselines and, as expected, the Binary method performs better than the Ternary one. Due to the binary action, all users are assigned to one of the two bands and strive to meet their QoS requirements while the Ternary Partitioner may not assign any band to some users, which will be unable to reach their QoS. In Fig. 6(b), we show the rate outage probability per user for Scenario 2 where we clearly see the effect of the ternary action space: user 5 is often unpartitioned and has a rate outage probability close to 0.8, allowing other users to meet their QoS requirements and experience a low rate outage probability. Despite that, the average rate outage probability for this



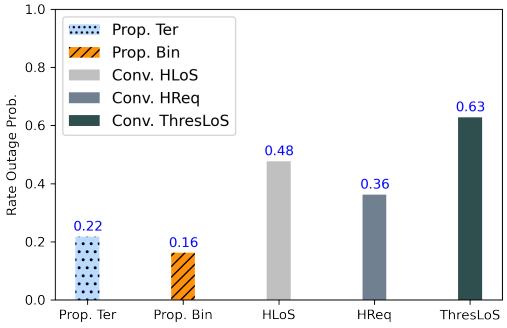
(a) Delay Outage Proba. per User



(b) Rate Outage Proba. per User



(c) Delay Outage Proba. per User



(d) Rate Outage Proba. per User

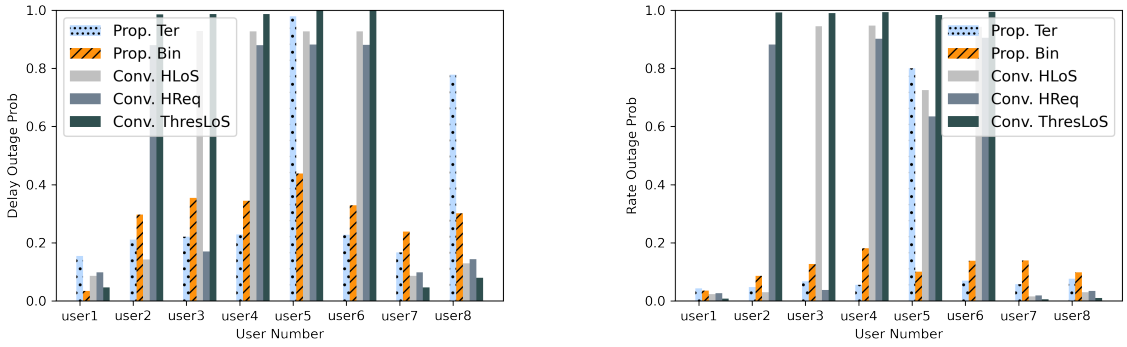
Fig. 5: Outage Probability Performances for Scenario 1, $K = 8$

scenario is significantly lower than the one of the baseline methods, even with an unpartitioned user. *Conv. HLoS* and *Conv. HReq* assign the users equally between Sub-6 and the mmWave frequency bands. Clearly, users partitioned into the Sub-6 GHz band reach outage probabilities close to 1, as there are not enough resources to fulfill their QoS requirements. Similarly, *Conv. ThresLoS* assigns in these scenarios most of the users to Sub-6 GHz and these users cannot fairly share the resources and reach their QoS targets.

In Table IV, we show the average sum-rate after convergence for scenarios 1 and 2 with 8 users. We can observe that the average sum-rate of *Conv. ThresLoS* is by far higher than that reached by the other methods, but in return, results in the worst outage probabilities as shown in Figs. 5 and 6. *Conv. ThresLoS* assigns only users with a LoS probability greater than the threshold $\zeta_{\text{LoS}} = 0.9$ to mmWave, given the high LoS conditions required for mmWave, and the rest to Sub-6 GHz. This creates an unbalanced environment with too many users assigned to

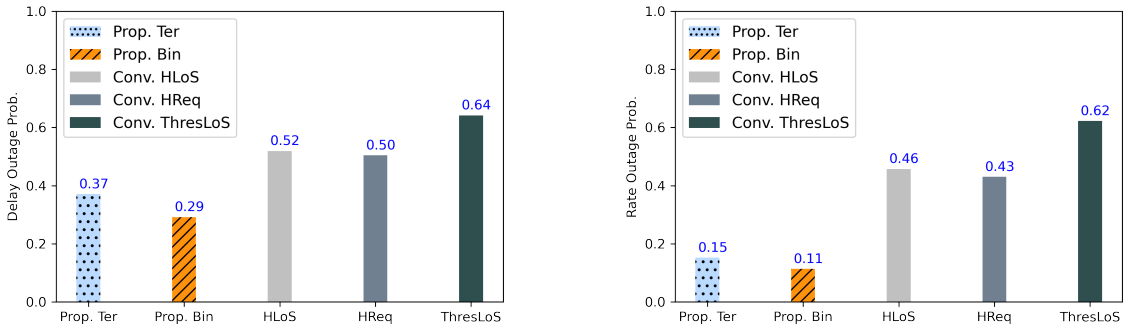
Method	Prop. Bin	Prop. Ter	*Conv. HLoS*	*Conv. HReq*	*Conv.ThresLoS*
Scenario 1	2.009	2.838	4.281	3.775	7.265
Scenario 2	2.470	3.094	4.928	4.771	7.710

TABLE IV: Average Sum-Rate (Gbit/s) after Convergence for Scenarios 1 and 2, $K = 8$ users



(a) Delay Outage Proba. per User

(b) Rate Outage Proba. per User



(c) Delay Outage Proba. per User

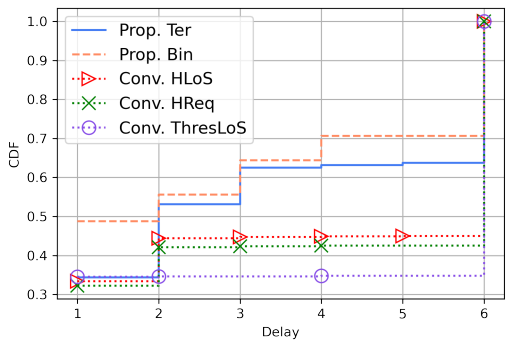
(d) Rate Outage Proba. per User

Fig. 6: Outage Probability performances for Scenario 2, $K = 8$

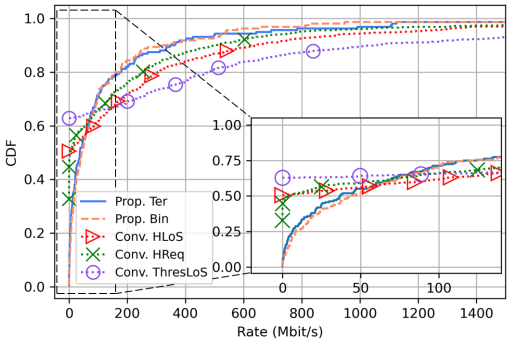
Sub-6 GHz in these scenarios. Moreover, *Conv. HLoS* and *Conv. HReq* reach a higher sum-rate than the proposed methods. In particular, the Binary Partitioner has clearly the lower average sum-rate after convergence but the resources are more equally shared and, as shown in Figs. 5(d) and 6(d), enables minimum rate outage probabilities.

Fig. 7 shows the Cumulative Distribution Function (CDF) of rate and delay for Scenario 1 with 8 users for all methods, namely Fig. 7(a) for the delay CDF and Fig. 7(b) for the rate CDF. When a user reaches a delay of 6 slots, it means that there are no resources allocated to this user, as each scheduling frame is composed of 5 time slots. We can see on these figures that the probability to have a delay lower than 6 is much smaller with the baselines methods than with the proposed approaches. As *Conv. ThresLoS* assigns a majority of users to Sub-6 GHz, they cannot share properly the resources thereby inducing the highest probability of having a delay equal to 6.

In Fig. 7(b) we show the rate CDF where the region between 0 and 150 Mbit/s is zoomed. Similarly to the case of outage probabilities, the probability for *Conv. ThresLoS* to reach a data rate equal to 0 is very high. In general, the baseline methods entail a higher probability to achieve



(a) Delay CDF, Scenario 1

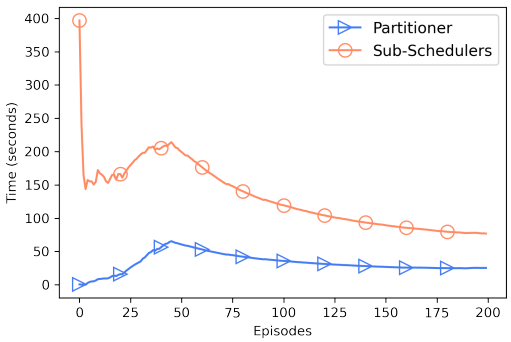


(b) Rate CDF, Scenario 1

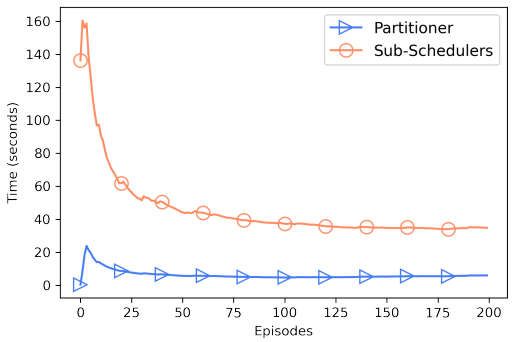
Fig. 7: CDFs of Delay (left) and Rate (right), Scenario 1, $K = 8$

high data rates, compared to our proposed methods, as they advantage data rate against fairness and user satisfaction. However, our proposed approaches outperform the baselines for small data rates lower than 50 Mbit/s, indicating a better rate fairness. The probability to have a rate equal to 0 is significantly reduced with the proposed approaches and, even if the rates are in general lower, this enables to significantly improve rate outage probabilities.

Execution Time: Due to the larger number of users to schedule, the execution time of the Binary Partitioner is by far higher than that of the Ternary Partitioner. Fig. 8 shows the computation time comparison between the sub-schedulers and the DRL algorithm for both the Binary (Fig. 8(a)) and Ternary (Fig. 8(b)) methods for Scenario 1. We can observe that the DQN computation time shows an almost constant behavior while the computation time of the sub-schedulers increases during the learning time before gradually decreasing. We can also see that the computation times of the sub-schedulers are much higher than the learning computation time and greatly influence the overall computation time.

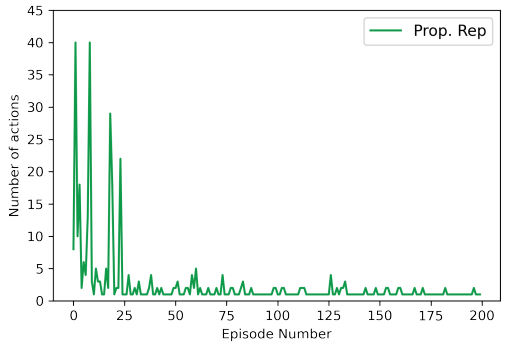


(a) DQN vs Optim, Binary, Scenario 1

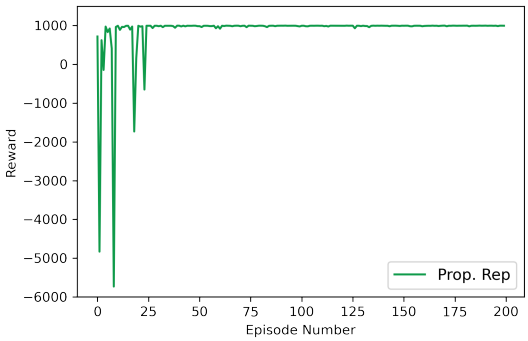


(b) DQN vs Optim., Ternary, Scenario 1

Fig. 8: Execution time DQN vs Sub-schedulers, Scenario 1, $K = 8$



(a) Number of Actions per Episode, $K = 10$



(b) Reward per Episode, $K = 10$

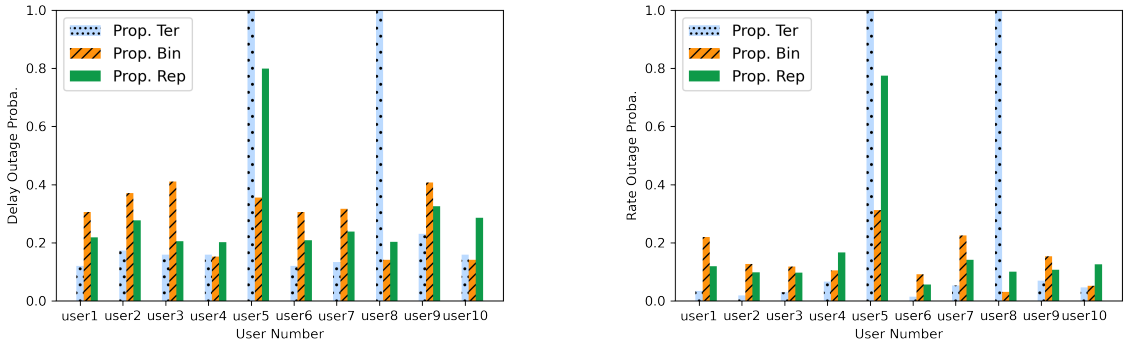
Fig. 9: Learning Behavior for the Repartition Method, $K = 10$

From these results, we can conclude that, compared to the Binary approach, the ternary action space allows to reduce the sub-schedulers optimization time complexity at the expense of outage. In the next section, we show the performance of the proposed Repartition Method whose aim is to reduce the computation time of the Binary Partitioner while trying to approach its performance.

D. Repartition Method

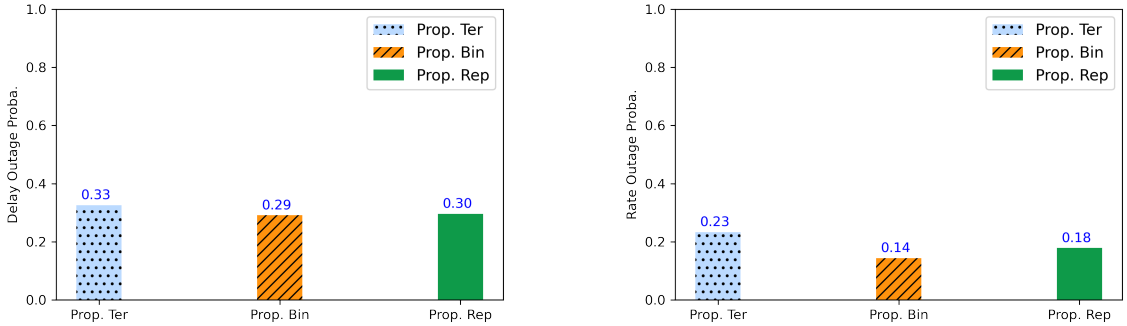
Learning Behavior: Fig. 9 depicts the learning behavior of the proposed Repartition Method for $K = 10$ users. The global learning behavior of this method is close to that of the Ternary approach. This method converges faster than the Binary approach but needs more episodes than the Ternary one due to the Repartition algorithm. As this method is designed to first use a ternary action and then to repartition users, the stopping criteria here is the same as the Ternary approach, but the final predicted action is a binary partition where every user is assigned to a band.

Performance Evaluation: We present in Fig. 10 the rate and delay outage probabilities of the Ternary, Binary and Repartition approaches for $K = 10$ users. The outage performance achieved by the Repartition approach is lying between the Binary and Ternary methods. In Figs. 10(a), 10(b) we clearly see that the Ternary method does not assign users 5 and 8, resulting in their high outage probabilities, while other users can share the available resources more fairly and reach a low outage. We also observe the behavior of the Repartition Method: user 5 has been unpartitioned for some time but, he is reassigned to a frequency band and benefits from reduced rate and delay outage probabilities (lower than 0.8). In Figs. 10(c), 10(d) we show the average delay and rate outage probabilities after convergence. As we expected, the outage probabilities



(a) Delay Outage Proba. per User

(b) Rate Outage Proba. per User



(c) Average Delay Outage Proba.

(d) Average Rate Outage Proba.

Fig. 10: Outage Probability Performance for $K = 10$

obtained with the Repartition approach are comprised between those of the Ternary and Binary methods.

Due to the above-mentioned behavior, the Binary method reaches the lowest average sum-rate (2.085 Gbit/s). The proposed Ternary method has the highest average sum-rate (3.889 Gbit/s) while the proposed Repartition Method achieves an intermediate performance with a sum-rate of 2.735 Gbit/s. The proposed Ternary approach unpartitions some users who cannot send any packets, allowing the partitioned users to reach a high data rate resulting in a larger sum-rate. On the other hand, the Binary and Repartition methods share the resources more fairly between all users and obtain a lower sum-rate, but at the same time lower outage probabilities.

Execution Time: In Fig. 11, we compare the execution time of all methods for $K = 10$ users. After convergence, the Binary method shows the highest average execution time. During the first episodes, the Repartition Method tries to find the best action and assigns unpartitioned users to one of the two bands, resulting in a longer execution time than the Ternary method and a close one to the Binary approach. After convergence, the Repartition Method predicts a potential binary action that satisfies the stopping criteria.

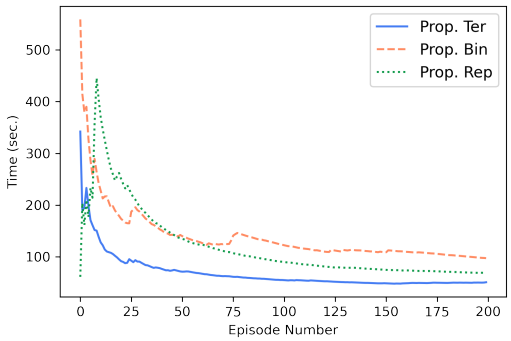


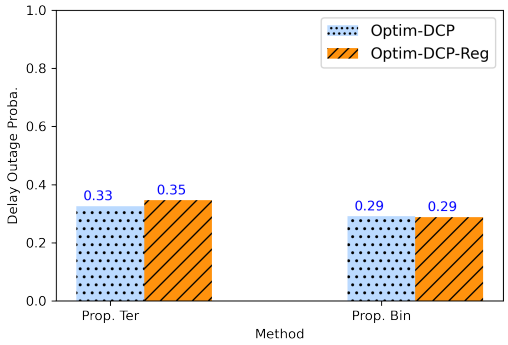
Fig. 11: Average execution time over episodes, $K = 10$

The execution time with the Repartition Method is longer than with the Ternary approach mainly due to the complexity of the predicted actions but it remains faster than that of the Binary approach. Even after convergence, the Binary method takes more time before stopping an episode but as discussed above, this solution outperforms the two other approaches in terms of outage probabilities and fairness among all users.

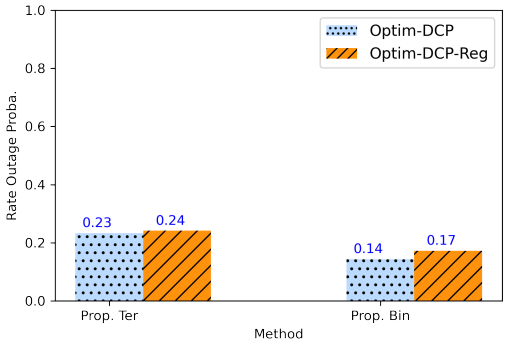
E. Optimization Methods

As shown in Fig. 8, the resource allocation optimization at the sub-schedulers is by far the most time-consuming part of the proposed framework. To reduce the scheduling time, we proposed in Section VI-B a second optimization method by leveraging regularized DCP. In the following evaluations, as the Repartition algorithm takes more random actions than the Ternary and Binary methods, we chose to compare only the Ternary and Binary approaches to avoid the randomness of the Repartition Method and to have a more accurate comparison between the two optimization methods. Furthermore, we fixed A_j as the identity matrix and $\rho_j = \frac{1}{j+1}$, therefore we respect all the required assumptions for convergence. We set the optimal tolerance as $\delta = 0.01$.

Fig. 12 illustrates the results obtained by using *Optim-DCP-Reg* for $K = 10$ users. As we can see, the average outage probabilities over users are similar for both optimization methods. Similarly to *Optim-DCP*, the Ternary approach gets a higher average sum-rate than the Binary method with *Optim-DCP-Reg*, which can be explained by the equity between users obtained by the Binary method, resulting into a lower sum-rate. For all cases, *Optim-DCP-Reg* is outperformed by *Optim-DCP* in terms of sum-rate. The proposed Binary approach achieves a sum-rate 20% lower with *Optim-DCP-Reg* (1.656 Gbit/s) than with *Optim-DCP* and the proposed Ternary approach shows a sum-rate 45% lower with *Optim-DCP-Reg* (2.171 Gbit/s). Nevertheless, despite



(a) Average Delay Outage for $K = 10$

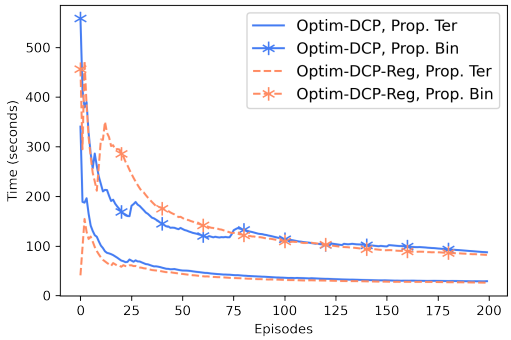


(b) Average Rate Outage for $K = 10$

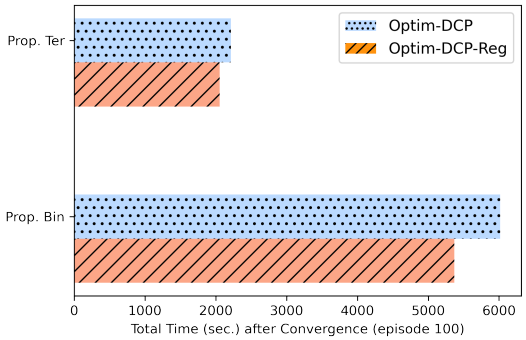
Fig. 12: Outage Probability Performance for $K = 10$

this sum-rate reduction, this optimization algorithm succeeds in meeting the QoS constraints requested by the users, similarly to the first optimization method.

Finally, we show in Fig. 13 the execution time of sub-schedulers for $K = 10$ users. We can clearly see that the second optimization method is faster than the first one for all the proposed methods, especially for the Binary approach where the optimization induces higher complexity. In Fig. 13(a), we illustrate the average execution time for the whole simulation. We can observe that the execution time per episode required by the sub-schedulers for the Binary method is significantly higher than that for Ternary, for both optimization methods. There is no clear difference between the two optimization methods in this figure due to the random actions of the learning algorithm. In Fig. 13(b), we plot the execution time after convergence for both optimization methods. After convergence, the random part of the DQN is reduced and we can compare more easily the optimization algorithms. *Optim-DCP-Reg* is clearly faster than *Optim-DCP*, especially for the Binary Partitioner where the optimization problem is more complex.



(a) Average execution time, $K = 10$



(b) Execution Time after convergence, $K = 10$

Fig. 13: Execution Time of the Sub-Schedulers, $K = 10$

VIII. Conclusion

In this paper, we investigated the resource allocation problem for short-packet communications over the mmWave and Sub-6 GHz frequencies. We have formulated our problem as a sum-rate maximization under users' QoS requirements expressed in terms of delay and rate while including reliability (PER) requirements. To solve it, we proposed a two-stage resolution framework based on a DRL approach that combines the advantages of optimization and learning methods. In particular, the proposed DRL-based Partitioner fully integrated users' QoS requirements and varying LoS situation in the partitioning decision while the sub-schedulers optimize the RBs and beams allocations. We designed different approaches for the DRL Partitioner's action-space, as well as the resource allocation optimization in order to find a trade-off between outage probabilities, sum-rate and execution time. Through extensive simulations, we have shown the advantages of our proposed methods compared to baseline partitioning methods.

In future works, we will extend our framework to a distributed one to cope with practical impairments pertaining to imperfect or outdated feedback information, and to handle massive device connectivity. Additionally, the sub-schedulers will be developed to optimize beamforming parameters under highly mobile environments.

References

[1] I. F. Akyildiz, A. Kak, and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, 2020.

[2] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[3] O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Integrated Millimeter Wave and Sub-6 GHz Wireless Networks: A Roadmap for Joint Mobile Broadband and Ultra-Reliable Low-Latency Communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 109–115, Apr. 2019.

[4] G. Durisi, T. Koch, and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication with Short Packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.

[5] Y. Polyanskiy, "Channel coding: Non-asymptotic fundamental limits." Princeton University, 2010.

[6] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, "Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges," *IEEE Vehicular Technology Mag.*, vol. 14, no. 2, pp. 44–52, 2019.

[7] A. T. Z. Kasgari and W. Saad, "Model-Free Ultra Reliable Low Latency Communication (URLLC): A Deep Reinforcement Learning Framework," *in IEEE ICC*, pp. 1–6, May 2019.

[8] Y. Huang, S. Li, C. Li, Y. T. Hou, and W. Lou, "A Deep Reinforcement Learning-based Approach to Dynamic eMBB/URLLC Multiplexing in 5G NR," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6439–6456, 2020.

[9] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, 2021.

[10] K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju, and B. Shim, "Deep Reinforcement Learning-based Network Slicing for Beyond 5G," *IEEE Access*, vol. 10, pp. 7384–7395, 2022.

[11] F. Qasmi, M. Shehab, H. Alves, and M. Latva-aho, "Effective Energy Efficiency and Statistical QoS Provisioning under Markovian Arrivals and Finite Blocklength Regime," *IEEE Internet of Things Journal (Early Access)*, 2022.

[12] Y. Prathyusha and T.-L. Sheu, "Coordinated Resource Allocations for eMBB and URLLC in 5G Communication Networks," *IEEE Transactions on Vehicular Technology (Early Access)*, 2022.

[13] Y. Zhao, X. Chi, L. Qian, Y. Zhu, and F. Hou, "Resource Allocation and Slicing Puncture in Cellular Networks with eMBB and URLLC Terminals Co-Existence," *IEEE Internet of Things Journal (Early Access)*, 2022.

[14] O. Semiari, W. Saad, and M. Bennis, "Joint Millimeter Wave and Microwave Resources Allocation in Cellular Networks With Dual-Mode Base Stations," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 7, pp. 4802–4816, Jul. 2017.

[15] C. Chaieb, Z. Mlika, F. Abdelkefi, and W. Ajib, "On the Optimization of User Association and Resource Allocation in HetNets with mm-Wave Base Stations," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3957–3967, 2020.

[16] S. Leblanc and M. Kaneko, "Deep Learning-based Sub-6GHz/mmWave Partitioning for Short-Packet Communications," *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2021.

[17] R. Ismayilov, B. Holfeld, R. L. G. Cavalcante, and M. Kaneko, "Power and Beam Optimization for Uplink Millimeter-Wave Hotspot Communication Systems," *IEEE Wireless Communications and Networking Conference*, pp. 1–7, Apr. 2019.

[18] W. R. Ghanem, V. Jamali, Y. Sun, and R. Schober, "Resource Allocation for Multi-User Downlink URLLC-OFDMA Systems," *IEEE International Conference on Communications Workshops*, pp. 1–6, 2019.

[19] C. Stadler, X. Flamm, T. Gruber, A. Djanatliev, R. German, and D. Eckhoff, "A Stochastic V2V LOS/NLOS Model Using Neural Networks for Hardware-In-The-Loop Testing," *2017 IEEE Vehicular Networking Conference*, pp. 195–202, 2017.

[20] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[21] Q. T. Dinh and M. Diehl, "Local Convergence of Sequential Convex Programming for Nonconvex Optimization," *Recent Advances in Optimization and its Applications in Engineering*, pp. 93–102, 2010.

[22] T. Pham Dinh and H. A. Le Thi, "Recent Advances in DC Programming and DCA," *Transactions on computational intelligence XIII*, pp. 1–37, 2014.

[23] C. Stadler *et al.*, "A Line-of-Sight Probability Model for VANETs," *International Wireless Communications and Mobile Computing Conference*, pp. 466–471, 2017.