# A Reinforcement Learning Agent for Mixed-Numerology Interference-Aware Slice Spectrum Allocation with non-Deterministic and Deterministic Traffic[⋆]

Marco Zambianco[a,*], Giacomo Verticale[a]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italy*

## Abstract

5G RAN slicing is an essential tool to support the simultaneous coexistence of enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (URLLC) network slices on a shared mixed-numerology physical layer. Moreover, due to recent advance of the private network paradigm, RAN slicing assumes a central role to provide dedicated radio coverage to industry 4.0 applications as a standalone RAN. Unlike the stochastic traffic behavior characterizing URLLC slices in classical mobile networks, industrial networks support URLLC services with deterministic and periodic traffic patterns. Based on this alternative network characterization, we design a deep reinforcement learning (DRL) agent that simultaneously provides a spectrum allocation fulfilling the eMBB and URLLC service requirements and mitigates the inter-numerology interference (INI). Furthermore, by exploiting the information about the deterministic traffic patterns, we specialize the agent reward function to improve the spectrum allocation effectiveness for URLLC slices deployed in industrial environments. We assess the agent performance with respect to resource allocation schemes that are INI agnostic. Results reveal that the proposed solution outperforms the benchmark schemes in terms of service provisioning performance in both network scenarios (e.g. mobile and industrial) and showcase the benefit of INI mitigation.

*Keywords:* Network slicing, Reinforcement learning, Interference

## 1. Introduction

5G radio access networks (RAN) support under a unified radio interface a plethora of different user applications that are characterized by heterogeneous service requirements. In detail, 5G applications can be matched into three main categories named as enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communication (mMTC) services. Each group is characterized by different values of data rate, latency and reliability.

In this work, we focus our analysis on eMBB services, which provide a high data rate communication, and on URLLC services, that ensure a real-time communication to applications requiring low communication delay. Leveraging the flexibility provided by RAN slicing, eMBB and URLLC services are multiplexed as logical networks on the same physical RAN infrastructure. The advantage of this architecture makes it possible to independently customize each logical network, also refereed as network slice, according to the specific service level requirements (SLA) of eMBB and URLLC users [1].

The performance analysis of coexisting eMBB and URLLC slices has been normally pursued under the typical network scenario of mobile networks. Specifically, eMBB services such as high-definition video streaming and augmented reality applications are characterized by a continuous transmission of data packets that are required to sustain the related throughput specifications. Conversely, URLLC services such as mission critical applications and vehicular-to-vehicular communications are characterized by a bursty transmission of a variable number of small-sized packets. However, the recent focus on 5G private networks for industry 4.0 applications has introduced an alternative network scenario which is disjoint from the classical mobile network one [2]. As a matter of fact, private networks are deployed to offer dedicated radio coverage to the production activities in industrial environments like factories and warehouse. The advantage of this business model, which makes it possible to separate mobile users from industrial users into physically different RANs, increases the flexibility capabilities of network slicing since the latter can be employed to tailor the network performance exclusively based on industrial applications [3]. Example of such services are robots remote control, machine coordination monitoring and automated vehicle driving.

The 3GPP standardization body has recently profiled the service communication requirements in the automation domain and has analyzed an effective integration of their functionalities within the 5G RAN. The main feature differentiating these applications from the URLLC

[⋆] This research work has been supported by project BASE5G (project id: 1155850) funded by Regione Lombardia within the framework POR FESR 2014-2020.

[*] Corresponding author

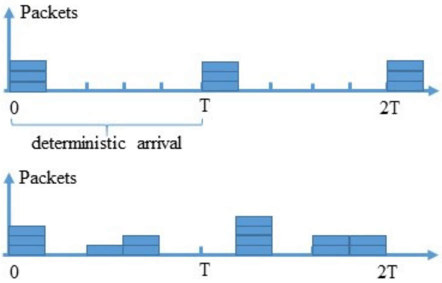

Figure 1: Examples of deterministic packet arrival rate and stochastic packet arrival rate in different time windows.

services deployed in classical mobile network is the statistics of the generated data traffic. In detail, a large percentage of URLLC industrial applications are characterized by deterministic and periodic traffic patterns [4]. In Fig. 1, we schematically depict this concept and we compare it to a stochastic packet arrival rate. The periodic nature of the industrial traffic derives from the operations performed by the various devices, involved in the production line, that require stringent communication delay as well as a constant bit-rate in order to enable their remote control and/or activity monitoring [5]. In this regard, the 5G Alliance for Connected Industries and Automation (5G-ACIA), which is a consortium investigating the employment of 5G technologies to support the industrial production pipeline, has defined practical industrial use-case scenarios that require such traffic type in order guarantee the production activity reliability [6]. Consequently, given this unprecedented usage of RANs for industrial applications, it is important to design communications algorithms specifically suited for industrial environments to fully benefit from the 5G technology.

Regardless of the considered RAN deployment scenario, the low latency requirements of time-sensitive applications are efficiently addressed by mixed-numerologies access schemes. Unlike the conventional orthogonal frequency division multiplexing (OFDM) scheme that employs a homogeneous subcarrier spacing for each symbol (in other words, OFDM can be considered as a single-numerology scheme), a mixed-numerology scheme supports a variable subcarrier spacing within the same transmission symbol, where each subcarrier spacing value denotes a different numerology [7]. On one hand, the benefit provided by such flexible transmission frame structure allows to tailor the subcarrier spacing according to the application requirements. For example, the tight delay requirements of URLLC users are effectively satisfied by a wider subcarrier spacing which reduces the packet transmissions time, whereas a lower subcarrier spacing is suited to support a high data rate under different propagation scenarios.

Following the observation about the multiplexing of eMBB and URLLC services in different network scenarios as well as the necessity of INI mitigation techniques, we address the problem of the design of an INI-aware slice spectrum allocation policy for URLLC and eMBB network slices. Moreover, differently from our previous work in this topic [8], we extend the proposed solution for URLLC slices deployed in industrial networks that we characterize according to the deterministic traffic model previously introduced. We solve the considered problem leveraging deep reinforcement learning (DRL). The benefit derived by employing this resolution scheme is twofold. First, the generalization capabilities of this scheme allow to automatically infer the relationship between INI and the related wireless channel fading fluctuations of each user. Such information can be exploited to boost the system data rate thanks to a smarter slice spectrum allocation that limits the INI. Second, the deterministic nature of the periodic traffic patterns in industrial URLLC slices makes the network behavior more predictable, which benefits the agent learning effectiveness. The main contributions of this work are as follows:

- We define an integer non-convex optimization problem that maximizes the cumulative throughput of the eMBB and URLLC users subject to the related SLA requirements, where the INI power dynamic is analytically embedded within the objective function formulation.

- Due to the prohibitive computational complexity derived from the combinatorial nature of the considered allocation problem, we employ deep reinforcement learning to design a multi-branch agent, based on Branching Dueling Q-networks (BDQ), which allows an efficient environment exploration. The designed agent provides an INI-aware spectrum allocation policy that approximates the original problem formulation.

- We improve the agent reward function design in order to boost the URLLC slice performance when deployed in industrial scenarios. This alternative formulation allows the agent to compute a more effective spectrum allocation by exploiting the deterministic traffic information that would not be available in the mobile network scenario.

- We compare the agent performance against different resource allocation algorithms that do not account for the INI. Moreover, we also analyze the results provided by the agent when the URLLC traffic statistic is modelled according to mobile and industrial networks.

The remainder of the paper is structured as follows. We discuss the related work in Section 2. We describe the system model in Section 3. We formalize the optimization problem as well as the agent environment formulation in Section 4. We present the multi-branch agent architecture in Section 5. We analyze the performance in Section 6. Finally, the conclusion is drawn in Section 7.

## 2. Related Work

Most of the research activity has proposed spectrum allocation algorithms to fulfil the service requirements of eMBB and URLLC slices. Similarly, many works have designed signal processing techniques for the INI minimization. In our opinion, a joint analysis of the mutual impact of these two problems has received a limited attention. Similarly, the majority of the research has focused on RAN slicing schemes that do not consider the traffic characteristics of industrial scenarios. According to this observation, we review the RAN slicing solutions in the context mobile network and industrial networks. A general overview of the main challenges related to the coexistence eMBB and URLLC services can be found in [9]. Similarly, the fundamental concepts of inter-numerology interference are introduced in [10].

### *2.1. RAN slicing in mobile networks*

The authors of [11] formulate a sum-rate maximization problem that enforces latency and minimum data rate constraints on a OFDMA physical layer with the incorporation of adaptive modulation and coding schemes. The authors of [12] present a mini-slot based resource allocation scheme to augment the URLLC service reliability while also ensuring a certain degree of fairness among eMBB users. Differently from [11] [12], we extend the eMBB and URLLC multiplexing analysis for a mixed-numerology access scheme. The authors of [13] design a DRL agent to allocate spectrum resource with a flexible numerology structure in order to accommodate URLLC and eMBB users. The authors of [14] propose a preemption puncturing scheme based on DRL that minimizes the performance degradation of eMBB users when punctured by URLLC traffic. Compared to the previous works, [13] and [14] propose solutions that consider different numerologies for the URLLC and eMBB services. However, the INI impact on the performance of the presented schemes is not covered. Conversely, we analytically consider the INI dynamic to increase the SLA accommodation reliability by designing an INI-aware agent based DRL. The authors of [15] propose an INI-aware mixed-numerology resource allocation scheme that analytically accounts for the INI in order to maximize the user data rate. However, they only consider the INI generated by higher numerologies over the lower ones. Moreover, their approach does not differentiate between eMBB and URLLC services. Instead, we consider the INI generated by each numerology in order to better model the performance degradation affecting eMBB and URLLC users.

### *2.2. RAN slicing in industrial networks*

The authors of [16] propose an allocation scheme of the radio resources to support deterministic traffic required by different time sensitive applications. The authors of [17] design a spectrum reservation scheme for industrial URLLC slices that preemptively allocates the resource based on the a-priori knowledge of the number of incoming packets. Authors of [18] propose a novel latency descriptor that identifies the number of transmission slots to reserve in order to satisfy the latency requirements of industrial applications. Authors of [19] design a fading correlation-aware resource allocation scheme to boost the reliability of time-critical applications in industry 4.0 scenarios. Differently from these works that employ optimization based solutions, we showcase the performance gain provided by a spectrum allocation scheme based on DRL when a the URLLC slice is characterized by a deterministic traffic pattern.

## 3. System model

We consider a RAN that provides radio coverage to a set $U_e$ of eMBB users, and a set $U_l$ of URLLC users. Let $U = U_e \cup U_l$ be the set of all the users. We characterize a *mobile network scenario* with a continuous transmission of data packets (e.g. full-buffer model) for the eMBB slice and with of a bursty transmission of small packets modelled as Poisson arrival process for the URLLC slice. Differently, we characterize an *industrial network scenario* by replacing the stochastic traffic behavior of the URLLC slice with a periodic and deterministic packet arrival rate. In other words, the number of incoming packets is fixed and it is scheduled in known transmission intervals. Note that this traffic type has been supported by 3GPP with the introduction of the Time-Sensitive Communication Assistance Information (TSCAI) [20]. The latter is a network descriptor that contains the details about the packet size, number of arrived packets and related transmission periodicity and it can be employed by the RAN scheduler to optimize the resource allocation for the URLLC users. Following this observation, we assume that TSCAI is available when we consider the spectrum policy design for the industrial network scenario.

The physical layer employs a mixed-numerology access scheme where each slice is multiplexed with a different numerology dictating a fixed subcarrier spacing. Following the 3GPP specification for NR, the available numerologies are $\Delta f_i = 15 \cdot 2^i$ kHz, $i \in \{0, .., 4\}$ [21]. In our model, we assume that $\Delta f_i^{(\mathrm{URLLC})} \geq \Delta f_i^{(\mathrm{eMBB})}$ due to the fact that higher numerology values are better suited to support a low latency communication.

The available spectrum is composed by $K$ non-overlapping bandwidth parts (BWP) of bandwidth $W$, which are accessed by the eMBB and URLLC users. A BWP is defined as a fixed number of contiguous resource blocks (RB), each one identifying a time transmission interval (TTI) of 1-ms duration and a frequency range of 12 subcarriers with subcarrier spacing depending on the numerology type [22]. The network owner (NO), which manages the physical network infrastructure resources, assigns a suitable number of BWPs to the users of each slice in order to accommodate the related SLA requirements. A scheme of the considered RAN architecture is shown in Fig. 2.

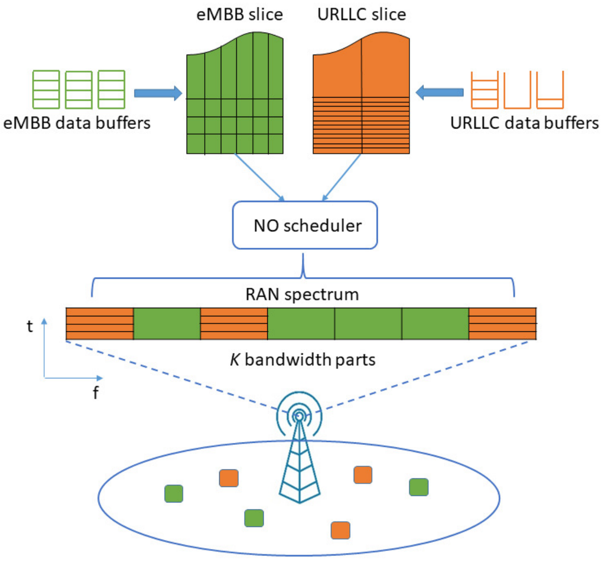

Figure 2: RAN slicing architecture for eMBB and URLLC services with mixed-numerologies. Based on the multiplexed service, each BWP is defined by a different numerology.

Every BWP is modelled as a flat-fading channel whose gain can be perfectly retrieved by the NO (in other words, we assume perfect channel state information (CSI)). In detail, such information is reported by the various users that estimate the subchannel gain in every BWP. Analytically, we compute the subchannel gain power of the generic user $u$ at time slot $t$ over the $k$-th BWP as $g^u_{t,k} = h^u_{t,k}\alpha^u_t$, where $h^u_{t,k}$ is an exponential random variable with unit mean that models the small-scale fading, whereas $\alpha^u_t$ expresses the channel attenuation derived from the large-scale fading consisting of path loss and shadowing.

Due to the loss of the orthogonality between subbands of different numerologies, the simultaneous allocation of eMBB and URLLC users over contiguous spectrum regions generates INI. The INI power dynamic can be analytically modelled relying on the expression proposed by authors of [23]. The latter provides the INI power value when two subbands of different numerologies are contiguously allocated. We generalize this result to compute the INI power for multiple mixed-numerologies subbands based on the BWP structure previously introduced. Practically, we can compute the INI power affecting each BWP by progressively summing each single contribution of the remaining BWPs. In detail, let $k$ and $k'$ be two different BWP, user $u$ (either belonging to eMBB or URLLC slices) is allocated to BWP $k$ using a numerology with subcarrier spacing $\Delta f_i$. Assume that another user is allocated to BWP $k'$ and employs a numerology with subcarrier spacing $\Delta f_{i'}$, then the INI power affecting user $u$ over BWP $k$ during time slot $t$ can be approximated as, if $\Delta f_i < \Delta f_{i'}$,

$$I^u_t(k,k') \approx \frac{P_{t,k'}}{N_{k'}} \sum_{z=1}^{N_k} \sum_{v=1}^{N_{k'}} \frac{g^u_{t,k'}}{N_{k'}N_k} \left[ \left| \frac{\sin[\frac{\pi}{N_k} w(z,v)\xi N^{(T)}_{k'}]}{\sin(\frac{\pi}{N_k} w(z,v))} \right|^2 + \xi \left| \frac{\sin[\frac{\pi}{N_k} w(z,v) N^{(T)}_{k'}]}{\sin[\frac{\pi}{N_k} w(z,v)]} \right|^2 \right], \tag{1}$$

otherwise, if $\Delta f_i > \Delta f_{i'}$, as

$$I^u_t(k,k') \approx \frac{P_{t,k'}}{N_{k'}} \sum_{z=1}^{N_k} \sum_{v=1}^{N_{k'}} \frac{g^u_{t,k'}}{N_{k'}N_k} \left| \frac{\sin[\frac{\pi}{N_{k'}} w(z,v) N_k]}{\sin[\frac{\pi}{N_{k'}} w(z,v)]} \right|^2 \tag{2}$$

where $N_k = W/\Delta f_i$ corresponds to the number of subcarriers of BWP $k$, $P_{t,k'}$ is the allocated transmission power, $N^{(T)}_k = N_k + N^{CP}_k$ denotes the total number of subcarriers considering also the number of subcarriers employed as cyclic-prefix $N^{CP}_k$, $\xi = \lfloor N_k/N^T_{k'} \rfloor$ is the number of OFMD symbols of the wider numerology that are transmitted within the time transmission window of one OFDM symbol of the smaller numerology, and $w(z,v)$ is the spectral distance between subcarriers of different numerologies and it is computed as the total number of subcarriers separating subcarrier $z$ from subcarrier $v$. From (1) and (2), we observe that both a wider numerology gap and subchannel power gap increase the resulting INI power over the surrounding BWPs.

We define the allocator indicator function $x^u_{t,k}$ that assumes value 1 when BWP $k$ is assigned to user $u$ at time slot $t$ or value 0, otherwise. Consequently, we compute the data rate $r^u_{t,k}$ achieved by user $u$ in time slot $t$ on BWP $k$ as

$$r^u_{t,k} = \mathrm{W} \cdot \log_2(1 + \gamma^u_{t,k}(x^u_{t,k})), \tag{3}$$

where $\gamma^u_{t,k}(x^u_{t,k})$ is the signal-to-interference-noise ratio (SINR) and it is computed as

$$\gamma^u_{t,k}(x^u_{t,k}) = \frac{P_{t,k} g^u_{t,k}}{\sigma^2_w + \sum\limits_{k' \neq k} x^u_{t,k'} I^u_t(k,k')}, \tag{4}$$

with $\sigma^2_w$ indicating the white Gaussian noise power. From (3), we can observe that data rate of the URLLC and eMBB users are mutually affected by the INI power generated by the related BWP multiplexing.

## 4. Problem Formulation

The INI mitigation improves the service provisioning performance by ensuring a higher user data rate (in other words, it enforces the inter-slice isolation). This goal can be achieved by a suitable multiplexing of the spectrum resource. Following this observation, we define an optimization problem that accounts for the INI and it maximizes the cumulative throughput of eMBB and URLLC users.

### 4.1. Optimal INI-aware resource allocation

We consider a spectrum allocation over $T$ time slots. During such time window, we indicate as $r_e$ the minimum per-user data rate requirements of the eMBB slice. Similarly, URLLC SLA requirements are fulfilled when the packets of each user are delivered within $T$ time slots. We indicate the number of pending packets in each user buffer as $b_u$ and the bit packet size as $L$. Based on these SLA definitions, we formulate the optimization problem as

$$\max_{\mathbf{x}} \sum_{u=1}^{U} \sum_{t=1}^{T} \sum_{k=1}^{K} x_{t,k}^u r_{t,k}^u \tag{5}$$

subject to

$$\sum_{t=1}^{T} \sum_{k=1}^{K} x_{t,k}^u r_{t,k}^u \geq r_e \quad \forall u \in U_e \tag{6}$$

$$\sum_{t=1}^{T} \sum_{k=1}^{K} x_{t,k}^u r_{t,k}^u \geq \frac{b_t^u \cdot L}{T} \quad \forall u \in U_l \tag{7}$$

$$\sum_{u=1}^{U} x_{t,k}^u \leq 1 \quad \forall t \in T, \forall k \in K \tag{8}$$

$$x_{t,k}^u \in \{0,1\} \leq 1 \quad \forall t \in T, \forall k \in K, \forall u \in U. \tag{9}$$

The objective function (5) maximizes the cumulative data rate achieved by both the eMBB and URLLC within the considered time windows. Note that the data rate formulation embeds the INI expression so its maximization indirectly mitigates the INI effect. Constraints (6) and (7) implement the data rate and latency requirements for the eMBB and URLLC service, respectively. Finally, constraints (8) and (9) enforce the solution feasibility by ensuring that BWPs are uniquely allocated to a single user only and that the allocation indicator function can assume binary values only, respectively.

The non-convexity of the objective function (5), which can be considered as a difference of convex functions, makes the proposed optimization problem NP-hard [24]. Moreover, the computational complexity is further exacerbated by the combinatorial nature of the considered allocation problem that has an integer optimization variable. Consequently, it is not practical to compute a BWP allocation employing classical optimization-based resolution schemes due to the strict latency requirements needed at the physical layer. Moreover, the proposed problem requires the a-priori knowledge of both the subchannel gains of each user and the number of incoming URLLC packets over the $T$ time steps. Note that the latter requirement can be removed under the deterministic traffic assumption since the packet arrival rate is provided by the TC-SAI descriptor. Nonetheless, generally, the solution computation also involves a channel and/or traffic prediction scheme, which is not feasible in most real-world scenarios. To overcome these challenges, we propose an alternative resolution scheme, based on deep reinforcement learning, that relies on a model-free formulation of the considered problem to compute a suitable spectrum allocation policy. In other words, our goal is to design a DRL agent that can approximate the solution of (5). In the next section, we describe the agent environment as a Markov Decision Process (MDP) that is employed to learn the allocation policy.

### 4.2. Agent environment design

Reinforcement learning (RL) makes it possible to compute the optimal policy for problems formulated as MPDs without any prior knowledge on its exact mathematical model [25]. Formally, an MDP is described as the 4-tuple

$$\{S, A, R(s,a), p(s'|s,a)\},$$

where $S$ indicates the set of states, $A$ expresses the set of actions, $R(s,a)$ is a reward function that depends on both the current visited state $s$ and the selected action $a$, and the transition probability distribution $p(s'|s,a)$. The latter models the probability to transition toward state $s'$ in time-slot $t+1$ when selecting action $a$ in state $s$ during time slot $t$. Unlike classical dynamic programming algorithms, RL schemes are able to compute the optimal policy defined as $\pi\colon S \rightarrow A$ without the knowledge of $p(s'|s,a)$. Instead, throughout an iterative trial and error process, they learn the optimal policy that maximizes the cumulative discounted reward obtainable from the environment over a fixed time horizon. Following the MPD definition, we describe the state space, the action space and the reward functions (one for the mobile network scenario and one for industrial scenario) as follows.

**State space.** The state space is composed by the network parameters that allow the agent to assess the effectiveness of BWP allocation in terms of SLA accommodation. Specifically, we denote the buffer status of the URLLC users at time $t$ as $\mathbf{b}_t^l = \{b_t^u\}$ with $u \in U_l$. We update the number of queued packets as

$$b_{t+1}^u = \max((b_t^u - r_{t,k}^u/L), 0) + l_t^u \tag{10}$$

where $l_t^u$ denotes the number of packets arrived in time slot $t$ for user $u$. Note that in the industrial network scenario, we assume that $l_t^u = Z$ if $t=1$ and $l_t^u = 0 \quad$ if $\quad 2 \leq t \leq T$ (in other words, the packets arrival rate is of $Z$ packets every $T$ time steps). With regard to the eMBB slice, we track the number of bits transmitted to each eMBB user up to time slot $t$, i.e. $\mathbf{r}_t^e = \{\hat{r}_t^u\}$, with $u \in U_e$. We formalize the state space as

$$S_t = \{\mathbf{b}_t^l, \mathbf{r}_t^e, g_{t,k}^u, r_{t-1,k}^u\}. \tag{11}$$

The subchannel gain coefficients $g_{t,k}^u$ of each BWP are updated according to the CSI reporting periodicity which dictates the frequency of the BWP quality estimation performed by the users. We remark that the CSI reporting granularity does not affect the training performance as $g_{t,k}^u$

is used by the agent to infer the system capacity in the considered transmission slot and to accommodate the users service requirements accordingly. Instead, the correlation between CSI values across consecutive time steps can impact the agent learning efficiency. Specifically, the observation of correlated subchannel gains can be exploited by the agent to predict the service performance in future time steps and thus to provide more effective BWP allocations.

In addition to the aforementioned network parameters, we also include the per-user data rate achieved in the previous time-slot, $r^u_{t-1,k}$. This information is beneficial for two main reasons: i) the data rate expression (3) already accounts for the INI power, thus the explicit usage of the complex formulations (1) and (2) is unnecessary ii) the agent can better correlate the impact of the subchannel gain on the INI power affecting the user data rate by observing the related performance in the previous time step.

**Action space.** The action space is composed by every possible BWP allocation. We formally define an action as a vector $\mathbf{a}_t$ of $K$ elements where each coordinate $a_k$ indicates the index of the user scheduled for the $k$-th BWP in time slot $t$. Specifically, eMBB users are identified by $a_k \in [1, ..., U_e]$, whereas the URLLC users are identified by $a_k \in [U_e + 1, .., U_e + U_l]$. Analytically, each action can be written as

$$\mathbf{a}_t = \{a_1, ..., a_K\}. \tag{12}$$

To simplify the notation, we neglect the index $t$ from each coordinate $a_k$ since the time step information is already included in the general action expression $\mathbf{a}_t$. The agent learning efficiency can be negatively impacted by the action space dimension, that is equal to $U^K$, since it slows the environment exploration. We address this issue by designing a simple metric that allows to discard unfeasible actions. We are going to discuss this approach in the next section.

**Reward function.** We design the reward function by leveraging (5)-(9). The idea is to characterize the same optimization goal of such formulation using an alternative expression that can be employed by the agent to learn a suitable spectrum policy. In particular, we propose two reward functions, one for the mobile network case and one for the industrial network case.

*Mobile network scenario.* Due to the stochastic nature of the URLLC data rate that denies a reliable prediction of future incoming packets, we express the reward function following this design choice. One one hand, the agent should maximize the data rate of both slices in each time slot $t$. One the other hand, it should satisfy the minimum service requirements of each slice as soon as possible in order to anticipate unexpected burst of URLLC packets that can affect the overall service quality. We formally define the reward as

$$R_t = \sum_{u \in U} \sum_{k \in K} r^u_t - \sum_{u \in U} p(u), \tag{13}$$

where

$$p_t(u) = \begin{cases} c_e & \text{if} \quad u \in U_e \wedge \hat{r}^u_t > 0 \\ c_l & \text{if} \quad u \in U_l \wedge b^u_t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $c_e, c_l \geq 0$ allows to differentiate the penalty given to unsatisfied eMBB and URLLC users, respectively. The value difference between the two penalties allows to find a trade-off between meeting the requirements of the URLLC or eMBB users, hence the optimal values for $c_e$ and $c_l$ should be tuned for the considered configuration scenario as well as the environment propagation characteristics. We remark that it is important to ensure a comparable magnitude between such parameters to prevent the agent from starving of one service or the other.

The first term in (13) is equal to the aggregate data rate of URLLC and eMBB users computed according to (5) using the BWP allocation expressed by the selected action $\mathbf{a}_t$. The second term, $p(u)$, enforces constraints (6)-(7) in the form of a penalty function that lowers the reward value when the SLA requirements of user $u$ are not satisfied in the current time slot. Consequently, this term encourages the agent to compute a BWP allocation that fulfills the minimum SLA requirements in few time steps within the considered time window $T$.

*Industrial network scenario.* The deterministic structure of the traffic in the URLLC slice ensures the a-priori knowledge of the packet arrival rate periodicity. As a matter of fact, the traffic can be perfectly predicted thanks to the TSCAI availability. Relying on this information, we can design a more flexible reward function. A shortcoming of the reward function (13) is that it implicitly assumes that the optimal BWP allocation (i.e. the solution of (5)-(9)) is the one that satisfies the minimum service requirements within a limited number of time slots due to the unpredictable behavior of both the wireless channel and the traffic statistics as mentioned earlier. This strategy could lead to a suboptimal allocation. When considering the industrial network scenario, we can overcome this limitation by designing the reward function as

$$R_t = \sum_{u \in U} \sum_{k \in K} r^u_t - \sum_{u \in U} \hat{p}(u). \tag{15}$$

where, if $t < T$,

$$\hat{p}(u) = 0,$$

otherwise

$$\hat{p}_t(u) = \begin{cases} 0 \quad \text{if} \quad \sum_{u \in U} \hat{r}^u_T = 0 \wedge \sum_{u \in U} b^u_T = 0 \\ \sum_{t \in T} \sum_{k \in K} r^u_t \quad \text{otherwise.} \end{cases} \tag{16}$$

As similarly proposed in previous the case, the first term of (15) provides a reward that is proportional to the system throughput. However, instead of penalizing the agent at each time step, the agent obtains a negative reward (16), which is equal to the total accumulated throughput, only in the last time step $T$ when constraints (6)-(7) are not satisfied. Such penalty erases any throughput gains obtained by the agent within the considered time window, thus leading to a net reward equals to zero.

This reward design approach allows the agent to autonomously decide within how many time steps to fulfil the service requirements since it is not affected by intermediate penalties. This feature increases the allocation flexibility thanks to a wider BWP allocation pool. For example, for a given a wireless channel configuration, (15) allows to approximate those optimal solutions characterized by the transmission of URLLC packets in the last time step. This strategy is possible due to the fact that no additional URLLC packets are expected to arrive until the subsequent time window, hence the agent can delay their transmission without incurring into the risk of facing burst of packets between different time steps.

## 5. Multi-branch Agent overview

We leverage the multi-branch architecture of Branching Dueling Q-Networks (BDQ) to design the agent [26]. Unlike deep Q-networks (DQN), this architecture supports discrete multi-dimensional action spaces and can be considered as an extension of DQN. The latter is hindered by large action spaces which make the agent exploration inefficient. In detail, DQN computes the optimal policy, $\pi^*$, by selecting the action maximizing the estimated Q-function value, $Q(s, a)$, that provides an approximation of expected obtainable reward when state $s$ is observed, i.e.

$$\pi^* = \underset{a \in A}{\operatorname{argmax}}\, Q(s, a). \tag{17}$$

Storing each Q-function action-state pair is unpractical in most scenarios, hence a deep neural network (DNN) of weights $\boldsymbol{\theta}$ is employed to simultaneously output the associated Q-function values $Q(s, a)$ for every action $a \in A$ when the input state is $s$. This strategy leverages the DNN generalization capabilities to efficiently encode the state space. However, due to the fact that every action has to be uniquely specified as a single output neuron, the estimation of the Q-function value for each action-state pair becomes unreliable when the number of actions is large, thus leading to poor learning performance. In our problem, this would require the enumeration of the $U^K$ BWP allocations.

### *5.1. BDQ general structure*

In order to overcome the large action space issue, we first provide an intuitive idea of the basic concept of multi-branch agent as follows. We re-formulate the original allocation problem in the form of smaller allocation sub-problems whose solution can be easily computed. Then, each single solution is merged together in order to retrieve the global BWP allocation. Practically, the agent DNN is composed by $K$ branches (i.e. one branch for each BWP), where each branch is in charge of scheduling the $U$ users in that specific BWP. In other words, every network branch can be viewed as a sub-agent having action space $a_k$. The global BWP allocation $\mathbf{a}_t$ is obtained by simply concatenating the different sub-actions $a_k$. We remark that the global BWP allocation performance affects the scheduling policies of various branches as to incentivize their cooperation in maximizing the reward function objective. In this regard, BDQ employs a state-value estimator that is shared between branches as well as a multi-branch dueling layer. Their purpose is to promote the branches cooperation and to increase the agent exploration efficiency, respectively.

**Shared state-value estimator.** This terms is implemented as an additional agent branch and provides the estimation of the state value function, $V(s)$. The latter approximates the obtainable reward that the agent can achieve when the environment state is $s$, i.e.

$$V(s) = \sum_{a \in A} \pi(a|s) Q(s, a). \tag{18}$$

As mentioned earlier, by sharing (18) across the different network branches, it is possible to enforce their coordination due to the fact that the agent can better correlate the impact of the per-branch action on the global reward.

**Multi-branch dueling layer.** To enhance the agent generalization capability during the training phase, the agent exploits the concept Dueling Networks [27]. In details, this network layer speeds up the agent convergence by computing the Q-function in each branch as a combination of the shared state-value function $V(s)$ and the per-branch advantage function $A_k(s, a_k)$. Formally, the Q-function is computed as

$$Q_k(s, a_k) = V(s) + \Big[A_k(s, a_k) - \frac{1}{U} \sum_{a'_k=1}^{U} A_k(s, a'_k)\Big]. \tag{19}$$

The advantage function quantifies the benefit of taking action $a_k$ from branch $k$ in terms of expected reward improvement when the agent observes state $s$. The resulting effect of such alternative Q-function computation increases the agent capability to discriminate between useful actions and inefficient actions providing similar rewards.

### *5.2. Action masking module*

We improve the training performance by including an action masking scheme that allows the agent to discard those actions that do not increase the accumulated reward (from the agent perspective, such actions are considered as unfeasible).

Since the URLLC traffic model do not assume a full-buffer behavior, the buffer queues of the related users can have zero pending packets in some time steps. Consequently, such URLLC users do not provide any reward due the fact that their potential data rate is zero. However, during the training phase, this external information

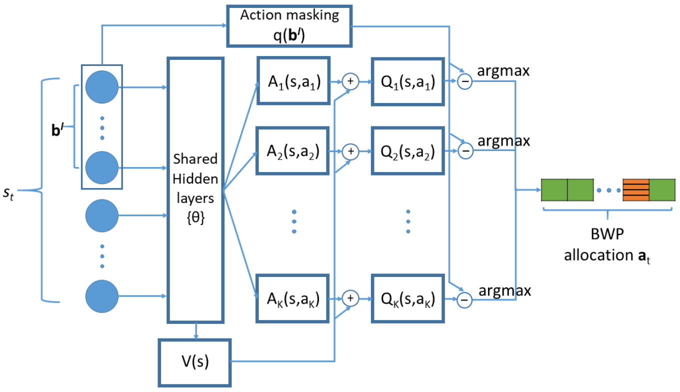

Figure 3: Multi-branch DNN agent architecture. Each branch allocates the $k$-th BWP to one of the $U$ users according to related Q-function value $Q_k(s, a_k)$. The action mask module adds a negative bias to the estimated $Q_k(s, a_k)$ whenever the allocated URLLC user has an empty buffer.

is not available to the agent that has to gradually infer it by correlating a low reward with the allocation of spectrum resources to empty-buffer URLLC users. Such inefficient learning behavior slows the convergence to the optimal policy. To overcome this issue, we artificially lower the Q-function value of empty-buffer URLLC users in order to avoid their selection, i.e.

$$\tilde{Q}_k(s, a_k) = Q_k(s, a_k) - q(b^u_t) \quad \forall u \in U_l, \forall k \in K \tag{20}$$

where

$$q(b^u_t) = \begin{cases} Q & \text{if} \quad b^u_t = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

By choosing a sufficiently high value of $Q$, the agent discards any empty-buffer URLLC user from the BWP allocation computation since the associated Q-function value provides the lowest expected reward in each branch. We schematically depict the general architecture of the described multi-branch agent in Fig. 3.

### *5.3. Training phase*

The objective of the training phase is the computation of the DNN weights $\boldsymbol{\theta}$ that ensure the best approximation of the Q-function. The training is performed on episodes composed by $T$ time steps, where each episode is initialized with a random user distribution over the base station coverage area. In general, it is reasonable to assume a low user mobility in industrial environments since devices operate in limited space regions. Similarly, the same assumption holds for the majority of users in mobile networks. According to these observations, the large-scale fading components (i.e. path loss and shadowing) can be treated as constants for the whole episode duration since the latter is defined by a tight time window of few milliseconds. Moreover, the resulting slow-fading behavior of the channel allows to safely neglect the CSI report delay since the channel quality estimate can be considered valid for a time window that is longer than the episode length. By evaluating different user distributions for each new episode, the agent can generalize the BWP allocation for different propagation scenarios. Indeed, from the agent perspective, every user displacement that provides a substantial modification of the large-scale fading values can be considered as a new episode having that specific user distribution. Since the computation of a suitable parametrization of DNN weights is computationally demanding, such procedure is performed offline. Differently, the testing phase supports the online deployment of the trained agent due to the fact that the BWP allocation is efficiently computed by simply feeding the environment status on the agent DNN and selecting the output neurons based on (17). However, we remark that the computational complexity required to compute each action should satisfy the system scheduling periodicity. In particular, to provide an intuitive evaluation of this requirement, the DNN implementation on a real system should be able to compute the allocation within the transmission window defined by the highest-order numerology service as the latter is characterized by the most stringent transmission interval (in our scenario, the URLLC numerology dictates the BWP computation deadline).

The optimal policy is computed throughout a process that involves environment exploration and exploitation. In this regard, the agent employs an $\epsilon-$greedy policy for the exploration that, with probability $\epsilon$, makes the action selection from each branch as random. More precisely, $\mathbf{a}_t$ is built by randomly selecting $a_k$ from each branch while excluding empty-buffer URLLC users. Otherwise, with probability $1 - \epsilon$, the agent chooses the action that maximizes the aggregate per-branch Q-function value according to the state observation, i.e.

$$\mathbf{a}_t = (\underset{a_1 \in U}{\operatorname{argmax}}\, \tilde{Q}_1(s, a_1), ..., \underset{a_K \in U}{\operatorname{argmax}}\, \tilde{Q}_K(s, a_K)). \tag{22}$$

The experience tuple $(s_t, \mathbf{a}_t, R_t, s_{t+1})$, that is sampled from the environment, is stored in the replay buffer, which records the most recent $N$ experience tuples. At each time step $t$, the agent samples a mini-batch of $D$ experience tuples from the replay buffer. The mini-batch is used to update $\boldsymbol{\theta}$ across the different branches by minimizing the loss function $L(\boldsymbol{\theta})$ defined as

$$L(\boldsymbol{\theta}) = \mathbb{E}_{i \in D}\left[\frac{1}{K}\sum_{k=1}^{K}(y_i - Q_k(s, a_k))^2\right]. \tag{23}$$

In (23), $y_i$ is the target Q-value computed by means of temporal-difference updates across the different branches as

$$y_i = R(s_t, \mathbf{a}_t) + \frac{\delta}{K}\sum_{k \in K} Q'_k(s_{t+1}, \underset{a'_k \in U}{\operatorname{argmax}}\, Q_k(s_{t+1}, a'_k)), \tag{24}$$

where $\delta$ is the discount factor. Note that $Q'_k(\cdot)$ refers to the Q-function value approximated by a second DNN of weights $\boldsymbol{\theta'}$ that are updated every few episodes as $\boldsymbol{\theta'} = \boldsymbol{\theta}$

and it is used to stabilize the Q-function computation convergence. The mini-batch sampling is performed according to the Prioritized Experience Replay procedure, introduced by the authors of [28], that has been adapted for BDQ. Unlike the uniform sampling procedure traditionally employed in the original experience replay algorithm, this scheme selects with higher probability the experience tuples with a relative high value of (23) (in other words, these are the tuples whose Q-function value is poorly approximated by the DNN). Moreover, we highlight that the agent computes Q-function values according to $Q_k(s, a_k)$ instead of the biased version generated by the action masking scheme $\tilde{Q}_k(s, a_k)$. The latter is actually transparent with respect to the Q-function computation procedure, hence the policy convergence is still achieved relying on the original update scheme.

## 6. Results

### 6.1. Simulation setup

Both the agent as well as the network environment have been implemented using MATLAB. We select the network parametrization following two guidelines. On one hand, we ensure a challenging environment configuration in term of action and state space dimension complexity that allows to showcase the benefit of the proposed multi-branch agent architecture in computing an effective solution. On the other hand, we tune the radio parameters as well as the service requirements in order to make sure that the system capacity is fully loaded.

As a network scenario, we considered a base station serving users located at a random distance $d$. The RAN spectrum is divided into 6 BWPs, which are allocated over $T = 4$ time slots (i.e. 4 ms), which is the maximum delay requirement for NR RAN [29], and provides connectivity to $U_e = 4$ eMBB users and $U_l = 4$ users. As a result, this configuration produces an action space composed by $8^6 = 262144$ possible BWP allocations which is an unfeasible number for a classical DQN agent as the latter needs to explore every action multiple times in order to estimate the associated expected reward. Differently, the employed multi-branch architecture can rapidly generalize among actions providing similar rewards, thus reducing the exploration overhead.

The eMBB slice employs a numerology of 15 kHz, whereas the URLLC slice can employ either the 15-kHz or the 60-kHz numerology. By changing the URLLC numerology configuration, we can better assess the benefits of INI mitigation when evaluating the agent performance with respect to a interference-free single-numerology access scheme (both slices have 15-kHz numerology) and with respect to a mixed-numerology access scheme.

We model the mobile network scenario by generating the URLLC traffic statistic according to a Poisson arrival process with arrival rate of 6 packets per TTI and a packet length of 100 bytes. Differently, for the industrial network scenario case, we assume that 24 packets of 100-byte length are generated with a periodicity of 4 ms within the first time step of the time window $T$. In both scenarios, we set the eMBB user data rate requirement as $r_e = 1.25$ Mbit/s. Finally, we assume that the BS transmission power is uniformly allocated among the available BWPs. We summarize the RAN parameters in Table 1.

Table 1: RAN parameters

| | |
|---|---|
| Transmission power | 30 dBm |
| RAN spectrum | 10 MHz |
| Number of BWPs | 6 |
| Coverage radius | 250 m |
| Total number of users | 8 |
| eMBB numerology | 15 kHz |
| URLLC numerology | 15, 60 kHz |
| Fading statistic | Rayleigh |
| Doppler shift | 35 Hz |
| Fading update | 1 ms |
| Path loss model [30] | $36.7 \log_{10} d + 33.05$ (dB) |
| Shadowing std. deviation | 4 dB |
| Noise PSD | $-174$ dBm/Hz |

Table 2: Agent parameters

| | |
|---|---|
| DNN layers | $1024/512/128K$ |
| Learning rate | $10^{-4}$ |
| Discount factor | 0.99 |
| Prioritized sampling | $\alpha_0 = 1$, $\beta_0 = 1$ |
| Experience-replay buffer size | $10^5$ |
| Mini-batch size | 32 |

The DNN agent is made of a total of 3 fully connected hidden layers. In detail, the shared network part that is common to all the branches and it is composed by 2 hidden layers having 1024 and 512 neurons each, whereas each network branch has 1 hidden layer having 256 neurons. The input layer size corresponds to the state space dimension $S_t$, which is equal to $U(K + 2)$. Similarly, the output layer size corresponds to the action space dimension $\mathbf{a}_t$, which is equal to $K \cdot U$ (i.e. $U$ output neurons per network branch). Every hidden neuron uses the Rectifier Linear Unit (ReLU) function, $f(x) = \max(0, x)$, as activator function.

The DNN weights are updated using stochastic gradient descent. Specifically, we trained the agent with mini-batches of $D = 32$ samples using the Adam optimizer [31] with learning rate $\alpha = 10^{-4}$ and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. The agent explores the environment with probability $\epsilon = 1$ that is gradually decremented after each episode following the update rule $\epsilon \leftarrow \max\{0.1, 0.99\epsilon\}$. We summarize the agent parameters in Table 2.

We analyze the agent performance in terms of SLA satisfaction and aggregated user throughput and we compare it to two resource allocation algorithms, namely Round Robin (RR) and Weighted Max Rate (WMR). The former is a simple scheme that ensures the highest user fairness as the resources are evenly shared by the users. The latter is a more advanced scheme that ensures throughput optimality in multi-queues systems [32].

**Round Robin (RR)**: it equally allocates the BWPs so that each user has access to a portion of the radio resources for the same amount of time regardless of the wireless channel status. To ensure a more meaningful comparison with the agent, we heuristically reduce the INI impact by contiguously allocating BWPs of the same numerology.

**Weighted Max Rate (WMR)**: it assigns every BWP by weighting the achievable data rate of each user with respect to its buffer size. Note that the achievable data rate is estimated from the subchannel gain $g^u_{t,k}$ without accounting for the INI since the latter is generated once the BWP has been already computed. Formally, for each BWP $k$, the scheduled user is selected according to

$$\underset{u}{\text{argmax}}\, r^u_{t,k} \cdot \frac{\tilde{b}_u}{\sum_{u'} \tilde{b}_{u'}}, \tag{25}$$

where $\tilde{b}_u = \hat{r}^u_t$ if $u \in U_e$ or $\tilde{b}_u = b^u_t$ if $u \in U_l$.

Although the aforementioned resource allocation metrics are used in many practical systems, we acknowledge that a comparison with the schemes discussed in the related work section could provide more insight about the performance of our approach. However, a direct comparison to these works is not feasible due to the lack of support of the mixed-numerology resource grid structure. Nonetheless, it is worth noting that the authors of [14] propose an URLLC and eMBB multiplexing scheme leveraging the NR frame flexibility, which can be configured in mini-slots of heterogeneous duration. This structure shares some similarities with the mixed-numerology structure considered in this paper. Therefore, we simulated our agent with URLLC traffic load conditions similar to [14] and calculated the resulting system spectral efficiency under a variable URLLC traffic. In both schemes, the increase of URLLC data rate lowers the cell spectral efficiency, as the granularity of the allocated time-frequency blocks cannot be perfectly tailored to the frequent arrival rate of small-size URLLC packets. In other words, the URLLC service is not able to exploit the available spectrum as efficiently as the eMBB slice which is characterized by a higher and more consistent data rate. The scheme in [14] shows a smooth spectral efficiency degradation since it minimizes the time-slot duration of the URLLC service in order to mitigate the negative impact over the eMBB slice. Differently, our scheme shows a stepwise spectral efficiency degradation as the URLLC traffic load is gradually increased. This behavior is due to the fact that mixed-numerology schemes are constrained by the temporal alignment between different numerology BWPs in order to maintain the symbol synchronization thanks to a fixed TTI duration. Consequently, the allocated spectrum is used less efficiently by the URLLC service compared to the mini-slot approach of [14] in the presence of a sporadic URLLC packet arrival rate. However, when the URLLC traffic assumes higher volumes, our agent is able to better exploit the multi-user diversity to boost the system capacity while mitigating the INI and thus limiting the spectral efficiency degradation.

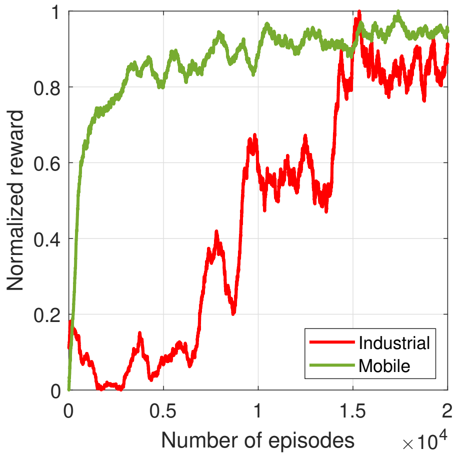

Figure 4: Normalized agent reward obtained using the rewards (15) and (13) with $c_e = 10$ and $c_l = 10$ when eMBB and URLLC numerology is 15 kHz and 60 kHz, respectively.

### 6.2. Training performance

In Fig. 4, we compare the agent learning performance between the two considered network scenarios by showing the achieved normalized reward. We can observe that the agent converges within a smaller number of episodes when employed for the mobile network scenario. This behavior is explained by taking into account that the reward designed for the industrial scenario does not prioritize the service provisioning satisfaction at each time step unlike the reward designed for the mobile network scenario. Consequently, the agent requires more episodes to find the optimal policy as there are many BWP allocations, ensuring the latency requirement fulfilment, that schedule the industrial URLLC users at different time steps.

### 6.3. Agent performance in mobile network scenario

We tested the trained agent by simulating the BWP scheduling for a duration of 1 s, corresponding to 250 4-ms episodes. Note that we discarded URLLC packets exceeding the delay constraint. For every simulation, a new user distribution is randomly generated and is kept fixed for its whole duration. We initially discuss the agent performance in the mobile network scenario, hence we employ (13) as reward function during the training phase.

In Fig. 5, we plot the CDF of the per-user data rate for the eMBB slice when the URLLC slice employs a numerology of 15 kHz and 60 kHz, respectively. We can observe that the INI significantly degrades the SLA fulfilment performance of both the RR and WMR schedulers. In particular, the RR scheme is the most affected as it is channel-unaware. Conversely, the agent shows a more limited performance degradation and guarantees the lowest violation

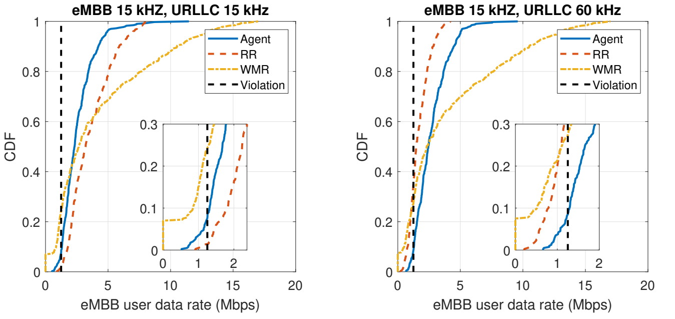

Figure 5: Data rate CDF of the eMBB users. The minimum data requirement is $r_e = 1.25$ Mbps.

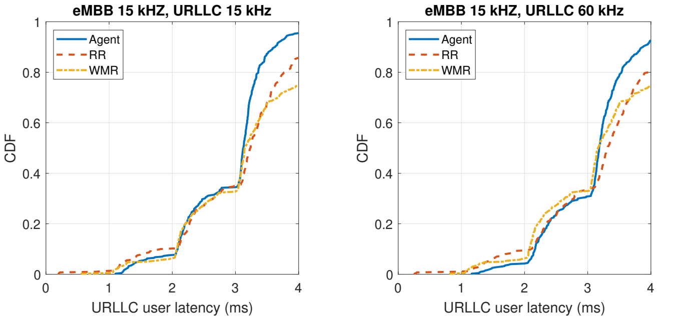

Figure 6: Latency CDF of the URLLC users. The maximum latency is $T = 4$ ms.

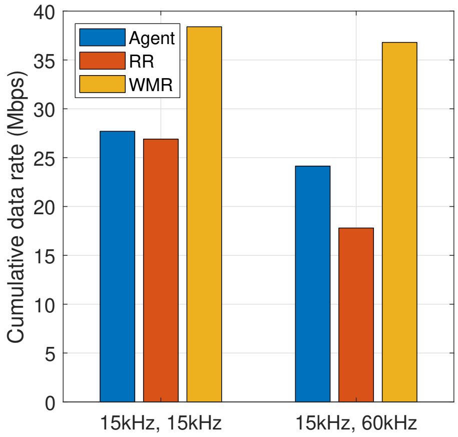

Figure 7: Average cumulative data rate of eMBB and URLLC slices.

probability (i.e. the probability that an eMBB user has a data rate lower than $r_e$) for the mixed-numerology case. This behavior exemplifies the agent capability to correlate the INI dynamic with the different subchannel gains of each user in order to boost data rate. This motivation also indirectly explains the lower agent performance in the single-numerology case when compared to the RR scheduler. In detail, the INI level generated by the BWP allocation is exploited by the agent to identify those subchannel gains configurations providing a favorable interference mitigation to the related users in a given time step. In other words, the deterministic effect of the BWP allocation on the INI generation helps the agent to reduce the purely stochastic nature of the channel, which changes independently of the computed action, and thus to improve its generalization capability. For this reason, when the INI effect is not present as in the single-numerology case, the agent struggles to efficiently discriminate the best BWP allocation that can simultaneously satisfy the minimum data rate requirement and, at the same time, that can maximize the aggregate throughput.

In Fig. 6, as similarly done for the eMBB slice, we plot the per-user latency performance of the URLLC service achieved by the single-numerology and the mixed-numerology cases. Differently from eMBB slice, the INI impact is much more limited as it is shown by the similar latency distributions for all schemes, where the RR scheduler is the most affected as it cannot counteract the INI effect with a suitable BWP allocation that maximizes the throughput as the WMR metric. In general, this result is explained by the asymmetric INI behavior that affects more heavily the numerologies with a small subcarrier spacing compared to the ones with a wider spacing. Nonetheless, the agent guarantees an average user latency that is lower than the other schemes since the INI mitigation together with the prioritized transmission of URLLC packets boosts the network capability in delivering the data traffic within the latency deadline.

In Fig. 7 we show the total network data rate computed as the aggregate performance of the two slices. We note that the Agent approach ensures a modest throughput loss when the mixed-numerology case is considered. Moreover, unlike the performance loss suffered by the other schemes, which heavily degraded the service reliability in terms of the minimum data rate and average latency, the agent ensures a performance comparable to the single-numerology case as previously discussed. In other words, in order to mitigate the unavoidable performance loss due to the INI, the proposed agent achieves a lower throughput in favor of a BWP allocation that limits the degradation of the service quality.

### *6.4. Agent performance in industrial network scenario*

We now discuss the performance obtained by the agent trained with the reward function (15). To better highlight the agent performance in the industrial network scenario that allows the agent to exploit the additional knowledge about the URLLC traffic statistics, we compare the service reliability of both slices to the mobile network scenario case. In details, we compare the two scenarios by showing the slice reliability as a function of $\alpha = c_l/c_e$, which expresses the magnitude of the penalty obtained when the requirements of the URLLC slice are not met, versus the penalty obtained when the requirements of the eMBB slice are not met in the mobile network scenario. Specifically, high values of $\alpha$ indicate that the agent prioritizes the URLLC service fulfilment.

In Fig. 8, we report the results for the single-numerology case. With regard to the eMBB slice, we plot the probability of providing a data rate higher than $r_e$, whereas for

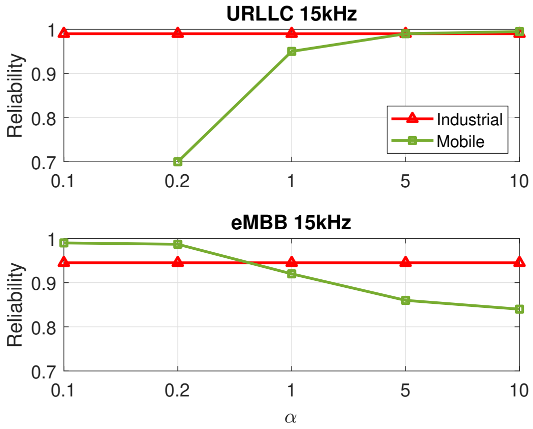

Figure 8: eMBB and URLLC slice reliability obtained with reward (15) and (13) in the single-numerology case.

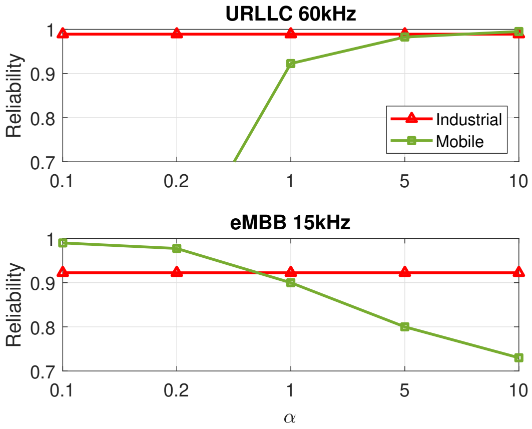

Figure 9: eMBB and URLLC slice reliability obtained with reward (15) and (13) in the mixed-numerology case.

URLLC slice, we plot the probability of transmitting the packets within the time deadline $T$. As expected, by increasing the value of $\alpha$, the URLLC reliability increases at the expense of the eMBB reliability. However, there is no value of $\alpha$ that simultaneously provides for both slices a reliability score that is higher than the scores obtained by the agent employed in the industrial scenario. As a matter of fact, even when considering the configuration with $\alpha = 1$, which ensures a balanced trade-off between the two services, the deterministic traffic patterns of the URLLC slice in the industrial scenario allow the agent to increase the related slice reliability without hindering the eMBB slice performance.

In Fig. 9 we show the results for the mixed-numerology case. We note that the industrial network scenario does not suffer from a performance loss due to INI, differently from the mobile scenario. Similarly, the eMBB service reliability degradation is limited. Such performance gain derives from the fact that the URLLC packet arrival is scheduled for a fixed time slot with known periodicity, hence the agent can explore multiple BWP allocations that mitigate the INI while ensuring that the packet is delivered within the latency deadline. This behavior is ensured by the reward (15), which is designed to provide a non-zero reward as long as the SLA requirements are met by the end of episode.

## 7. Conclusion

We proposed a DRL agent to compute a slice spectrum allocation policy for eMBB and URLLC network slices. We considered a mixed-numerology access scheme at the physical layer. The proposed solution analytically considers the impact of the inter-numerology interference on the user performance and it maximizes the system throughput subject to the eMBB and URLLC service requirements. Moreover, we differentiated the agent reward functions in order to effectively accommodate the stochastic nature of URLLC traffic, which is typical in mobile networks, as well as the deterministic nature of URLLC traffic, which is typical in industrial networks. We compared the agent performance to two INI-unaware benchmark schemes based on different metrics. Results showed that the proposed DRL solution provides a higher service reliability thanks to the INI mitigation. Moreover, the deterministic feature of the URLLC traffic allows the agent to further enhance the service provisioning performance when compared to stochastic traffic.

## References

[1] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega *et al.*, “Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks,” *IEEE Communications magazine*, vol. 55, no. 5, pp. 72–79, 2017.

[2] C. Guimarães *et al.*, “Public and non-public network integration for 5Growth industry 4.0 use cases,” *IEEE Communications Magazine*, vol. 59, no. 7, pp. 108–114, 2021.

[3] A. Aijaz, “Private 5G: The future of industrial wireless,” *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 136–145, 2020.

[4] “Study on Communication for Automation in Vertical Domains,” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 22.804, 2020, version 16.3.0.

[5] W. Nakimuli, J. Garcia-Reinoso, J. E. Sierra-Garcia, P. Serrano, and I. Q. Fernández, “Deployment and evaluation of an industry 4.0 use case over 5G,” *IEEE Communications Magazine*, vol. 59, no. 7, pp. 14–20, 2021.

[6] 5GACIA, “A 5G Traffic Model for Industrial Use Cases,” *white paper*, 2019.

[7] P. Guan, D. Wu, T. Tian, J. Zhou, X. Zhang, L. Gu, A. Benjebbour, M. Iwabuchi, and Y. Kishiyama, “5G Field Trials: OFDM-based Waveforms and Mixed Numerologies,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1234–1243, 2017.

[8] M. Zambianco and G. Verticale, “Mixed-numerology interference-aware spectrum allocation for embb and urllc network slices,” in *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*. IEEE, 2021, pp. 1–8.

[9] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, “A Survey on 5G Usage Scenarios and Traffic Models,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 905–929, 2020.

[10] A. B. Kihero, M. S. J. Solaija, and H. Arslan, "Inter-numerology Interference for Beyond 5G," *IEEE Access*, vol. 7, 2019.

[11] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN Resource Slicing Mechanism for Multiplexing of eMBB and URLLC Services in OFDMA based 5G Wireless Networks," *IEEE Access*, vol. 8, pp. 45 674–45 688, 2020.

[12] H. Yin, L. Zhang, and S. Roy, "Multiplexing URLLC Traffic within eMBB Services in 5G NR: Fair Scheduling," *IEEE Transactions on Communications*, 2020.

[13] C. Tang, X. Chen, Y. Chen, and Z. Li, "Dynamic Resource Optimization Based on Flexible Numerology and Markov Decision Process for Heterogeneous Services," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 610–617.

[14] 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.

[15] L. Marijanović, S. Schwarz, and M. Rupp, "Optimal Resource Allocation with Flexible Numerology," in *2018 IEEE International Conference on Communication Systems (ICCS)*. IEEE, 2018, pp. 136–141.

[16] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, "Latency-sensitive 5g ran slicing for industry 4.0," *IEEE Access*, vol. 7, pp. 143 139–143 159, 2019.

[17] D. Ginthör, R. Guillaume, M. Schüngel, and H. D. Schotten, "5g ran slicing for deterministic traffic," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[18] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, "Latency-based 5g ran slicing descriptor to support deterministic industry 4.0 applications," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 1359–1362.

[19] D. Ginthör, R. Guillaume, M. Schüngel, and H. D. Schotten, "Robust end-to-end schedules for wireless time-sensitive networks under correlated large-scale fading," in *2021 17th IEEE International Conference on Factory Communication Systems (WFCS)*. IEEE, 2021, pp. 115–122.

[20] "System architecture for 5G system," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 2021, version 17.2.0.

[21] "5G; NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 2019, version 15.5.0.

[22] "5G; NR; Physical procedures for control," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.213, 2018, version 15.2.0.

[23] X. Zhang, L. Zhang, P. Xiao, D. Ma, J. Wei, and Y. Xin, "Mixed Numerologies Interference Analysis and Inter-Numerology Interference Cancellation for Windowed OFDM Systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7047–7061, 2018.

[24] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*. Springer Science & Business Media, 2013.

[25] R. S. Sutton, A. G. Barto *et al.*, *Introduction to Reinforcement Learning*. MIT press Cambridge, 1998, vol. 135.

[26] A. Tavakoli, F. Pardo, and P. Kormushev, "Action Branching Architectures for Deep Reinforcement Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[27] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *International conference on machine learning*, 2016, pp. 1995–2003.

[28] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," *arXiv preprint arXiv:1511.05952*, 2015.

[29] "Minimum Requirements related to Technical Performance for IMT-2020 Radio Interface(s)," ITU-R, Tech. Rep. M.2410-0, 2017.

[30] "Guidelines for Evaluation of Radio Interface technologies for IMT-2020," ITU-R, Tech. Rep. M.2412-0, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] Y. Chen, X. Wang, and L. Cai, "On Achieving Fair and Throughput-optimal Scheduling for TCP flows in Wireless Networks," *IEEE Transactions on wireless communications*, vol. 15, no. 12, pp. 7996–8008, 2016.