LACO: A Latency-Driven Network Slicing Orchestration in Beyond-5G Networks

Lanfranco Zanzi, *Associate Member, IEEE*, Vincenzo Sciancalepore, *Senior Member, IEEE*, Andres Garcia-Saavedra, Hans Dieter Schotten, *Member, IEEE*, and Xavier Costa-Pérez, *Senior Member, IEEE*

***Abstract*— *Network Slicing* is expected to become a game changer in the upcoming 5G networks and beyond, enlarging the telecom business ecosystem through still-unexplored vertical industry profits. This implies that heterogeneous service level agreements (SLAs) must be guaranteed *per slice* given the multitude of predefined requirements. In this paper, we pioneer a novel radio slicing orchestration solution that simultaneously provides *latency* and throughput guarantees in a multi-tenancy environment. Leveraging on a solid mathematical framework, we exploit the exploration-vs-exploitation paradigm by means of a multi-armed-bandit-based (MAB) orchestrator, LACO, that makes adaptive resource slicing decisions with no prior knowledge on the traffic demand or channel quality statistics. As opposed to traditional MAB methods that are blind to the underlying system, LACO relies on system structure information to expedite decisions. After a preliminary simulations campaign empirically proving the validness of our solution, we provide a robust implementation of LACO using off-the-shelf equipment to fully emulate realistic network conditions: near-optimal results within affordable computational time are measured when LACO is in place.**

***Index Terms*— 5G-and-beyond, virtualization, network slicing, MAB, latency control, resource management.**

I. Introduction

THE quest for new sources of revenue that revitalizes the mobile industry has spawned an unprecedented hype around the fifth-generation of mobile networks (5G) and, in particular, the network slicing concept. Enabled by software-defined networking (SDN) and network function virtualization (NFV), network slicing allows telco operators to offer virtualized slices of infrastructure resources *on-demand* to heterogeneous 3rd-party services [1]. A high-level view of the system considered in this paper is described in Fig. 1. The figure represents a series of *sliceable* base stations as a pool of radio resources (coloured cubes in the figure). The resource allocation process is considered hierarchical: while *bundles* of radio resources are assigned to different tenants (namely radio slices), each tenant autonomously schedules its bundle of radio resources to each individual user following classic radio scheduling policies. The difference between such operations is subtle but of paramount importance: a slice controller operates at a larger timescale and thus over a coarser granularity [2], [3]. While most prior work on network slicing focuses on average bit-rate guarantees [3], [4], *latency* considerations have received little attention. Latency aspects however are gaining more and more attraction as a quest to face advanced use-cases requirements, e.g., autonomous driving and platooning [5] in Vehicle-to-everything (V2X) enabled scenarios. In this context, accurate resource allocation schemes and inter-slice isolation aspects are fundamental features to support the provisioning of latency-constrained services.

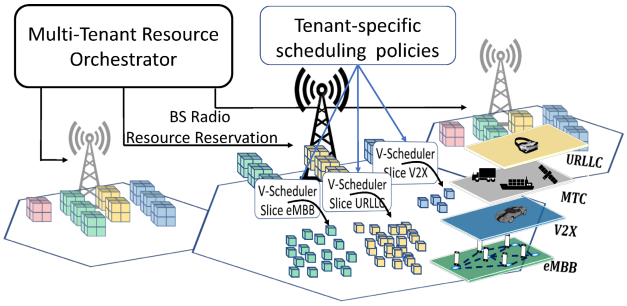


Fig. 1. Illustration of the network slicing concept.

Given the plethora of works on classic radio scheduling [6], [7], we keep this aspect out of the scope of this paper and we focus instead on the former impelling need: a proper design of an orchestration solution that autonomously assigns chunks of radio spectrum (slices) in relatively larger time-scales pursuing the goal of *guaranteeing simultaneously latency and throughput constraints*. From the best of our knowledge, there is a non-negligible lack of works focusing on both aspects simultaneously in sliced-network environments.

Manuscript received March 22, 2020; revised July 11, 2020 and September 7, 2020; accepted September 22, 2020. Date of publication October 7, 2020; date of current version January 8, 2021. This work was supported in part by the H2020 MonB5G Project under Grant 871780 and in part by the H2020 5Growth Project under Grant 856709. The associate editor coordinating the review of this article and approving it for publication was L. Duan. *(Corresponding author: Lanfranco Zanzi.)*

Lanfranco Zanzi is with the NEC Laboratories Europe GmbH, 69115 Heidelberg, Germany, and also with the Department of Wireless Communication and Navigation (WICON), Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany (e-mail: lanfranco.zanzi@neclab.eu).

Vincenzo Sciancalepore and Andres Garcia-Saavedra are with the NEC Laboratories Europe GmbH, 69115 Heidelberg, Germany (e-mail: vincenzo.sciancalepore@neclab.eu; andres.garcia.saavedra@neclab.eu).

Hans Dieter Schotten is with the Department of Wireless Communication and Navigation (WICON), Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany (e-mail: schotten@eit.uni-kl.de).

Xavier Costa-Pérez is with the NEC Laboratories Europe GmbH, 69115 Heidelberg, Germany, and also with the i2CAT Foundation and ICREA, 08034 Barcelona, Spain (e-mail: xavier.costa@neclab.eu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.3027963

To fill this gap, we design a LAtency-Controlled Orchestrator (LACO), a network slice controller that maps virtual radio resource allocations to physical resources while still guaranteeing latency requirements.[1] Specifically, LACO augments such prior work by accommodating resources to (granted) slices such that latency agreements are satisfied. This unlocks a new business opportunity for the telco operators that may apply customized pricing models according to the *elasticity* of offered slice latency constraints.

Technical Challenges: While designing LACO, two sources of uncertainty need to be under control: i) the behavioral dynamics of the (aggregated) demand across involved tenants and ii) the inherent randomness of the wireless channel. These system dynamics have been traditionally modeled via either complex solutions that are hard to solve in realistic settings or via simplistic assumptions at the expense of low performance figures. In our work, we explore a novel approach by designing a scheme that *learns* the implications that allocation decisions have on per-slice latency without explicitly making assumptions on the underlying dynamics. To this aim, we first model our decision-making problem as a Markov Decision Process[2] (MDP), which allows us to neglect low-level details of the tenant demands and channel dynamics while letting us retain some knowledge on the consequences that a given action may have on the most immediate next system state.

An MDP model helps us to fully explore the problem features. However, the process of learning the state transition probability matrix of each of the embedded Markov chains incurs in prohibitive overhead as a reinforcement learning agent has to explore the whole space of state-action trajectories—the so-called *curse of dimensionality*. To address this, we resort to a Multi-Armed Bandit (MAB) model where the attained reward depends only on the action taken from a bounded set of possible actions. Importantly, in contrast to traditional MAB methods, LACO is *model-aware* (though not model-dependant), i.e., it exploits (abstracted) information regarding the underlying system to expedite the selection of highly rewarding actions, which is particularly attractive when dealing with dynamic non-stationary scenarios.

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

- We introduce a Discrete-Time Markov Chain (DTMC) model to capture the dynamics of the (instantaneous) aggregate slice traffic demand and the wireless channel variations.
- We present a latent variable regression model to accurately anticipate the transition probability matrix of the proposed DTMCs.
- We formulate the dynamic slice resource provisioning as a Markov Decision Process (MDP).
- We design a *model-aware* Multi-Armed Bandit (MAB) method to guide the decision-making process, which relies on the above DTMC models and anticipated transition probabilities to speed up convergence.
- We present an exhaustive simulations campaign to assess the performance of our approach.
- We implement and field-test our solution using off-the-shelf equipment that emulates real network conditions: LACO shows its innovative performance gain against considered legacy techniques.

The remainder of the paper is structured as follows. Section II formulates our problem and presents the main building blocks of LACO. Section III introduces an DTMC model that helps us expedite the action-space exploration phase and Section IV deeply analyzes it. In Section V, we introduce our decision process as a Markov Decision Process (MDP) and present a model-aware Multi-Armed Bandit decision-making engine integrated in LACO. Section VI presents our preliminary simulation campaign to validate the design principles of LACO, whereas Section VII details the implementation of our novel solution into off-the-shelf equipment with realistic network performance. Finally, Section VIII summarizes related literature and Section IX concludes the paper with some final remarks.

II. LACO: The Framework Overview

Our solution relies on the concept of slicing-enabled networks wherein multiple network tenants are willing to obtain a network slice with predefined service level agreements (SLAs). Such SLAs may be expressed in terms of maximum slice throughput and average access latency. Within the context of our paper, we define the average access latency as time the traffic belonging to a certain slice needs to wait before being served due to scheduling procedures. In particular, we focus on the radio access network (RAN) domain and design LACO, a RAN controller that dynamically provisions spectrum resources to admitted network slices while providing latency guarantees. In the following, we overview the main system building blocks with detailed notation and assumptions.

A. Business Scenario

We consider different entities in our system: i) an *infrastructure provider* owning the physical infrastructure who offers isolated RAN slices as a service, ii) *tenants* who acquire and manage slices with given SLAs to deliver services to end-users, and iii) *end-users*, who demand radio resources from such tenants/slices.

Let us define $\mathcal{I}$ as the set of running network slices and $\mathcal{U}_i$ as the set of end-users associated to the i-th slice. The total amount of wireless resources (radio spectrum) is split into multiple non-overlapping network slices, each one belonging to one single tenant $i \in \mathcal{I}$.[3] Based on fixed SLAs, each network slice is characterized by *maximum throughput* and *expected latency* denoted by Λ_i and Δ_i, respectively.

[1]Note that LACO does *not* compete with state-of-the-art throughput-only slice controllers—in fact, we purposely assume the presence of an admission controller that ensures that the aggregate load incurred by granted slices is within the system capacity region.

[2]With a little misuse of nomenclature, we will refer to Markov Decision Process (MDP) rather than Semi-Markov Decision Process (SMDP) despite considering continuous time scales.

[3]We assume a one-to-one mapping between slices and tenants. Therefore, we use $i \in \mathcal{I}$ interchangeably throughout the paper as a tenant identifier or its associated slice. Note that this assumption can be easily relaxed in the model.

TABLE I
NOTATION TABLE

Notation	Description	Notation	Description	Notation	Description
y_i	Slice configuration	z_σ	Arm selection freq.	$\phi \in \Phi$	Action index
$n \in N$	Decision epoch index	$\mathcal{N}_i(\mu_i, \nu_i^2)$	Normal distribution	$\omega(\cdot)$	Latent var. weight
$u_i \in \mathcal{U}_i$	User index	$m \in \mathcal{M}$	MCS index	$R(\cdot)$	Reward function
$d \in \{0, 1\}$	Exceed delay flag	$i \in \mathcal{I}$	Slice index	$\psi(\sigma)$	Accuracy value
$\lambda_i^{(n)}$	Inst. traffic demand	$r_i^{(n)}$	Bits not served	L_i	Latency constraint
$f(x, \rho_i)$	Traffic demands distr.	$\bar{u}_i$	Aggregate user	Γ_m	Bits per subframe
$\zeta(\cdot)^{(n)}$	Throughput mapping	$g \in \mathcal{G}$	Channel level	$f(x, \theta_i)$	Channel distr.
$\sigma \in \Sigma$	MDP state index	$\gamma_i^{(n)}$	Inst. SNR	Δ_i	Latency tolerance
$w \in \mathcal{W}$	(Latent) Channel quality	C	Capacity of BS	τ	Rayleigh scale param.
$T(\cdot)$	Transition function	ϵ	Decision interval duration	Θ	PRB chunk

We assume that an admission control process[4] is concurrently running on a higher tier so that the average aggregate load can be accommodated within the overall system capacity.

B. Notation

We use conventional notation. We let $\mathbb{R}$ and $\mathbb{Z}$ denote the set of real and integer numbers. We use $\mathbb{R}_+$, $\mathbb{R}^n$, and $\mathbb{R}^{n \times m}$ to represent the sets of non-negative real numbers, n-dimensional real vectors, and $m \times n$ real matrices, respectively. Vectors are denoted as column vectors and written in bold font. Subscripts represent an element in a vector and superscripts elements in a sequence. For instance, $\langle \boldsymbol{x}^{(t)} \rangle$ is a sequence of vectors with $\boldsymbol{x}^{(t)} = [x_1^{(t)}, \ldots, x_n^{(t)}]^T$ being a vector from $\mathbb{R}^n$, and $x_i^{(t)}$ the i'th component of the t'th vector in the sequence. Operation $[\cdot]^T$ represents the transpose operator while $[x_1, \ldots, x_n]_{\text{diag}}$ translates the vector into a diagonal matrix. Last, $\mathbb{1}$ and $\mathbb{0}$ indicate an all-ones and all-zeroes vector, respectively, and $\lceil \cdot \rceil$ is the ceiling operation.

C. Problem Definition

Assuming that an instance of LACO is executed per base station (BS) as shown in Fig. 1, we focus our problem design and performance evaluation on a single BS characterized by a capacity C, which is the sum of a discrete set of available physical resource blocks (PRBs) of fixed bandwidth. This resource availability must be divided into subsets of PRBs (i.e., slices), and our job is to dynamically assign such subsets to each network slice $i \in \mathcal{I}$. We refer to such assignment as the *configuration* of slice i, denoted by the variable y_i. Obviously, we shall guarantee $\sum_{i \in \mathcal{I}} y_i \leq C$. For the sake of clarity, we summarize all mathematical variables used throughout the paper in Table I.

We consider a time-slotted system where time is divided into *decision epochs* $n = \{1, 2, \ldots, N\}$. The decision epoch duration ϵ may be decided according to the infrastructure provider policies, ranging from few seconds up to several minutes. While the admission controller (pre-)selects a subset of slices that can co-exist without exceeding the capacity of the system *in average*, the dynamic nature of the slice's load and wireless channel may cause instantaneous load surges or channel quality fading effects and hence induce a non-zero mean delay.

We denote the experienced instantaneous signal-to-noise ratio (SNR) of slice i (averaged out across all users of the slice) and the instantaneous aggregate traffic demand within time-slot n as $\gamma_i^{(n)}$ and $\lambda_i^{(n)}$, respectively. As each tenant i may show different behavior in terms of wireless channel evolution (according to θ_i) and traffic demands (according to ρ_i), we also assume $\gamma_i^{(n)}$ and $\lambda_i^{(n)}$ are drawn from different univariate probability density function, i.e., $\gamma_i^{(n)} \sim f(x, \theta_i)$ and $\lambda_i^{(n)} \sim f(x, \rho_i)$. Importantly, we do not assume any knowledge on such random variables; we exploit machine learning techniques to *learn* the inherent channel and demand models, which allow our system to dynamically adapt the slice configurations $y_i^{(n)}$ at every decision epoch n while mitigating latency constraint violations.

Formally, the above-described problem becomes:

Problem `LATENCY-CONTROL`*:*

$$\begin{aligned}
\text{minimize} \quad & \lim_{N \to \infty} \sum_{n=1}^{N} \mathbb{E}\left[\sum_{i \in \mathcal{I}} r_i^{(n)} \right] \\
\text{subject to} \quad & \mathbb{E}\left[\frac{\lambda_i^{(n)}}{\zeta\left(y_i^{(n)}, \gamma_i^{(n)}\right) + r_i^{(n)}} \right] \leq \Delta_i, \quad \forall i \in \mathcal{I}; \\
& \sum_i y_i^{(n)} \geq C, \quad \forall n; \\
& y_i^{(n)}, r_i^{(n)} \in \mathbb{Z}_+, \quad \forall i \in \mathcal{I};
\end{aligned}$$

where $\zeta(\cdot)^{(n)}$ is a mapping function that returns the number of bits that can be served using the allocated number of PRBs ($y_i^{(n)}$) and the current SNR level $\gamma_i^{(n)}$, as per [8, §7.1.7]. The traffic demand might not be satisfied within a single decision epoch incurring in packet queuing and additional delay. Therefore, in our formulation we introduce $r_i^{(n)}$ as a deficit value indicating the number of bits not served within the agreed slice latency tolerance Δ_i during the time-slot n (i.e., dropped). The objective of Problem `LATENCY-CONTROL` is hence to find a sequence of $\langle y_i^{(n)} \rangle$ configurations such that the expected total non-served traffic demand is minimized. Hereafter whenever is evident from context, we drop the superscript (n) to reduce clutter. To address the problem, we rely on a two-layer scheduling approach commonly adopted in the network slicing context [3], [9]. On the one side, an

[4]Given the plethora of solutions in the literature, the admission control design is out of the scope of this work. We refer the reader, for example, to [2], [4] for more details.

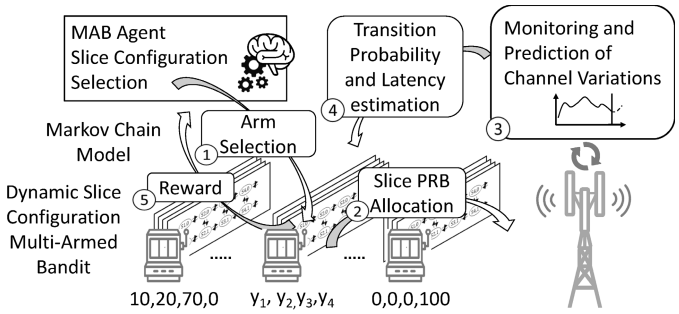


Fig. 2. Workflow illustration.

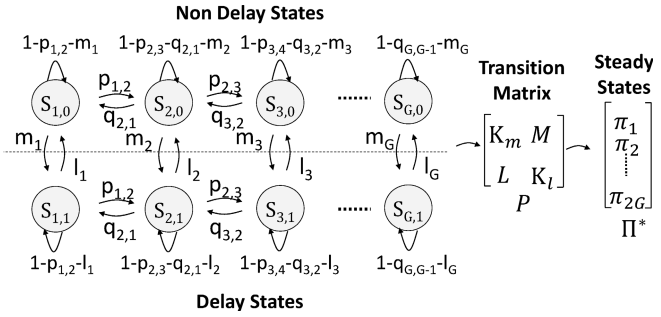


Fig. 3. Radio channel variations as Markov chain.

inter-slice scheduler is in charge of defining the PRB allocation strategy to meet the networking requirements while ensuring resource isolation among slices. On the other side, a lower layer intra-slice scheduler enforces the assignment of the pre-allocated subset of PRBs to the connected end-users. Our work mainly focuses on the higher-level inter-slice scheduler, leaving the implementation of intra-slice scheduling strategies open to address tenant-specific requirements.

D. Working Flow

For a given slot n, problem LATENCY-CONTROL can be easily linearized[5] and solved with standard optimization tools. However, this approach may exhibit sub-optimal behavior in future epochs if the statistical distributions of $f(x, \theta_i)$ and $f(x, \rho_i)$ are not stationary. Hence, we propose a novel two-fold approach that: i) models channel and traffic demand variations based on previous observations, and ii) iteratively applies slice settings towards the goal of honouring SLAs.

Fig. 2 depicts the building blocks of our solution. LACO relies on the concept of Markov Decision Process (MDP) as described in Section V to decide which configuration y_i should be enforced to all active slices i, adapting its choice at every epoch n according to the observed *reward* function that measures the incurred latency. In turn, this information is asymptotically calculated within Discrete-Time Markov Chain (DTMC) model described in Section III. The transition probabilities of such DTMC are updated according to previous observations in the *Monitoring and Prediction of Channel Variations* module, described in Section IV.

III. DTMC Model

Hereafter, we analyze the system dynamics through a Markov Chain-based (MC) model that computes expected channel conditions and violations on latency tolerance. It should be noted that channel variations and traffic demands are independently obtained according to each slice, thus each DTMC may be treated individually without the need to setup a Markov chain accounting for the overall system configuration. Such global DTMC could anyway be easily obtained as linear combination of the individual DTMCs. For the sake of tractability, we consider a single (virtual) user $\bar{u}_i$ with an aggregate traffic demand resulting from the set of users $u_i \in \mathcal{U}_i$ belonging to slice i.[6] We also assume a finite number of channel quality levels G, which may bound each instantaneous user channel quality γ_i, as depicted in Fig. 3. This is a system design choice and allows operators to trade off high accuracy for convergence speed, by ranging from a fine-grained scale (large G), e.g. by letting each channel quality level be equal to the modulation and coding scheme (MCSs) as defined in the 3GPP standard document [8], to a coarse-grained scale that may capture the channel variation behaviors with limited accuracy, as detailed in Section IV.

Let us consider a discrete-time stochastic process X_t[7] that takes values from a finite and discrete state space, which is denoted by $\mathcal{S} = \{S_{0,0}, \ldots, S_{g,d}, \ldots, S_{G,1} \mid 0 \leq g \leq G, d \in \{0,1\}\}$.[8] In particular, a realization of X_t when visiting state $S_{g,d}$ represents virtual user $\bar{u}_i$ experiencing channel level $g \in \mathcal{G}$ with an associated delay exceeding the one specified by the slice SLA ($d = 1$) or otherwise ($d = 0$). When considering wireless channel conditions as Rayleigh distributed, it is common practice to model the variations as a sequential visiting of consecutive states, as the channel does not vary faster than the Markov chain time-slot [11]. Hence, we define the probability to improve the user channel condition from level g to level $g+1$ as $p_{g,g+1}$ whereas the probability to get a bad channel from level g to level $g-1$ as $q_{g,g-1}$. As shown in recent works like [12], [13], accurate scheduling strategies might mitigate the interference effects coming from multiple base stations serving the same sets of slices thus improving the overall channel conditions. However, such schemes introduce additional complexity and synchronization overhead, which hardly fit with our view of a lightweight base station oriented solution. Last, given the available physical resource blocks assigned to a particular slice y_i, the channel quality level g and the overall traffic demand within the time-slot, we model the probability to incur in delay constraint violation as m_g and the probability to keep the access delay within the agreed bound as l_g. This process can be formulated as a two-dimensional DTMC $M := (\mathcal{S}, P)$,

[5]Function $\zeta(\cdot)$ can be easily approximated with a linear function by applying piece-wise linearization.

[6]This assumption can be readily relaxed by considering the convolution of single cumulative distribution functions of every user channel and demand variation [10].

[7]The time scale t of DTMC state switch is much shorter than the decision epoch n used in the MDP described in Section V.

[8]Each DTMC is defined within a state space $\mathcal{S}^i$. We remove the index i to limit the clutter, as the analysis can be easily extended to any other slice.

where P denotes the following transition probability:

$$P = \begin{vmatrix} K_m & M \\ L & K_l \end{vmatrix}, \text{where } K_{x=\{m,l\}} = \{k_{ij}^{(x)}\} \quad (1)$$

$$\text{with } k_{ij}^{(x)} = \begin{cases} 1 - p_{i,i+1} - q_{i,i-1} - x_i & \text{if } i = j, \\ q_{i,j} & \text{if } i = j+1, \\ p_{i,j} & \text{if } i = j-1, \\ 0 & \text{otherwise;} \end{cases}$$

$$\text{and } M = \{m_i\}_{\text{diag}}, L = \{l_i\}_{\text{diag}}.$$

Note that we assume $p_{G,G+1} = q_{1,0} = 0$ and each square block $K_{x=\{m,l\}}, M$ and L with $[G \times G]$ size so that the square matrix P has dimension $[2G \times 2G]$. Without loss of generality, we assume that such transition probabilities do not depend on the particular time-slot we are evaluating. Thus, we define our DTMC as a time-homogeneous MC where the process X_t evolves based on $\Pi(t) = \Pi(0)\ P^t$ where the row vectors $\Pi(t)$ and $\Pi(0)$ represent the first order state probability distribution at time n and 0, respectively. In order to evaluate the long-term behavior of our system, we need to calculate the steady-state probability $\Pi^* = \{\pi_s^*\}$ of being in each of the defined states. It yields that

$$\Pi^* = \lim_{n\to\infty} \Pi(t) = \Pi(0) \lim_{n\to\infty} P^t = \Pi(0)P^*. \quad (2)$$

The above-described Markov chain is irreducible, as each state may reach through available paths any other state. Therefore, by stochastic theory, if a Markov chain is irreducible and non-periodic, the steady-state probability distribution Π^* always exists, is unique and is independent from the initial conditions.

Recalling the total probability theorem and using Eq. (1), we calculate the steady-state probability distribution as the solution of the following equations

$$\begin{cases} (P^T - \mathbb{1}_{\text{diag}})\,\Pi^* & = 0 \\ \mathbb{1}\,\Pi^* & = 1 \end{cases} \quad (3)$$

where $\mathbb{1}_{\text{diag}}$ is the identity matrix.

IV. DTMC Monitoring and Prediction

The asymptotic behavior of a Markov chain depends on the transition probability matrix P, which in turn depends on the stochastic processes of the slice traffic demands and wireless channel variations. While several models have been already defined in the literature to derive such probabilities [14], the latency control objective and the need of an accurate estimation exacerbate the problem and render model-fitting approaches impractical. This brings additional complexity and delay the convergence process to the optimal solution.

We apply the concept of *unsupervised learning* to estimate the transition probabilities based on previous observations. In particular, we rely on the well-known theory of *probabilistic latent variable* [15]. Let us consider $w \in \mathcal{W}$ as the stochastic latent variable denoting the current channel quality level. Formally, we redefine the transition probability of the above-described DTMC as

$$\rho_{a,b}^g = Pr(X_t = S_{g,b} \mid X_{t-1} = S_{g,a}, g = w) \quad (4)$$

that is the probability to move from state $S_{g,a}$ to $S_{g,b}$ when the channel level is exactly $g = w$. To easily understand this, note that $\rho_{0,1}^g = m_g$, $\rho_{1,0}^g = l_g$ whereas $\rho_{0,0}^g$ and $\rho_{1,1}^g$ are the probabilities to stay within the same state $S_{g,0}$ and $S_{g,1}$, respectively. We use an expectation maximization technique to estimate such probabilities. To this aim, we enumerate the transitions between a and b upon g in $h_{a,b}^g$ based on the number of times X_t switches to another state (or stays within the same state) between t and $t+1$. We then derive the *a posteriori* probability as follows

$$Pr(g = w \mid X_t = S_{g,b}, X_{t-1} = S_{g,a}) = \frac{Pr(X_t = S_{g,b} \mid X_{t-1} = S_{g,a}, g = w) Pr(g = w)}{\sum_{z\in\mathcal{W}} Pr(X_t = S_{g,b} \mid X_{t-1} = S_{g,a}, g = z) Pr(g = z)}, \quad (5)$$

and the likelihood probability as the following

$$Pr(X_t = S_{g,b} \mid X_{t-1} = S_{g,a}, g = w) = \frac{\sum_{g\in\mathcal{G}} h_{a,b}^g Pr(g = w \mid X_t = S_{g,b}, X_{t-1} = S_{g,a})}{\sum_{\{\alpha,\beta\}\in\{0,1\}^2} \sum_{g\in\mathcal{G}} h_{\alpha,\beta}^g Pr(g = w | X_t = S_{g,\beta}, X_{t-1} = S_{g,\alpha})} \quad (6)$$

and

$$Pr(g = w) = \frac{\sum_{\{\alpha,\beta\}\in\{0,1\}^2} h_{\alpha,\beta}^g Pr(g = w \mid X_t = S_{g,\beta}, X_{t-1} = S_{g,\alpha})}{\sum_{\{\alpha,\beta\}\in\{0,1\}^2} h_{\alpha,\beta}^g} \quad (7)$$

The above system of equations can be solved using an iterative method that yields $\rho_{a,b}^g$. Finally, we calculate the weight of each latent variable based on a given set of previous observations as per the following equation

$$\omega(w \mid \hat{\mathcal{S}}_i) = \frac{\sum_{\{\alpha,\beta\}\in\hat{\mathcal{S}}_i} \rho_{\alpha,\beta}^w}{\sum_{g\in\mathcal{W}} \sum_{\{\alpha,\beta\}\in\hat{\mathcal{S}}_i} \rho_{\alpha,\beta}^g}, \quad (8)$$

where $\hat{\mathcal{S}}_i$ denotes the history of transitions (or lack thereof) across X_t among different states belonging to level 0 or 1 in the DTMC depicted in Fig. 3. We can generalize the probability to move from a state wherein the latency is under control $S_{g,0}$ to a state incurring unexpected latency $S_{g,1}$, i.e., exceeding the threshold defined in the slice SLA, using the following expression

$$\rho_{a,b} = Pr(X_{t+1} = S_b | X_t = S_a, \hat{\mathcal{S}}_i) = \sum_{w\in\mathcal{W}} \omega(w \mid \hat{\mathcal{S}}_i) \rho_{a,b}^w. \quad (9)$$

In the next section, we design a control-theory process by means of a Markov Decision Process (MDP) that optimally selects the best slice configuration y_i based on the probability to exceed the access latency constrained by the slice SLA.

V. Markov Decision Process

We model the decision problem as a Markov Decision Process (MDP) defined by the set of states $\Sigma = \{\sigma\}$, the set of actions $\Phi = \{\phi\}$, the transition function $T(\sigma, \phi, \sigma')$, and the reward function $R(\sigma, \phi)$. The set of states accounts for all the radio resource splitting options among different tenants, namely *slicing configuration* $c_\sigma = \{y_1, y_2, \ldots, y_i, \ldots, y_I\}$ expressed in terms of PRBs, where $\sum_{i \in \mathcal{I}} y_i = C$, i.e., the overall capacity is exactly split between running slices. We assume that each slicing configuration is issued at every decision epoch n. The transition function characterizes the dynamics of the system from state σ to state σ' through action ϕ. Analytically, $P(\sigma' \mid \sigma, \phi)$ is the probability to visit state σ' given the previous visited state σ and the action ϕ. Finally, the function $R(\sigma, \phi)$ measures the reward associated to the transition from the current state σ performing action ϕ. We shall consider an MDP with an infinite time horizon. Future rewards will be discounted by a factor $0 < \chi < 1$ to ensure the total reward obtained is finite.

When dealing with MDPs is common practice to define a "policy" for the decision agent, namely a function $\mathcal{P}^{(n)} : \Sigma^{(n)} \rightarrow \Phi^{(n)}$ that specifies which action ϕ to perform at time n when in state σ. As soon as the Markov decision process is combined with a defined policy, this automatically fixes the next action for each state so that the resulting combination exactly behaves similarly to a Markov chain. The final aim of the decision agent is to find the policy that maximizes the expected total reward, or, equivalently, to discover the policy $\mathcal{P}^*$ that maximizes the value function.

A. Reward Definition

Each state (or slicing configuration) is associated with a reward value that influences the agent during the decision process. The rationale behind is that we need to bind the action reward to the probability of exceeding the latency constraints defined in the slice SLA. In the following, we introduce the reward function used in our experiments with a detailed overview of its behavior.

Given a slicing configuration $c_\sigma = \{y_i \mid i \in \mathcal{I}\}$, we can analytically build a Discrete-Time Markov Chain, as described in Section III. If the associated transition probability matrix P is perfectly known, we can also derive the steady-state probabilities $\Pi^* = \{\pi_s^*\}$ to be within any single state using Eq. (3). Thus, we can compute the probability to have the access latency of our system under control. This can be used to formulate the instantaneous reward value

$$R(\sigma^{(n)}, \phi^{(n)}) = \left(\sum_{s \in \mathcal{S}_g, 0} \pi_s^* \right)^\eta \tag{10}$$

where s is the index of all states $S_{g,0}, \forall g \in \mathcal{G}$ such that the slice latency is under control, whereas $\eta \in [0, 1]$ is an adjustable value decided by the infrastructure provider to provide action fairness in the reward function when η tends to 0, or maximum likelihood of keeping latency under control when η tends to 1. Then our objective is to maximize the expected aggregate reward obtained as $\lim_{N \rightarrow \infty} \sum_{n=1}^{N} \mathbb{E}\left[\chi^n R\left(\sigma^{(n)}, \phi^{(n)}\right)\right]$. However, given the fully-connected structure of our Markov Decision Process, i.e., all states are reachable from any MDP state, our objective is equivalent to maximize the instantaneous reward given by (10) at each decision epoch n.

Nonetheless, the assumption of perfect knowledge on the transition probability matrix P might be not realistic. Therefore, we need to rely on the transition probabilities $\rho_{a,b}$ inferred based on the previous observations, as explained in Section IV, Eq. (9). The larger the set of observations, the higher the accuracy of our probability estimation and the higher the reward attained to the instantaneous best action taken by the MDP.

B. Complexity Analysis

Once we have fully characterized our proposed MDP, we can solve it by using dynamic programming solutions such as Value Iteration [16]. These approaches require exploring the entire state space of the MDP (several times) and the associated rewards. Let us consider a scenario with I online slices running in our system. Assume that each slice configuration y_i can take values from integer multiples of a minimum PRB chunk size Θ and that the slicing configuration must be consistent, i.e., $\sum_{i \in \mathcal{I}} y_i = C$. Then, we can calculate the overall number of states equal to $\frac{(\frac{C}{\Theta}+I-1)!}{(I-1)!\frac{C}{\Theta}!}$. This poor state scalability, as well known as *the curse of dimensionality*, compromises the feasibility of MDP models under practical conditions. However, MDPs provide insights regarding the structure of the problem itself and are very helpful to design ausiliary solutions, such as Multi-Armed Bandit (MAB) models, which are better suited for functional deployments. Therefore, in the next section we rely on a novel MAB design that exploits information from the underlying MDP to expedite the learning process while attaining near-optimal results.

C. Multi-Armed Bandit Problem

The online decision-making problem has been addressed in the past with several mathematical tools [17]. The limited information about real-time channel quality and effective traffic demand forces the operator to choose, like a gambler facing diverse options to play, the number of radio resources to assign to each running slice. This automatically falls in the fundamental *exploration-vs-exploitation* dilemma: the gambler needs to carefully balance the exploitation operations on known slicing configurations that provided the best revenues in the past against the exploration of new slicing configuration that might eventually produce higher revenues.

This class of decision process can be formulated as a Multi-Armed Bandit (MAB) problem, which emulates the action of selecting the best (single) bandit (or slot machine) that may return the best payoff. Each slot machine returns unpredictable revenues out of fixed statistical distribution, not known *a priori*, that is iteratively inferred by previous observations. This matches well the randomness of the channel quality and the traffic demand we aim to capture whereas each bandit can be mapped onto a state of the MDP, i.e., a specific slicing configuration. The final objective of such a problem is

to maximize the overall gain after a finite number of rounds. This class of problems is usually assessed by a defined metric called *regret* Ω, which is defined as the difference between the reward that can be gained by an optimal oracle, i.e., using an optimal policy that knows the reward distributions *a priori*, and the expected reward of the myopic online policy.

Reusing notation from our MDP model, let us define each arm $\sigma \in \Sigma$ as a different slicing configuration $c_\sigma = \{y_i \mid i \in \mathcal{I}\}$. Once selected, each arm provides an instantaneous reward $R(\sigma)$ defined as the following

$$R(\sigma) = \sum_{i \in \mathcal{I}} \left(\zeta(y_i, \gamma_i) - \frac{\lambda_i}{\Delta_i} \right) \tag{11}$$

where the slicing configuration is $y_i \in c_\sigma$, $\zeta(\cdot)$ computes the number of bits that can be served using y_i configuration and given the current channel quality γ_i, and λ_i is the slice traffic demand, as described in Section II-C.

While using such reward function requires low overhead, as it only needs to calculate the incurred latency after selecting a slicing configuration, it only converges to a near-optimal solution after exploring several configurations, which results in overly long training periods (as shown in Section VI). This is an inherent issue with classic MAB methods, which are *blind* to the underlying system structure. Conversely, in this paper we resort to a novel model-assisted approach that exploits the system model of Section V-A to guide the exploration/exploitation process with (abstract) system information. In this way, as opposed to using the traditional reward model of Eq. (11), we define our bandit's reward as the expectation of access latency exceeding slice SLA defined in Eq. (10). This has a two-fold advantage: i) during the initial training period, the DTMC associated to each state of the MDP is updated (and enhanced) with more accurate values of the transition probabilities: this helps to find steady-state probabilities (and in turn an updated reward per slicing configuration) that reflect the real behavior of our system as time goes on; and ii) the slicing configuration selection accounts directly for stochastic behaviors of both channel quality and traffic demand, while reducing the state space to those that may benefit the entire system. Many algorithms have been proposed to optimally solve the MAB while learning from previous observations [18]. One of the main issues is that collecting rewards on a short-time basis may negatively impact on the decision of the best bandit. Thus, we rely on a modified version of the so-called *Upper Confidence Bound* (UCB) algorithm devised by [19] that overcomes this issue by measuring not only the rewards collected up to the current time interval, but also the *confidence* in the reward distribution estimations by keeping track of how many times each bandit has been selected $z_{\sigma,n}$. The pseudo-code is listed in Algorithm 1.

Initially, we explore all bandits, i.e., slicing configuration $\sigma \in \Sigma$, to get a consistent reward (line 2-6). Then we select the best configuration that maximizes the empirical distribution $\hat{\rho}_\sigma$ accounting for a confidence value. This confidence value depends on the number of times we have explored that particular configuration as well as the accuracy of the transition probabilities we calculate for the associated DTMC. Note that this is different to traditional UCB algorithms. Specifically,

Algorithm 1 LACO

1. **Input:** $\Sigma, N, \Psi = \{\psi(\sigma)\}, \mathcal{I}, \omega, \epsilon, \mathcal{S}$
2. **Initialization:** $z_\sigma, \hat{\rho}_\sigma = 0, \ \forall \sigma$
3. **Procedure:**
4. **for all** $n \in N$ **do**
5. **if** $n = 0$ **then**
6. **for all** $\sigma \in \Sigma$ **do**
7. GET reward: $\hat{\rho}_\sigma = R_\sigma^{(n)}$
8. $z_\sigma = z_\sigma + 1$
9. **end for**
10. **else**
11. $\sigma^* = \arg\max_{\sigma \in \Sigma} \hat{\rho}_\sigma + \psi(\sigma) \sqrt{\frac{2 \log \sum_k z_k}{z_\sigma}}$
12. UPDATE $\hat{\rho}_{\sigma^*} \leftarrow R_{\sigma^*}^{(n)}$
13. $z_\sigma = z_\sigma + 1$
14. **end if**
15. **for all** TTIs $\in \epsilon$ **do**
16. **for all** $i \in \mathcal{I}$ **do**
17. UPDATE $\omega(w \mid \hat{\mathcal{S}}_i) \leftarrow \mathcal{S}_i$
18. **end for**
19. **end for**
20. UPDATE $\psi(\sigma^*) \leftarrow \omega(\cdot)$
21. **end for**
22. **End Procedure**

we define a Markov accuracy value $\psi(\sigma) = (\frac{(\sum_w \omega(w|\hat{\mathcal{S}}_i))^2}{W \sum_w \omega(w|\hat{\mathcal{S}}_i)^2})$, where W represents the cardinality of the set $\mathcal{W}$. Note that $\psi(\sigma)$ depends on the weights $\omega(\cdot)$ obtained through the performed observations $\hat{\mathcal{S}}_i$, as reported in Eq. (8). Interestingly, $\psi(\sigma) \in (0, 1]$, i.e., when the DTMC has no relevant observations to build its transition probabilities this function returns $\psi(\sigma) = 1$ whereas, when a relevant number of observations allow to determine accurate transition probabilities, its value tends to 0. The value of $\psi(\sigma)$ is updated at the end of every decision interval (line 20) after monitoring the effects of the last decision on the Markov Latent variable distribution (lines $15 - 19$).

D. Regret Analysis

Here, we mathematically calculate the bounds of our solution, LACO, for multi-armed bandit problems. Let us consider a player selecting an arm $\sigma \in \Sigma$ every decision epoch n. Every time arm σ is pulled down, it returns a reward $R_\sigma^{(n)}$ drawn from an unknown distribution with mean $\bar{\rho}_\sigma$ and empirical mean value calculated until time n as $\hat{\rho}_\sigma^{(n)} = \frac{\sum_{s=1}^n R_\sigma^{(s)}}{n}$. We denote σ^* as the arm providing the maximum average reward such that $\bar{\rho}_{\sigma^*} > \bar{\rho}_\sigma, \forall \sigma \neq \sigma^*$. If the arm selection is performed using LACO, it yields that the regret is obtained as

$$\begin{aligned} \Omega_N^{\text{LACO}}(\Sigma) &= N \bar{\rho}_{\sigma^*} - \mathbb{E}[\sum_{n=1}^{N} R_\sigma^{(n)} \mid \sigma \in \mathcal{P}^{\text{LACO}}] \\ &= N \bar{\rho}_{\sigma^*} - \sum_{\sigma=1}^{\Sigma} \bar{\rho}_\sigma \mathbb{E}[z_\sigma^{(n)}]; \end{aligned} \tag{12}$$

where $\mathcal{P}^{\text{LACO}} = \{\sigma_n\}$ is the policy as defined in Section V that consists of a set of moves that LACO will play at time

n whereas $z_{\sigma,n}$ is the overall number of decision epochs arm σ has been pulled down till time instant n. Now consider LACO as a uniformly good policy, i.e., any suboptimal arm $\sigma \neq \sigma^*$ is chosen by our policy up to round n so that $\mathbb{E}[z_{\sigma,n}] = o(n^\alpha), \forall \alpha > 0$. It holds that

$$\lim_{\mathrm{N}\to\infty} \sum_{\sigma=1}^{\Sigma} N^{-1} \bar{\rho}_\sigma \mathbb{E}[z_\sigma^{(N)}] = \Sigma \bar{\rho}_{\sigma^*}. \tag{13}$$

Hence, we can express the regret lower bound as the following

$$\lim_{\mathrm{N}\to\infty} \inf \frac{\Omega_N^{\mathrm{LACO}}(\Sigma)}{\log N} \geq \sum_{\sigma:\ \bar{\rho}_\sigma < \bar{\rho}_{\sigma^*}} \frac{\bar{\rho}_{\sigma^*} - \bar{\rho}_\sigma}{Div(\bar{\rho}_\sigma, \bar{\rho}_{\sigma^*})} \tag{14}$$

where $Div(\bar{\rho}_\sigma, \bar{\rho}_{\sigma^*})$ is the Kullback-Leibler divergence of one statistical distribution against the other and it is used to measure how one distribution might diverge from another probability distribution.

Now consider the Hoeffding's inequality for multiple i.i.d. variables x_n with mean μ. It yields that $Pr(|\frac{\sum_{i=1}^{n} x_i}{n} - \mu| \geq \delta) \leq 2e^{-2n\delta^2}$. Our algorithm LACO applies an upper confidence interval $\delta = \sqrt{\frac{2 \log \sigma_k z_k}{z_\sigma}}$. Therefore, it yields that

$$Pr\left(|\hat{\rho}_{\sigma,n} - \bar{\rho}_\sigma| < \sqrt{\frac{2 \log \sum_k z_k}{z_\sigma}} \right) \geq 1 - \frac{2}{n^4} \tag{15}$$

and also that

$$Pr\left(\mathcal{P}^{(n+1)} = \sigma \mid z_\sigma^{(n)} > \frac{4 \log n}{\bar{\rho}_{\sigma^*} - \bar{\rho}_\sigma} \right) \leq \frac{4}{n^4}. \tag{16}$$

We can then derive the expectation of number of times sub-optimal arm $\sigma \neq \sigma^*$ is pulled down as follows

$$\mathbb{E}[z_\sigma^{(N)}] \leq \frac{4 \log N}{\bar{\rho}_{\sigma^*} - \bar{\rho}_\sigma} + 8 \tag{17}$$

and the regret upper bound as the following

$$\mathbb{E}\left[\Omega_N^{\mathrm{LACO}}(\Sigma)\right] \leq \sum_{\sigma:\ \bar{\rho}_\sigma < \bar{\rho}_{\sigma^*}} \frac{4 \log N}{\bar{\rho}_{\sigma^*} - \bar{\rho}_\sigma} + 8\left(\bar{\rho}_{\sigma^*} - \bar{\rho}_\sigma\right). \tag{18}$$

VI. Performance Evaluation

In this section, we evaluate our solution through an exhaustive simulation campaign that takes into account complexity, revenue and SLA violation metrics.

A. Simulations Setup

To assess heterogeneous slices, we simulate the network load demand of slice i at each time-slot (i.e., each transmission time interval (TTI) in Long Term Evolution (LTE) systems) by extracting a random value from a Normal distribution $\mathcal{N}_i(\mu_i, \nu_i^2)$, where μ_i and ν_i represent the mean value and standard deviation, and let L_i describe its latency constraint. Moreover, we model the SNR channel variation as another random variable drawn by a Rayleigh distribution and derive the probability distribution encompassing the whole SNR range. For every channel instantiation, we extract the corresponding Modulation and Coding Scheme (MCS) as defined by the 3GPP standard.[9] The MCS index $m \in \mathcal{M}$ combines one possible modulation scheme and a predefined coding rate providing a compact way to represent a simple concept: the better the radio conditions, the more bits can be transmitted per time unit, and *vice versa*. Fixing the channel bandwidth, the expected average throughput achievable by one slice during one epoch depends on both the modulation and coding schemes used and, most importantly, on the number of PRBs reserved for the slice. In a wider timescale,[10] the average capacity can be approximated as $C_i = \left(\sum_m^M \Gamma_m \pi_{m,i}\right) T_i y_i$ where Γ_m represents the average number of bits per LTE subframe that can be transmitted using the m-th MCS index, $\pi_{m,i}$ is the steady-state probability distribution output of the first stage Markov chain model, T_i defines the decision interval size, and y_i accounts for the number of PRBs allocated to the i-th slice. We refer the reader to Table I. In the LTE radio interface, the maximum amount of PRBs is fixed to 100 when operating at conventional bandwidth values of 20 MHz. In order to support massive type communication and Ultra-Reliable Low-Latency Communication (URLLC) use-cases, the 5G New Radio (NR) introduces significant enhancements in the radio frame composition. Not only 5G NR will support wider channel bandwidth (up to 100 MHz), but also introduce the support for multiple different types of subcarrier spacing. For back-compatibility reasons, even in 5G NR the time duration of radio frames and subframes are fixed to 10 ms and 1 ms, respectively [20]. The number of slots within each subframe however would change according to the subcarrier configuration, which eventually translates in shorter PRB time duration and thus a different PRB availability depending on the selected configuration. It must be noticed that all the subcarrier spacing are defined as $\Delta f = 2^j \cdot 15$ KHz, $j = \{0, \ldots, 4\}$, thus leading at the definition of time-frequency grids containing an amount of PRBs which is multiple of those contained in the traditional LTE grids. In this context, we assume a simple mapping function, as the one described in [21], implemented at intra-slice scheduler to homogenize the resources of potentially heterogeneous radio access technologies.

Traffic demands are compared with the current channel availability to derive the possibilities to pass from one state to another. It must be noticed that the accuracy of the resulting steady-state distribution strictly depends on the precision of such comparison. For this reason, we constantly monitor and update the transition probabilities of the Markov chain based on the resource allocation adopted in the current decision interval. During the arm selection, if the chosen configuration does not provide enough resources to meet the latency requirements, the steady-states will be mostly distributed in the lower part of the Markov chain leading to a minor reward that, in turn, guides the MAB agent to take a different action (i.e., selecting a different arm) in the following decision round.

For benchmarking purposes, we implement two widely used MAB algorithms, namely "legacy" UCB and Thompson

[9] We refer the reader to [8] for an exhaustive explanation of the mapping between SNR and MCS.

[10] Note that we assume a timescale larger than our epochs used in the decision-making process.

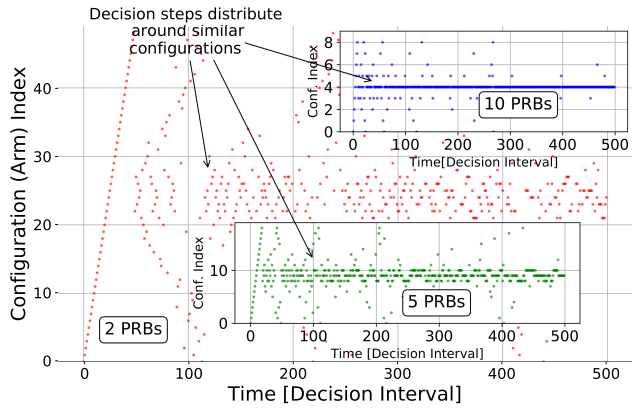


Fig. 4. Impact of different resource allocation chunk sizes.

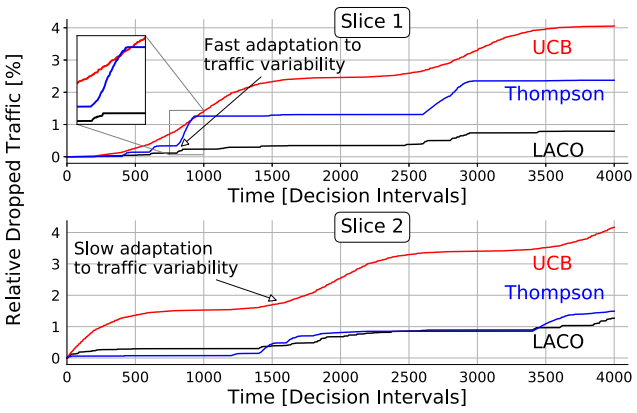


Fig. 5. Cumulative dropped traffic due to latency constraints violations.

Sampling (TS).[11] On the one hand, UCB adopts a deterministic approach to deal with the exploration-vs-exploitation dilemma, but its performance generally degrades as the number of arms increases. On the other hand, Thompson sampling adopts a probabilistic approach that scales better with the number of arms, but it may provide sub-optimal results when the distribution of reward changes over time (i.e., in non-stationary scenarios). Conversely, LACO combines the advantages of them both by adopting a probabilistic model (MDP) guiding an exploration phase derived from UCB.

B. Multi-Armed Bandit Problem Behavior

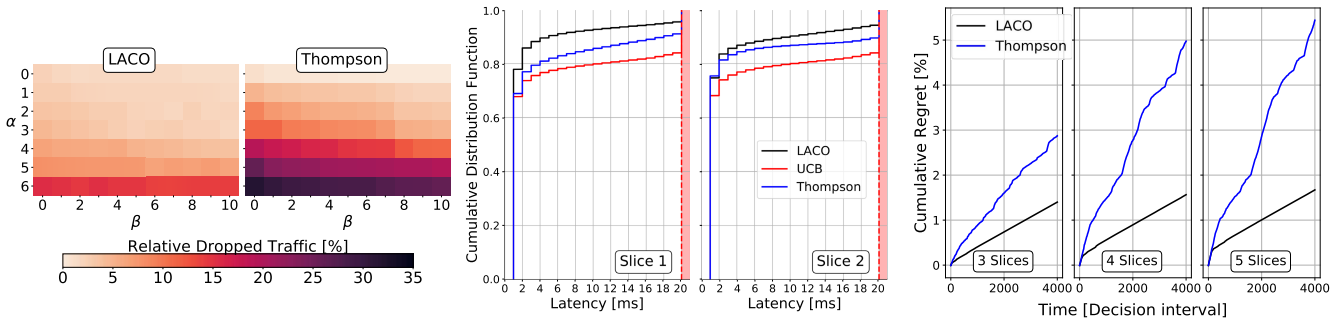
We first explore the trade-off between action space (and its granularity) and the associated reward loss. To this aim, we set up a simple experiment with 2 slices with equal SLA requirements in a deterministic and static environment. We then apply LACO using 3 different action sets: $\{0, 2, 4, \ldots, 100\}$, $\{0, 5, 10, \ldots, 100\}$ and $\{0, 10, 20, \ldots, 100\}$ PRBs (with 50, 20 and 10 available configurations each), labelled "2 PRBs", "5 PRBs" and "10 PRBs", respectively. The results, shown in Fig. 4 make it evident that the higher the granularity the longer the exploration phase(s): over 50 intervals for "2 PRBs" whereas it takes around 10 intervals for "10 PRBs". Interestingly, the loss in reward attained to the latter configuration is only 2%. Therefore, due to a faster convergence time at the expense of minimal reward loss, we empirically select $\Theta = 10$ PRBs for our purposes.

C. Slice SLA Violation Analysis

We thus grant spectrum-time resources in the granularity of chunks of 1 second $\times$ 10 PRBs. In the first scenario, we investigate the capacity of LACO to adapt the resource allocation at variable traffic loads. For this reason, we consider only two slices with equal requirements, i.e., $\nu_i^2 = 10$ Mb/s and $\Delta_i = 20$ ms for $i = 1, 2$. To assess real scenarios with non-stationary traffic patterns, we vary the mean load of each slice i following a sinusoidal curve in counter-phase between $\mu_i = 8$ Mb/s and $\mu_i = 40$ Mb/s. This forces the resource allocation process to span across the whole configuration set when dealing with SLAs guarantees. As shown in Fig. 5, the cumulative dropped traffic of each slice changes when different MAB algorithms are used. The behaviour of UCB shows high variability after few decision intervals. As soon as all the arms are selected, the agent starts learning about the statistics of the outcomes and builds a ranked list. The need for a comprehensive knowledge of all the arms leads to several "bad" choices during the exploration phase. This slows down the convergence to the optimal configuration and penalizes performance. From the obtained rewards, TS builds a bivariate probability distribution across the expected reward of each arm, extracts a random sample and chooses the arm associated to the maximum value. This approach performs well in static scenarios as TS favours exploitation of the empirical results obtained in the first attempts; but in time-varying scenario as the one we are considering, the reward distribution associated to each arm fluctuates over time rendering TS unable to adapt fast enough in highly-dynamic scenarios. In contrast, the LACO's model-awareness allows for quicker convergence and so it accommodates real-time traffic requirements in dynamic environments and as a result reduces the amount of data violating delay deadlines.

Obviously, heterogeneous throughput/latency requirements impact the system differently. Fig. 6a shows the effect of such variations on the system extending the previous scenario and considering increasing values of resource requirements as $10 \cdot \alpha$ Mb/s, and $10 \cdot \beta$ ms, respectively. As expected, smoother delay requirements (horizontal direction in the figure) allow to serve more traffic within the latency bounds defined by the SLA, although the impact becomes negligible after few incremental steps. This is due to long decision intervals when compared to the timescale of fast channel variations. A proper resource configuration selection allows to match the offered traffic requirements with the expected channel capacity, allowing the incoming traffic to be served within few milliseconds. As the offered traffic approaches the channel capacity boundary (vertical direction in the figure), the same task becomes more challenging and the admission and control process should consider this aspect when granting/rejecting access to new network slices. LACO 's abilities to adapt to

[11] Due to space limits, we refer the reader to the literature introducing such algorithms, e.g. [22].



(a) Effects of different slice reqs. (b) CDF of experienced latency. (c) Empirical cumulative regret.

Fig. 6. (a) Effects of different slice requirements; (b) CDF of latency experienced by served traffic; (c) Empirical cumulative regret for a variable number of slices.

demand variations not only mitigates the amount of traffic violating delay requirements but also improves the distribution of data delivery delay overall. As shown by Fig. 6b, the empirical CDF of delay for each slice in the same scenario presented above remarkably improved with a mean delay equal to 2.6, 3.9 and 4.9 ms for LACO, TS and UCB, respectively.

Finally, we implement an optimal offline policy with full knowledge of the system, i.e., an oracle policy that knows the future with the corresponding latency violations. We compare both LACO and TS to this optimal policy for a variable number of slices. The aggregated demand is adapted to ensure we operate within the system capacity. In Fig. 6c, we depict the temporal evolution of the cumulative reward loss over time (regret) for both approaches. The figure illustrates how the regret increases with time much rapidly for TS, a difference that increases with the number of slices.

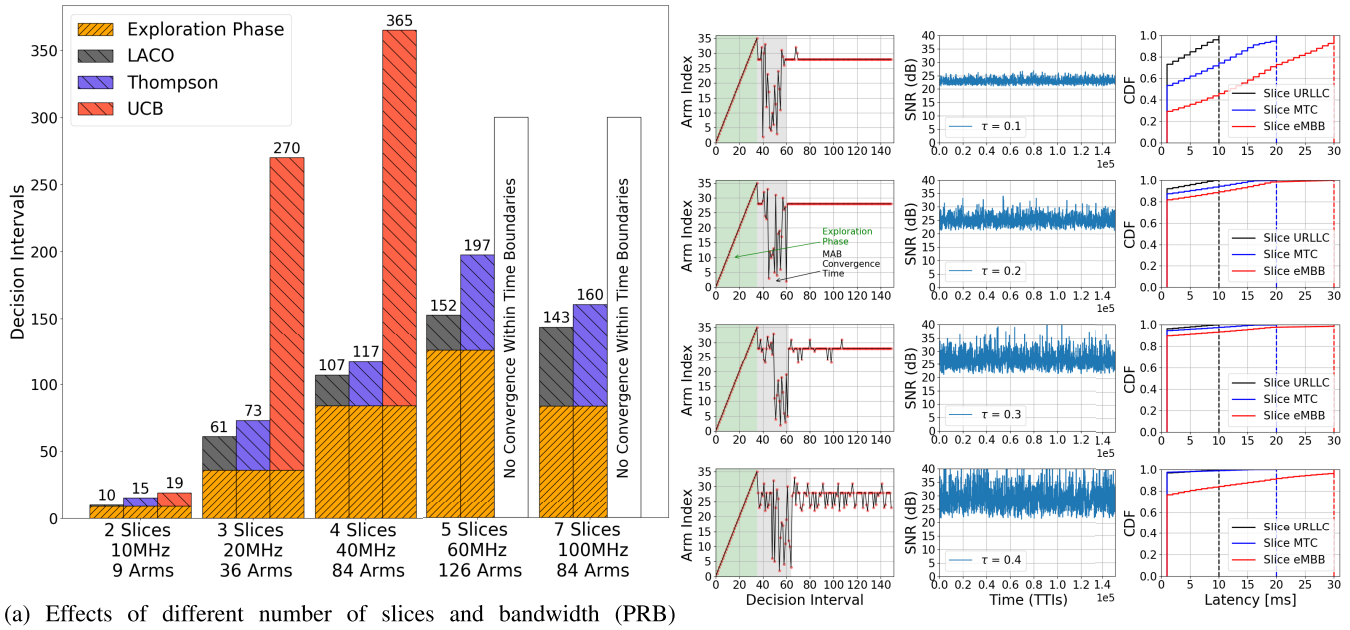
D. Convergence time

The next generation of mobile networks (5G) promises to support the provisioning of high throughput and low-latency services even in highly dense scenarios [2]. These capabilities are tightly bounded with the possibility to exploit higher communication frequencies together with wider spectrum bandwidth. In the 5G context, bandwidth is expected to increase up to 100MHz, leading to additional complexity in the management of radio resources. In order to assess LACO performances in such scenarios, we investigate the convergence time of our solution to the optimal slice configuration in different bandwidth settings. To enable more efficient use of the spectrum resources and reduce the power consumption at UE side, 5G New Radio (NR) introduces the concept of bandwidth parts (BWP) [20], where each BWP can be configured by different numerologies defining specific signal characteristic, e.g., in terms of subcarrier spacing. Without loss of generality, we assume all the end-users belonging to the same slice operating under similar numerology settings. Moreover, we keep the subcarrier spacing fixed to $\Delta f = 15$ KHz as in legacy LTE systems. Such coarse resource allocation scheme is mandatory to support LTE devices but, it can be easily mapped to finer resource block structures as defined within the 5G domain at lower layer intra-slice schedulers [21].

Fig. 7a compares the convergence time of different MAB algorithms for an increasing number of slices and bandwidth availability over a time period of $N = 1000$ decision intervals. It should be noticed that depending on channel statistics and real-time slice requirements, *multiple* resource allocation settings (namely arms) may provide *optimal performance* making unfeasible a single convergence point. Thus, we opted to simulate the worst-case scenario allowing for a unique optimal resource configuration in each simulation run. In line with previous observations, we fix $\Theta = 0.1\,C$. Despite a common initial exploration phase (highlighted in orange), from the picture it is evident how the curse of dimensionality affects the overall convergence time. This is more evident for the legacy UCB approach (depicted in red), which hardly copes with the increasing size of the action space and in some runs did not converge to a solution within the time boundary of our experiment. Focusing on LACO performances (depicted in black), the number of decision intervals necessary to converge to the optimal resource allocation outperforms Thompson Sampling (in blue) by scaling almost linearly with the number of slices (and PRB availability) after the initial exploration phase.

Convergence to the optimal slice configuration also shows its dependency on the radio channel statistics. To measure the sensitivity of the decision process at the SNR fluctuations, Fig. 7b considers a fixed number of slices (i.e., 3) deployed in a system characterized by average channel statistics with an increasing variance. In every scenario, the average (per slice) channel realization is derived from a Rayleigh distribution characterized by a scale parameter $\tau = \{0.1, 0.2, 0.3, 0.4\}$, respectively. This introduces an increasing level of variability in the SNR distribution according to the formula $Var = \frac{4-\pi}{2}\tau^2$, as depicted in the plots of the central column. On the left-hand side of the same picture, it can be noticed how higher SNR variability has very limited impact on the decision steps. This feature is inherited by the Markov Chain model described in Section III. In particular, provided that the slice requirements fit within the admissibility region of the system, a higher SNR variability will simply map into a wider excursion over the Markov chain steady states without affecting the final reward of the same arm.

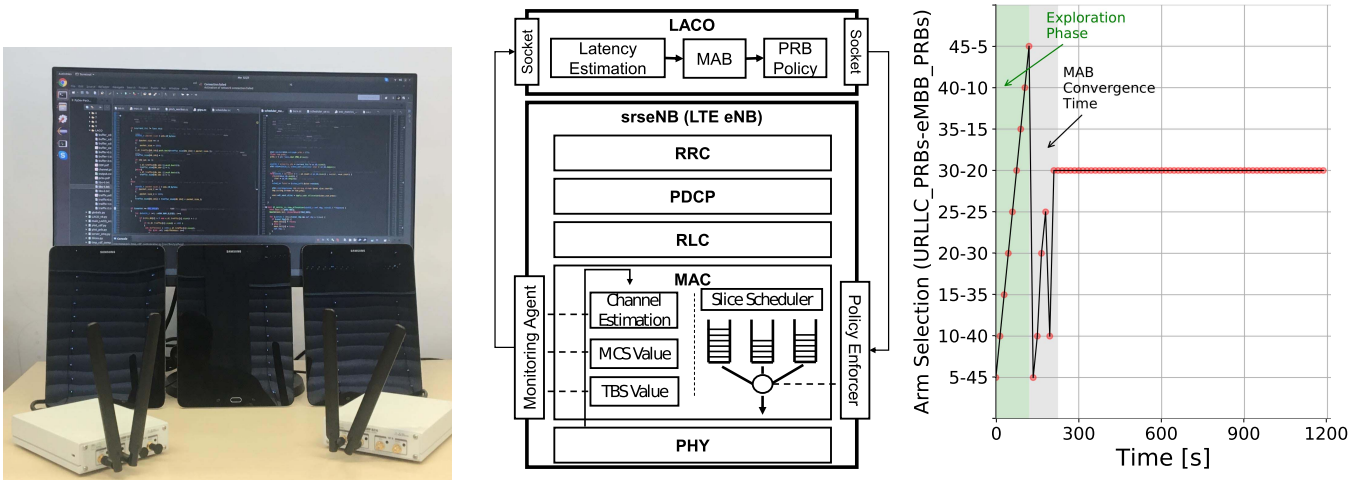
Finally, on the right-hand side of the picture, we depict the empirical CDFs of the overall latency occurred per slice.



(a) Effects of different number of slices and bandwidth (PRB) availability.

(b) Effects of increasing variability in the channel conditions (SNR).

Fig. 7. Sensitivity analysis of bandwidth availability and SNR variability on the convergence time to the optimal slice resource allocation.



(a) Experimental setup.

(b) Architecture overview.

(c) Arm selection.

Fig. 8. Experimental setup (a); Architecture overview (b); Arm selection over time (c).

In (almost) static channel conditions, slices' latency distribution suffer from having poor channel conditions, which are barely sufficient to support requested data volumes. In this context, slices with less stringent delay requirements, namely the MTC and eMBB, are lightly penalized to meet the expected latency threshold w.r.t. the URLLC one. When increasing the channel variability, the average channel conditions improve easing the allocation resource task thus favouring the satisfaction of overall latency requirements.

VII. Experimental Proof of Concept

In order to illustrate, validate and analyze the performance of our LACO solution, we developed it as a standalone software module running on top of an open source platform that implements the LTE protocol stack, namely `srsLTE` [23], attached to a USRP[12] Software-Defined Radio (SDR) device as radio front-end. Our testbed is depicted in Fig. 8a and consists of one LTE eNB (a modification of `srseNB`) and commercial Android tablets[13] as UEs. Any single UE emulates the aggregated traffic of multiple UEs within one slice. We use `mgen`[14] to generate different downlink traffic patterns. Due to our LTE spectrum testing license restrictions, we use 10 MHz bandwidth in LTE band 7 and use SISO configuration for

[12]USRP B210 from National Instruments/Ettus Research (https://www.ettus.com/all-products/UB210-KIT/).

[13]Samsung Galaxy Tab S2 (https://www.samsung.com/de/tablets/galaxy-tab-s2-9-7-t813/SM-T813NZKEDBT/).

[14]mgen (https://www.nrl.navy.mil/itd/ncs/products/mgen).

simplicity. This renders a maximum capacity of ~ 36 Mb/s with highest SNR. Finally, in accordance with the findings described in Section VI-B, we set the minimum PRB allocation value at 10% of the overall availability.

A. Implementation

The architecture of our software implementation and LACO's interfaces with `srseNB` are depicted in Fig. 8b. LACO interacts with the eNB's Medium Access Control (MAC) layer to implement two key features:

- **Monitoring agent**. This feeds LACO with real-time SNR reports generated by the physical (PHY) layer from feedback received from the UEs, the selected MCSs and corresponding transport block size (TBS) value used to encode information at the MAC layer, and other traffic statistics such as packet size and arrival times;
- **Policy Enforcer**. This allows LACO to dynamically enforce the PRB allocation policies calculated by our MAB model, as described in Section V-C.

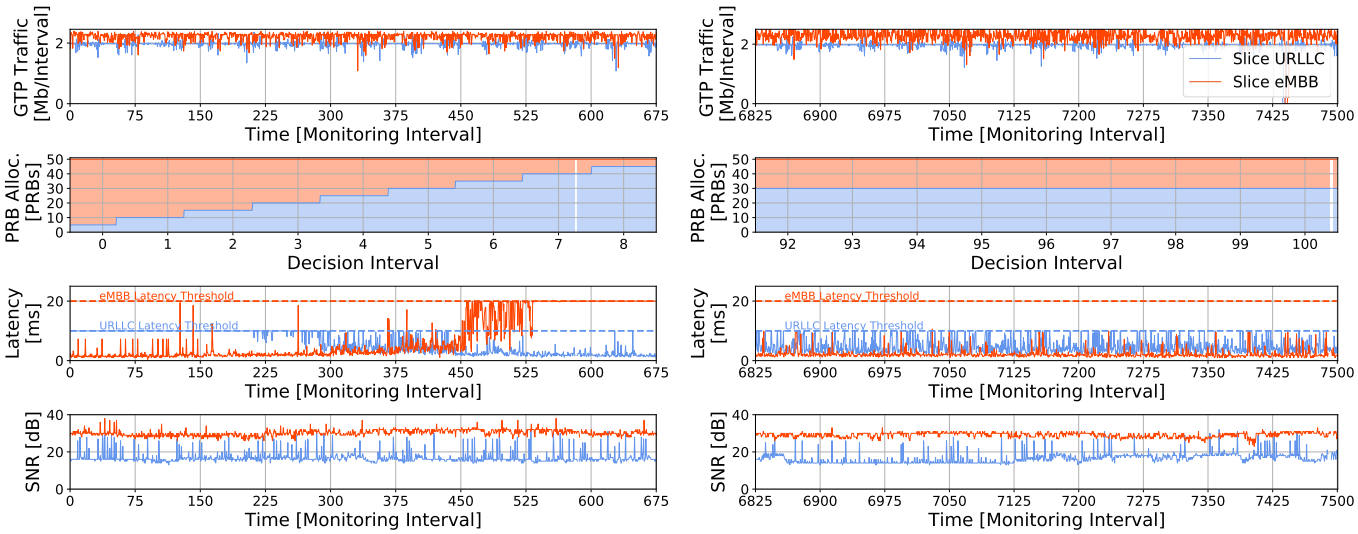
The main feature of our implementation is the possibility to collect, with TTI granularity, the traffic arrival rate and the TBS values to be used in each transmission frame. This information, together with the scheduling buffer size and data arrival times, is essential to compute the latency experienced by the different slices running in the system.

The different metrics are collected in a time series database, namely `InfluxDB`, and periodically reported to LACO which constructs a virtual queue (one per slice) to track the dynamics of packets arriving at the eNB, from their entrance into the scheduling buffer to their transmission. This approach is particularly useful as Internet Protocol (IP) packets are multiplexed while advancing the transmission path in the eNB, complicating the computation of slice latencies by external modules. Our approach aims to characterize the PRB allocation policy currently enforced into the system. In case of constant traffic and low latency requirements for example, poor channel conditions will result in lower TBS values and a sudden increase of the virtual queues size. Such event directly maps into an additional delay suffered by IP packets at the Radio Link Control (RLC) layer. Note that higher packet rates also lead to larger waiting times, which might result in exceeding slices SLAs boundaries. In such cases, the violation of pre-defined SLA latency boundaries triggers the DTMC model described in Section III to a *delay state* and the selected PRB allocation policy is assigned with a lower reward value. Conversely, in a stable system where serving rate and packet arrival rate are balanced, the size of the virtual queues get smaller and the DTMC model is mostly characterized by *non-delay states*.

B. Experimental Results

We consider a scenario accounting for two slices characterized by the following requirements. The first slice (labelled Ultra Reliable Low Latency Communications or URLLC) demands $\Delta_{URLLC} = 10$ ms communication delay and is characterized by a constant bit rate equal to 9.6 Mb/s. The second slice (labelled enhanced Mobile Broaband or eMBB) is characterized by a constant throughput equal to 11.2 Mb/s with a more relaxed latency requirements $\Delta_{eMBB} = 20$ ms. We set LACO's decision interval to 15 seconds and let our experiment run over the downlink direction for 100 decision intervals. Fig. 8c shows the evolution of the PRB allocation configuration decisions taken by LACO over this time span and how fast the convergence to a suitable layout is achieved. The monitoring information about incoming traffic at GTP level collected during the experiment are depicted in the upper plots of Figs. 9a and 9b. It should be noticed that these values represent aggregated values (sum) over monitoring intervals of 200ms. Latency and SNR information are depicted in the third and fourth plots of each figure. In this case, we use maximum and average as aggregation functions, respectively.

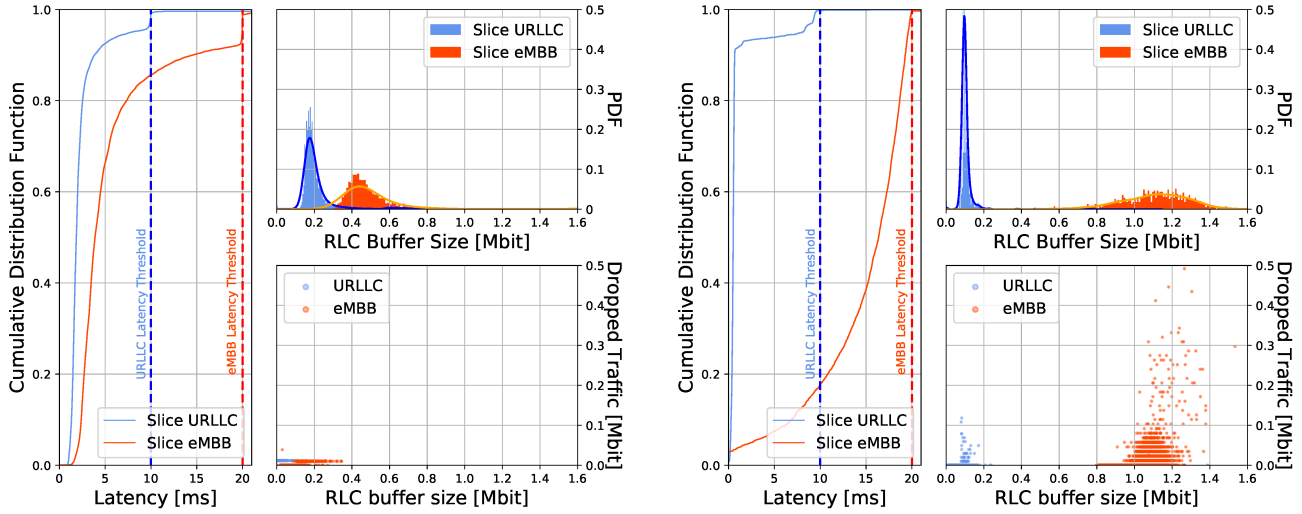
As described in Section V-C, during the starting procedure the MAB algorithm explores all available arms with the aim of collecting an initial feedback on the system dynamics. Fig. 9a depicts the effects of these sequential choices on the latency experienced by the ongoing traffic flows. The initial steps drive the allocation of radio resources towards the eMBB slice thereby providing significant advantages in terms of experienced delay with respect to the URLLC one. In this phase, traffic coming from the URLLC might be dropped due to delay violation Δ_{URLLC}. The scenario changes after the 6-th decision interval, when the agent selects the configuration (35-15). Given the current channel quality, that arm does satisfy the URLLC radio requirements but does not reserve enough radio resources for the eMBB slice, thus increasing the latency experienced by its users. Subsequent arm selections within decision intervals 7 and 8, further reduce the radio resources assigned to the eMBB slice thus leading the traffic to violate Δ_{eMBB}. The MAB agent collects this information and quickly converges to a satisfactory configuration. In Fig. 9b, we focus on the system dynamics once the convergence is achieved and clearly notice how both the latency requirements are satisfied. Interestingly, despite similar traffic requirements, the algorithm selects the configuration (30-20), which assigns more resources to the first slice. This is justified by the lower SNR value experienced by such a slice during the experiment, as depicted in the bottom plot of Figs. 9a and 9b. The URLLC slice thus requires more PRBs to compensate for the lower MCS used during the communication and successfully meets the latency requirements. For illustration purposes, we select a vanilla PRB allocation policy, namely round-robin (RR), as a generic non-latency-aware benchmark and compare the performance of the two schemes running in the same scenario. The results of our experiments are summarized in Fig. 10, where both plots depict the empirical CDF of the latency, the RLC buffer density and the dropping rate incurred by each slice for the two allocation schemes. The performances of the system when LACO is in place are depicted on the left-hand side picture, whereas the right-hand side shows the results of the RR-based slice scheduling scheme. In both plots, the URLLC slice is shown in blue and the eMBB one in orange. Based on these results, we can observe that LACO successfully meets both slices latency requirements. This is achieved by providing the required resources to the URLLC and eMBB slices (Fig. 8c) according to their different



(a) System dynamics during MAB discovery phase.

(b) System dynamics at convergence.

Fig. 9. Comparison of system dynamics during a) discovery phase and b) MAB convergence.



(a) LACO

(b) Round Robin (RR)

Fig. 10. Evaluation of different performance metrics for different scenarios.

latency needs. This results in the URLLC slice allowing SLA latency requirements ($\leq$ 10ms) at a very low average latency cost increase for the eMBB slice. In our experiments, very few traffic ($\sim$ 2%) experienced a latency above the 10 ms target of URLLC when using LACO, in contrast to $\sim$ 10% experienced with RR. Despite of negligible impact, note that by our design choice parts of fragmented packets are sent even if above the latency threshold to avoid long HARQ based retransmission procedures [24], which may negatively affect the slice performance. Moreover, we wish to highlight that for LACO the amount of violations due to the exploration and convergence period could be significantly reduced if desired by introducing a policy aimed at minimizing such cases. The performance gap further increases when comparing the eMBB results. Given that RR sequentially allocates resources to the URLLC slice and, when the buffer is empty, to eMBB, it consistently favours the URLLC slice over the eMBB one. Thus, despite the higher channel quality condition experienced by the eMBB slice, in every scheduling period the resource availability for the eMBB slice is highly reduced. This provides a better performances for URLLC traffic, but at a significant degradation cost for eMBB users, as confirmed by Fig. 10b (bottom-right), which depicts the amount of traffic dropped during the experiment. The latency performance is strongly related with the traffic queue waiting in the transmission buffers. For this reason, the two figures depict the buffer size density distribution obtained during the experiments. It is clear from the comparison how different PRB allocation schemes affect the transmission buffer size at RLC layer. In the LACO case, they are generally lightly loaded, finally providing shorter serving time for incoming packets. In the RR scenario however, the eMBB traffic suffers higher congestion, which

leads to augment packet's waiting time, and consequently increases the rate of latency violations.

VIII. Related Work

The RAN design problem has always been at the forefront of the mobile operators and a vast amount of research has been devoted to novel RAN architectures [25], [26] and efficient radio resource schedulers [21], [27]. Recently, network slicing has been proposed as a new means for mobile operators to deploy isolated network services owned by different customers over a common physical infrastructure. However, as highlighted in [28], RAN needs additional functionalities to fully exploit SDN and NFV principles, specially in the partition and isolation of radio resources. The authors of [3] focus on efficient sharing of the RAN resources and proposed a RAN slicing solution that performs adaptive provisioning and isolation of radio slices. Their work is based on dynamic virtualization of base station resources, which gives tenants the ability to independently manipulate each slice. Although the proposed architecture may guarantee isolation through different control planes, no mechanism is in place to ensure the satisfaction of delay requirements. [29] provides an empirical study of resource management efficiency in slicing-enabled networks through real data collected from an operational mobile network, considering different kinds of resources and including radio access, transport and core of the network. Similarly, the authors of [30] formulate an optimization framework to deal with resource partitioning problem, where inter-slice isolation is assured through a virtualized layer that decouples the reservation choice from the physical resource availability and proposing different abstraction types of radio resource sharing. In [31] the authors present an Earliest Deadline First (EDF) scheduling approach in the context of network slicing. Differently from us, their approach works on a single MAC scheduler and assumes for every TTI a complex fine-tuning of the quota of resources to be assigned to each slice, thus limiting the implementation of dedicated intra-scheduling solutions.

The exploration-vs-exploitation trade-off, typical of Multi-Armed Bandit (MAB) problems is particularly suited to problems that require sequential decision-making. For this reason, a wide set of variations from the classical MAB model has been proposed in the literature [17], [32], together with novel algorithms to address them [33]. In this regard, the work of [34] investigates the MAB problems in case of Markovian reward distribution, where arms change their state in a two-state Markovian fashion. The authors addressed the problem assuming that the Markov chain evolves only when the arm is played, showing that the proposed sample mean-based index policy achieves regret performances comparable to legacy UCB algorithm. The authors of [22] performed a complete regret analysis of the TS algorithm, generalizing the original formulation to distributions other than the Beta distribution. The MAB framework is also applied in [35] to deal with rate adaptation problem in 802.11-like wireless systems. The authors demonstrate that exploiting additional observations significantly improve the system performance. Similarly, [36] deals with scheduling transmissions in presence of unknown channel statistics. The proposed algorithm learns the channels' transmission rates while simultaneously exploiting previous observations to obtain higher throughput. This led to the design of a queue-length-based scheduling policy using the channel learning algorithm as a component in time-varying environment. The authors of [37] presented an algorithm for multivariate optimization on large decision spaces based on an innovative approach combining hill climbing optimization and Thompson sampling. While the scalability of their algorithm has been proven through exhaustive simulations, the framework lacks a complete analysis of regret bounds aimed at demonstrating the impact of hill climbing in combinatorial decision making. Finally, similar to us, [38] deals with an MAB formulation where the reward distributions are characterized by temporal uncertainties. Interestingly, they were able to mathematically capture, in terms of reward, the added complexity embedded in the non-stationarity feature when compared to the legacy framework.

The key novelty of LACO relies on the exploitation of (abstract) information of the underlying system structure to expedite solutions. Conversely, prior works are blind to such type of information and need to spend substantial time exploring very bad decisions before achieving it.

IX. Conclusion

Major efforts in the design of next-generation mobile systems pivot around network slicing and (mobile edge) low-latency services. This paper aims to bridge the gap between them both by designing LACO, a RAN-specific network slice orchestrator that considers network slice requests with *strict latency requirements*. Despite the efforts devoted by 5G researchers and engineers to network slicing, to the best of our knowledge, this is the first radio slicing mechanism that provides *formal delay guarantees*. To make network slicing decisions in environments with varying wireless channel quality and user demands, LACO builds on a learning Multi-Armed Bandit (MAB) method that is *model-aware* as opposed to classic MAB approaches that are blind to information regarding the underlying system. In addition, we exploit information from the system model to expedite the exploration-vs-exploitation process. Our results derived from an implementation with off-the-shelf hardware show that LACO is able to guarantee strict slice latency requirements at affordable computational costs.

References

[1] 5GPPP Alliance. (Mar. 2017). *5G Innovations for New Business Opportunities*. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2017/01/5GPPP-brochure-MWC17.pdf

[2] J. X. Salvat, L. Zanzi, A. Garcia-Saavedra, V. Sciancalepore, and X. Costa-Pérez, "Overbooking network slices through yield-driven end-to-end orchestration," in *Proc. ACM CoNEXT*, Nov. 2018, pp. 353–365.

[3] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 127–140.

[4] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Perez, "A machine learning approach to 5G infrastructure market optimization," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 498–512, Mar. 2020.

[5] H. Lee, J. Flinn, and B. Tonshal, "RAVEN: Improving interactive latency for the connected car," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2018, pp. 557–572.

[6] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Sep. 2006, pp. 227–238.

[7] S. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu, "Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2611–2615.

[8] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures*, document 3GPP TS 36.213 V15.1.0, Third Generation Partnership Project (3GPP), Mar. 2018.

[9] A. Ksentini, P. A. Frangoudis, A. Pc, and N. Nikaein, "Providing low latency guarantees for slicing-ready 5G systems via two-level MAC scheduling," *IEEE Netw.*, vol. 32, no. 6, pp. 116–123, Nov. 2018.

[10] A. Antonić, M. Marjanović, and I. P. Žarko, "Modeling aggregate input load of interoperable smart city services," in *Proc. ACM Int. Conf. Distrib. Event-Based Syst.*, Jun. 2017, pp. 34–43.

[11] R. Munos and A. Moore, "Influence and variance of a Markov chain: Application to adaptive discretization in optimal control," in *Proc. 38th IEEE Conf. Decis. Control*, vol. 2, Sep. 1999, pp. 1464–1469.

[12] M. Zambianco and G. Verticale, "Interference minimization in 5G physical-layer network slicing," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4554–4564, Jul. 2020.

[13] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 442–450.

[14] H. S. Wang and N. Moayeri, "Finite-state Markov channel—A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[15] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *J. Mach. Learn.*, vol. 42, pp. 177–196, Jan. 2001.

[16] R. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA, USA: MIT Press, 1960.

[17] S. Bubeck, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[18] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proc. 16th Eur. Conf. Mach. Learn.*, 2005, pp. 437–448.

[19] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multi-armed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002.

[20] *Technical Specification Group Radio Access Network; NR; Physical Channels and Modulation*, document 3GPP TS 38.211 V16.1.0, Third Generation Partnership Project (3GPP), Mar. 2020.

[21] L. Diez, A. Garcia-Saavedra, V. Valls, X. Li, X. Costa-Perez, and R. Aguero, "LaSR: A supple multi-connectivity scheduler for multi-RAT OFDMA systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 624–639, Mar. 2020.

[22] S. Agrawal and N. Goyal, "Near-optimal regret bounds for thompson sampling," *J. ACM*, vol. 64, no. 5, pp. 30.1–30.24, Sep. 2017.

[23] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An open-source platform for LTE evolution and experimentation," in *Proc. 10th ACM Int. Workshop Wireless Netw. Testbeds, Exp. Evol. Characterization (WiNTECH)*, Oct. 2016, pp. 25–32.

[24] *Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification*, document 3GPP TS 36.321 V13.2.0, Third Generation Partnership Project (3GPP), Jun. 2016.

[25] A. Rostami, P. Öhlén, M. A. S. Santos, and A. Vidal, "Multi-domain orchestration across RAN and transport for 5G," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, Aug. 2016, pp. 613–614.

[26] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2433–2443, Nov. 2018.

[27] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 721–735, Mar. 2017.

[28] I. da Silva *et al.*, "Impact of network slicing on 5G radio access networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2016, pp. 153–157.

[29] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Pérez, "How should i slice my network: A multi-service empirical evaluation of resource sharing efficiency," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Oct. 2018, pp. 191–206.

[30] C.-Y. Chang, N. Nikaein, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2018, pp. 668–673.

[31] T. Guo and A. Suárez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2865–2877, Mar. 2019.

[32] A. Mahajan and D. Teneketzis, *Multi-Armed Bandit Problems*. Boston, MA, USA: Springer, 2008, pp. 121–151.

[33] V. Kuleshov and D. Precup, "Algorithms for the multi-armed bandit problem," *J. Mach. Learn. Res.*, vol. 1, pp. 1–48, Oct. 2000.

[34] M. L. C. Tekin, "Online algorithms for the multi-armed bandit problem with Markovian rewards," in *Proc. 48th Annu. Allerton Conf.*, vol. 2, Sep. 2010, pp. 1675–1682.

[35] D. Yun, A. Proutiere, S. Ahn, J. Shin, and Y. Yi, "Multi-armed bandit with additional observations," in *Proc. Meas. Anal. Comput. Syst.* New York, NY, USA: ACM, Apr. 2018, pp. 13/1–13/22.

[36] T. Stahlbuhk, B. Shrader, and E. Modiano, "Learning algorithms for scheduling in wireless networks with unknown channel statistics," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (Mobihoc)*, Jun. 2018, pp. 31–40.

[37] D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan, "An efficient bandit algorithm for realtime multivariate optimization," in *Proc. ACM SIGKDD (KDD)*, Sep. 2017, pp. 1813–1821.

[38] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Proc. Conf. Neural Inf. Process. Syst.*, Dec. 2014, pp. 199–207.

Lanfranco Zanzi (Associate Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunication engineering from Politecnico of Milan, Italy, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Technical University of Kaiserlautern. He works as a Research Scientist with the NEC Laboratories Europe. His research interests include network virtualization, machine learning, blockchain, and their applicability to 5G mobile networks in the context of network slicing.

Vincenzo Sciancalepore (Senior Member, IEEE) received the M.Sc. degree in telecommunications engineering and telematics engineering in 2011 and 2012, respectively, and the double Ph.D. degree in 2015. He is currently a Senior 5G Researcher with the NEC Laboratories Europe, Heidelberg, focusing his activity on network virtualization and network slicing challenges. He is currently involved in the IEEE Emerging Technologies Committee leading the initiatives on SDN and NFV. He is the Chair of the Emerging Technologies Initiative (ETI) on Reconfigurable Intelligent Surfaces (RIS). He was also a recipient of the national award for the best Ph.D. thesis in the area of communication technologies (wireless and networking) issued by GTTI in 2015.

Andres Garcia-Saavedra received the M.Sc. and Ph.D. degrees from the University Carlos III of Madrid (UC3M), in 2010 and 2013, respectively. He then joined the Hamilton Institute, Ireland, as a Research Fellow till 2014, when he moved to the Trinity College Dublin (TCD). Since July 2015, he has been a Senior Researcher with the NEC Laboratories Europe. His research interests lie in the applications of fundamental mathematics to real-life communications systems and the design and prototype of wireless systems and protocols.

Hans Dieter Schotten (Member, IEEE) received the Diploma and Ph.D. degrees in electrical engineering from the Aachen University of Technology RWTH, Germany, in 1990 and 1997, respectively. Since August 2007, he has been a Full Professor and the Head of the Institute of Wireless Communication and Navigation, Technische Universität Kaiserslautern. Since 2012, he has also been a Scientific Director with the German Research Center for Artificial Intelligence, heading the Intelligent Networks Department.

Xavier Costa-Pérez (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications from the Polytechnic University of Catalonia (UPC), Barcelona. He is currently the Head of Beyond 5G Networks Research and Development, NEC Laboratories Europe, a Scientific Director with the i2Cat Research and Development Center, and a Research Professor with ICREA. His team contributes to products roadmap evolution as well as to European Commission Research and Development collaborative projects and received several awards for successful technology transfers. In addition, the team contributes to related standardization bodies: 3GPP, ETSI NFV, ETSI MEC and IETF. He has been a 5GPPP Technology Board member, served on the Program Committee of several conferences (including IEEE Greencom, WCNC, and INFOCOM), published at top research venues and holds several patents. He also serves as an Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING AND TRANSACTIONS ON COMMUNICATIONS journals. He was a recipient of a national award for his Ph.D. thesis.