Received August 10, 2021, accepted September 6, 2021, date of publication September 7, 2021,
date of current version September 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111143

Timely Admission Control for Network Slicing in 5G With Machine Learning

MATTEO VINCENZI, ELENA LOPEZ-AGUILERA, AND EDUARD GARCIA-VILLEGAS
Department of Network Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

Corresponding author: Matteo Vincenzi (matteo.vincenzi@upc.edu)

This work was supported in part by the EU Horizon 2020 Research and Innovation Programme (5GAuRA) under Grant 675806, and in part by the Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya under Grant 2017 SGR 376.

ABSTRACT For guaranteeing the strict requirements foreseen for 5G, network slicing has been proposed as a dynamic and scalable mechanism for the allocation of customized resources to service providers. Many solutions have been proposed in the literature for the scenario where multiple service providers share the same pool of resources, while the exclusive allocation to different providers is still an open issue due to the associated complexity. In this work, we define a policy-based admission mechanism for exclusive intra-service slice allocation, at fine and adaptable timescales. In particular, we consider the case where optimal admission strategies are pre-computed offline for network state conditions that are representative of typical traffic loads and resource availability. This offline phase is also used to train a Machine learning algorithm; a neural network (NN) learns the best admission policies from a more computationally expensive mechanism in previously studied network conditions. Thus, the NN is used for providing near-optimal admission decisions at runtime under network conditions for which no optimal policy has been computed. The potential of the 5G marketplace in terms of revenue and quality of service is demonstrated for the particular case of services with strict latency constraints by means of a proof of concept tested over network traces from a real network operator. Different strategies are compared for the computation of the admission strategies and results are provided in terms of efficiency in resource utilization, fairness to the service providers, network owners' revenue and complexity. This study confirms the feasibility of a policy-based approach for exclusive intra-service resource allocation, especially if computationally-efficient mechanisms are adopted in the case of missing information about network states.

INDEX TERMS 5G networks, mobile networks, network slicing, admission control, machine learning, neural networks, clustering, Markov processes, pricing.

I. INTRODUCTION

After one decade since the first studies on next-generation networks, and a few years since early regulations and rollouts, 5G deployments are entering into a more mature phase. Indeed, if research and standardization efforts initially focused on architectures and enabling technologies, 5G ecosystem's drive is becoming progressively service-oriented. On the one hand, manufacturers and network owners are willing to fully exploit the potential of 5G's marketplace, on the other hand, regulation authorities and standardization bodies implement solutions for a healthy coexistence among parties [1]–[3].

Service requirements defined for 5G are very strict: i) sub-millisecond latencies for delay-critical services, ii) a 100-fold capacity increase to serve the needs of new applications and network hyperdensification, and, iii) Quality of Service (QoS) and policy control for reliable communications [1], [4]–[6]. In particular, 5G is considered as the enabling technology for three main service types: i) Enhanced Mobile Broadband (eMBB) with high throughput and mobility demands, ii) ultra-low latency and high reliability (uRLLC) services setting strict requirements in terms of delays and reliability, and, iii) Massive

The associate editor coordinating the review of this manuscript and approving it for publication was Yuan Gao.

Machine-to-Machine Communication (mMTC) requiring low data rates for massive IoT-like deployments.

Recent research and standardization efforts confirm the role of *network slicing* as a keystone for next-generation services, enabling dynamic isolation of physical network resources into QoS-tailored logical networks [7]–[9]. According to this paradigm, infrastructure providers (InPs) lease to 5G service providers (SPs), namely the *slice tenants*, portions of the network resources in a scalable and programmable manner, in the form of customized virtual networks, that is, the *network slices*. In other words, network slices are the traded commodity within the 5G marketplace, where InPs are responsible for continuous technology upgrade, whereas SPs are the middlemen in charge for the bargain over the network resources and for the provision of the finished product to the end users. From a contractual point of view, QoS requirements associated with a specific service are guaranteed by a service level agreement (SLA) between InP and SPs detailing the characteristics of the slice leased (e.g., nominal throughput, maximum delay, resource holding time, shared/exclusive access to slice resources, etc.) [3].

In this context, one of the greatest challenges lies in the definition of mechanisms for the management and orchestration of mobile network slices composed of heterogeneous resources from different infrastructures (e.g., access and core network, transport network, cloud infrastructure), while guaranteeing, among others: i) end-to-end (E2E) QoS, ii) isolation from other tenants, iii) efficient resource utilization, and, iv) timely adaptation to traffic fluctuations in time and space [1], [2], [5]–[8], [10]. The scenario is very similar to that of cloud computing where computational, storage, and communication resources are combined in order to abstract customized virtual machines out of the same infrastructure. However, because of the scarcity and high cost of access network resources, standard over-provisioning mechanisms, typical of cloud computing, cannot be exploited for network slice allocation [11].

Two macro categories for slice allocation approaches exist, based on different InPs' business models and target services: reservation-based and share-based, respectively [11]. The first category foresees the reservation of exclusive and customized resources for different network slices, thus providing tenants with strict and stable QoS guarantees, at the cost of higher complexity and lower efficiency. On the other hand, in share-based allocation schemes multiple tenants coexist within a given slice according to prearranged shares, thus improving efficiency and limiting complexity. However, the sharing of slice resources harms tenants' isolation and provides guarantees only on a statistical basis. If fairness is naturally guaranteed in share-based approaches by fixing prearranged shares among tenants, admission control mechanisms are needed when adopting reservation-based solutions, thus leading to a possible degradation in fairness. Efficient solutions exist in the literature for share-based slice allocation, on the other hand, the high complexity associated with reservation-based mechanisms represents an open issue, as it could harm timeliness, customization and efficiency, thus preventing InPs from meeting SPs' requirements [12].

In this article, we make an effort to demonstrate the feasibility of a reservation-based slicing mechanism for services characterized by strict QoS requirements (e.g., uRRLC), by providing a Proof of Concept (PoC) for the *periodic* admission control solution presented in [3]. More into detail, we perform *intra-service slice allocation* (i.e., slice allocation to SPs providing the same kind of service), by enforcing a policy based on bid selection at fine and dynamic timescales, pursuing maximum revenues to the InPs, while improving efficiency and guaranteeing timeliness and fairness towards SPs. An *offline* implementation is followed[1] and optimal admission strategies (i.e., slicing timescale and bid admission policies) are computed according to the following approaches: i) an exhaustive search (ES) over a limited set of network state conditions, and, ii) by using machine learning (ML) mechanisms for providing near-optimal admission strategies for untested network conditions.

Performance is assessed in terms of fairness to the SPs, resource utilization efficiency and InP's revenue. A comparison is provided on network traces from a real mobile operator with respect to reference solutions applied to urban cells of different sizes and traffic patterns. In addition, as centralized architectures are being standardized for 5G networks, based on software defined networking (SDN) [7] principles, the room for a further complexity reduction is investigated by adopting the following procedure: i) clustering cells according to available network traces, ii) obtaining the adaptable admission strategies only for a candidate cell in each cluster, and, iii) comparing the gap in performance when candidates' admission strategies are enforced to other cells within, or outside of, a given cluster.

In conclusion, the main contributions of this work are a PoC on real network traces for intra-service reservation-based slicing at timescales suitable for 5G services, and the assessment of the performance provided by an offline and dynamic implementation of the methodology in [3]. To the best of our knowledge, this is the first study considering a variable timescale for improved customization in slice provision at a reduced increase in complexity. In particular, we compare the performance for Above Threshold (AT), First-Come-First-Served (FCFS) and Best Bid (BB) admission strategies. Results in terms of complexity and performance extend the conclusions in [3] for the case with variable timescales, confirming the AT scheme as an intermediate solution between FCFS and BB strategies, in terms of fairness guarantees and negotiation power to the SPs, while providing InPs with near-optimal revenues and reduced expenditures. Besides, the advantages in adopting more computationally-efficient solutions are studied. On the

[1] Online admission control algorithms are trained at runtime, while offline schemes require an initial training phase, whose computational burden is typically justified by a better performance [12].

one hand, the benefits of adopting a ML-based approach for the computation of the admission strategies has been demonstrated in case of low congestion levels, or, in absence of network state statistics. Finally, clustering proved the possibility of a complexity reduction for the admission strategy enforcement at a network level, as well as a possible solution, with a negligible or limited decrease in performance, when current network state information is not available.

In the remainder of the paper, we first introduce the related works (Section II) and the system model (Section III). Then, we provide the system analysis by defining the performance metrics and studying the complexity corresponding to the different solutions considered (Section IV). Afterwards, we define the system setup and compare the results in terms of strategies enforced and performance provided by exhaustive search with respect to the case of approaches based on ML and clustering (Section V). Finally, we present the conclusions of this work (Section VI).

II. RELATED WORK

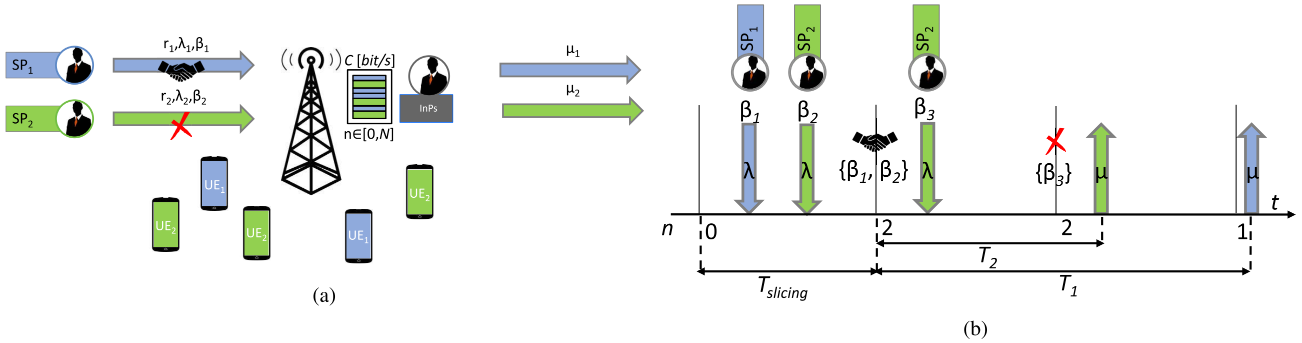
In Section I, we introduced some of the open issues related to 5G slice admission control mechanisms with strict QoS guarantees, in particular, the high complexity of reservation-based schemes, which can prevent the enforcement at fine timescales, thus harming customization and efficiency. In this section, we first introduce the most relevant solutions in the literature with focus on timeliness and, afterwards, we provide a detailed comparison with the approach in this work.

A methodology for defining a policy-based admission strategy is provided in [3]. A continuous-time Markov chain (CTMC) is employed for the computation in an AT scheme of the optimal admission criterion for slice requests, that is, the threshold to adopt for bids associated with incoming requests. In case of admission, bids are registered in the SLAs as the tariff per unit of time charged by the InP to tenants throughout their holding time. Slice admission control is studied both at fixed timescales (i.e., periodic) and upon each request arrival (i.e., on-demand). While on-demand approaches allow a faster response to slice requests, thus minimizing its contribution to delay, periodic admission control limits technological and complexity requirements. When sufficiently small timescales (i.e., negligible with respect to the average service time) are adopted in a way that it is suitable for short-lived services such as emergency or surveillance services [5], [6], both schemes show very similar performance; hence, our interest in the timescales used for the admission process. Finally, both *State Independent (SI)* and *State Dependent (SD)* policies are studied, which foresee fixed or adaptable thresholds for different states of the CTMC. Optimal AT admission policies for specific congestion levels (i.e., the ratio between the arrival and departure rates with respect to available resources) are computed according to exhaustive search. Results show that, when optimal admission policies are computed with sufficient granularity in the search space (i.e., accuracy in the discretization of the bid interval), comparable results are provided by less complex SI solutions with respect to more accurate SD alternatives. Besides, when compared with reference approaches (i.e., on-demand Always Admit, and, periodic FCFS and BB admission strategies), the AT strategy is capable of providing near-optimal revenues to InPs, reducing expenditures and providing a fair slice provision to competing SPs.

An alternative approach is the one described in [12], where an *online* and reduced complexity admission control policy is derived by means of reinforcement learning, which is capable of maximizing InP's revenue while reducing the penalties due to SLAs' violation (i.e., on rejection of slice requests) under different network conditions. One of the key contributions of this solution is its applicability to a scenario where slice requests are issued simultaneously over the same infrastructure for different service types (i.e., eMBB, uRLLC, and mMTC). Three possible algorithms are considered for the computation of the optimal admission policies (i.e., Q-Learning, Deep Q-Learning, and Regret Matching), and performance is assessed by means of computer simulation in terms of: i) maximization of the revenue-to-penalty ratio, and, ii) learning ability of online and offline strategies.

Most of the solutions in the literature are based on online schemes, which are typically characterized by a lower bound for the slice provision promptness set by: i) the time needed by the traffic forecasting algorithm for collecting sufficient data on the network conditions, and, ii) the execution time of ML approaches for efficient enforcement of the admission strategy at runtime (i.e., including the learning phase) [11]. Besides, to the best of our knowledge, slice allocation is typically performed at fixed timescales, which coincide with SPs' holding time as specified in SLAs. This approach alone is generally associated with low efficiency in resource utilization, mostly if coarse timescales are used for slice provision [3], [13]. Solutions exist for improving the efficiency of slice management mechanisms by implementing resource reallocation within slices [13].

From a service modeling perspective, a general characterization is provided in the literature for different service types (e.g., eMBB, mMTC and uRLLC), together with studies on their coexistence and prioritization [12], [14], [16], [17]. The majority of the solutions in the literature adopt a per-SP slicing approach, foreseeing a two level resource allocation: i) per-SP slice allocation used by each tenant for serving multiple customers, ii) a lower level, per-user allocation, adopting more complex mechanisms for resource allocation within a given slice (e.g., scheduling) or across multiple slices [11]. In addition, performance is typically assessed over constant arrival and departure rates [3], [11], [12], with the exception of [13]–[15], which provide results on real network traces. Finally, performance is usually provided by aggregating results from different cells, which is a reasonable strategy in order to provide a network representation. However, this approach hides the suitability



(a)

(b)

FIGURE 1. System model for the slice admission control mechanism (a) when one InP leases resources to competing SPs [3]. Slice requests are associated with: bids, resource requirements, and average arrival/departure rates. Colors identify requests/departures of different tenants and resources used. Rejected requests are marked with a red cross. A possible instance of the periodic admission control policy is represented in (b) for a single service class.

of a specific approach to cells with different features (e.g., coverage, pool of resources, traffic patterns and location).

In our work, we implement an offline and dynamic implementation of a periodic slice allocation policy, which exploits the statistical nature of the CTMC-based scheme in [3] for pre-computing, during a one-time training phase, the optimal admission strategies for known states of InPs' networks (e.g., obtained from historical data). Besides, we include the slicing timescale as one of the admission control parameters adapted by the InP. Indeed, although the highest performance in terms of customization and efficiency is achieved by adopting the smallest timescales for slicing [3], [11], we propose the adoption of slicing mechanisms with adaptive timescale with respect to network congestion. This novel approach enables: i) the limitation of the overall computational requirements without experiencing significant losses in performance, ii) congestion reduction and customization guarantees by adapting the admission strategy on-the-fly with respect to SPs' traffic fluctuations in time and space, and, iii) performance comparable to an on-demand scheme in a cost efficient manner. This strategy, combined with *edge computing*, has the potential to provide the promptest type of slice provision to services with very strict time constraints (e.g., uRLLC), while maintaining a good revenue.

The number and values of network conditions considered for the pre-computation of the admission strategies sets a tradeoff between performance and complexity. In this regard, a neural network is trained with optimal decisions provided by an exhaustive search algorithm for state conditions that are representative of the system. Therefore, an efficient solution is provided for extending the admission strategy to unexplored network conditions (both in time and/or space), and customization is improved in exchange for a limited complexity increase (i.e., the initial learning phase). Besides, clustering-based solutions for limiting the global complexity over the network are studied for centralized architectures. Finally, in this work, we study performance on real network traces from a real mobile operator, and we provide a comparison for urban cells of different sizes and traffic patterns.

III. SYSTEM MODEL

In this section, we present the system model considered for performance assessment of policy-based slice admission control mechanisms, performed on real network traces representing Y different network nodes. In this regard, we refer to Fig. 1a, where multiple users (UEs) subscribing services offered by different SPs coexist within a given geographical area. SPs issue requests for QoS-tailored network slices (i.e., slice requests) to the InP providing coverage over the area, submitting a bid β_s for each request, while the latter takes decisions on which requests to admit. From the InP's perspective, requests from different SPs for a specific service class c are associated with: i) vector $\overline{\boldsymbol{r}}_c$, specifying resource requirements for each resource kind e, ii) average arrival rate λ_c, iii) average service (or departure) rate μ_c, and, iv) maximum waiting time τ_c accepted, from the slice request until its provision. Every time a SP is admitted in the network, it is regarded as a slice tenant with identifier s, with whom the InP stipulates a SLA containing information on the slice customization (i.e., $c = \{\overline{\boldsymbol{r}}_c, \lambda_c, \mu_c, \tau_c\}$) and the agreed tariff β_s in monetary units per second (e.g., [*euros/sec*]). Conversely, in case of rejection, requests are dropped, and no mechanism is implemented for recovery in successive allocation intervals. In Fig. 1a, different colors are used for identifying different SPs and corresponding UEs, SLAs and resources allocated (e.g., assigned portion of the total access link capacity C).

The heterogeneous resource profiles of different slice classes are mapped by the InP into a feasibility region $\mathcal{F}$, whose contours are defined according to the resource pool of the InP. In particular, the allocation state at the i-th slice interval is modeled by position vector $\overline{\boldsymbol{n}}(i)$ in a multi-dimensional space, which is defined in each dimension by the number of slices n_c currently allocated to a specific slice class c. The set of feasible allocation states is formed by the number of slices that can be simultaneously allocated to each class (i.e., $\overline{\boldsymbol{n}}(i) \in \mathcal{F}$), and a resource sharing vector $\overline{\boldsymbol{\sigma}}_c(i)$ is associated to each slice class c, where sharing factor $\sigma_c^e(i)$ indicates the share over total amount of resource e allocated to c at interval i. If we consider a policy region within $\mathcal{F}$ that limits the actual number of slices that can be allocated to each class according to InP's prioritization of different services (similar to [16]), we can split the joint allocation of heterogeneous slice classes into c separate allocation problems. Therefore, the projection of the policy region over a specific dimension provides a variable maximum number of slices N_c that can be allocated to a given slice class c at a given instant (i.e., $n_c(i+1) \leq N_c(i+1)$).

We remark that, according to the system model in [3], the problem is modeled focusing on the aggregate resource demand to the InP, therefore, multiple slices can correspond to the same tenant, or even to the same UE subscribing services from one or multiple SPs. For a given service class, we assume that bids can vary between a minimum and maximum tariff: β_m^c and β_M^c, respectively. Besides, as the focus of our work is on the timeliness of the slice admission control process rather than on strategic bidding, we model SPs as irrational entities following a random bidding model. We represent with $T_c = 1/\mu_c$ the average *holding time* of slice class c, while we employ T_s for referring to the exact time interval during which resources are exclusively retained by a generic tenant with identifier s. In the periodic case, we remark the difference between the holding time T_s of a generic s-th tenant and the timescale $T_{i,c}^{slicing}$ adopted by the InP for periodic slice allocation to service class c. In particular, T_s is the exact holding time for a generic slice tenant s, during which the agreed tariff is applied if the SLA is respected (i.e., the total price paid equals $\beta_s T_s$). On the other hand, $T_{i,c}^{slicing}$ is the length of the time interval during which InP collects slice requests for service class c, which will be admitted or rejected at the beginning of the following allocation interval. We assume that $T_{min}^{slicing}$ is the minimum timescale offered by InP to SPs in order to keep complexity and overhead costs limited. A possible instance of the slice allocation process is proposed in Fig. 1b for the case with a single service class.

From the slice allocation mechanism's perspective, time is a discrete variable represented as a sequence of Ψ slice intervals $\{T_{i,c}^{slicing}\}_{i=1,\ldots,\Psi}$. In order to account for InP's capability to timely adapt the slicing timescale as a part of the admission strategy, we adopt the following representation for the initial time instant of the i-th interval of service class c: $t_{i,c}^0 = t^0 + \sum_{\zeta=1}^{i} T_{\zeta-1,c}^{slicing}$, with t^0 and $T_{0,c}^{slicing}$ representing, respectively, the first time instant observed, and the first interval for slice request collection. For a specific slice class c, we represent the ρ_i^c slice requests received within the i-th interval with $\{s_{i,q}^c\}_{q=1,\ldots,\rho_i^c}$, disposed in order of arrival according to index q. Assuming that the average arrival rate varies in time, thus identifying periods with higher or lower load in terms of traffic, it holds $\mathbf{E}[\rho_i^c] = \lambda_c(i)\ T_i^{slicing}$. On the other hand, we assume that departure rates for a given service class do not vary with time. Similar to the arrival rate, we assume that resource requirements $r_c(i)$ can also vary with time, thus accounting for QoS customization within a specific service class. Therefore, every slice allocated for the i-th interval deduces an amount $r_c(i)$ from the resource pool until departure.

The *admission policies* $\mathcal{P}_i^c$ that InPs can enforce for a specific service class c at the end of allocation interval i, are defined below for the sequence $\{\beta_{s_{i,q}^c}\}_{q=1,\ldots,\rho_i^c}$ of bids received within the i-th slice interval. We represent with $n_c^a(i+1)$ the number of slice requests admitted at the beginning of current slice interval and we remark that policies enforced at interval i depend on the maximum number of slices $N_c(i+1)$ that can be allocated to class c according to the policy region defined by the InP.

1) FIRST-COME-FIRST-SERVED (FCFS) AND BEST BID (BB)

FCFS and BB represent two antithetical admission strategies in terms of fairness towards SPs and revenue to InP because, although they both maximize the number of admissions by allowing resource exhaustion, the former admits requests according to the order of arrival (i.e., independently from the associated bids), while the latter orders requests from the highest to the lowest bid (i.e., prioritizes SPs with highest spending power). In other words, FCFS applies the policy described below to incoming bids $\{\beta_{s_{i,q}^c}\}$ for increasing values of index q. On the other hand, BB first sorts bids values from the greatest to the smallest according to a new listing index $\hat{q}$, then, it applies the policy described below to $\{\beta_{s_{i,\hat{q}}^c}\}$ for increasing values of index $\hat{q}$.

$$\begin{aligned} &\mathcal{P}_i^c(\beta_{s_{i,q}}^c) = \begin{cases} \textit{Admit}, & \text{if } n_c(i+1) \leq N_c(i+1) \\ \textit{Reject}, & \text{otherwise} \end{cases} \\ &\textit{FCFS:}\ \{s_{i,q}^c\}_{q=1,\ldots,\rho_i^c} \\ &\textit{BB:}\ \{s_{i,\hat{q}}^c\}_{\hat{q}|\beta_{s_{i,\hat{q}}^c} \geq \beta_{s_{i,\hat{q}+1}^c}} \end{aligned} \tag{1}$$

2) ABOVE THRESHOLD (AT)

AT strategy represents a tradeoff between FCFS and BB solutions in terms of fairness and revenue to the InP. Indeed, similarly to the FCFS approach, slice requests are admitted in order of arrival, but only if associated bids are above a specific threshold $\dot{\beta}_i^c$, which can be set by the InP to any value within the interval $[\beta_m^c, \beta_M^c]$ based on the congestion level of the network. In other words, on the one hand, it enforces a more conservative strategy in terms of resource utilization and, on the other hand, it can pursue the maximization of InP's revenue by choosing a suitable admission thresholds, or it can favour fairness by adopting thresholds closer to $\dot{\beta}_i^c = \beta_m^c$ (i.e., tending to a FCFS strategy). Consequently, AT applies to incoming bids $\{\beta_{s_{i,q}^c}\}$ the policy described below for increasing values of index q.

$$\begin{aligned} &\mathcal{P}_i^c(\beta_{s_{i,q}}^c) = \begin{cases} \textit{Admit}, & \text{if } \beta_{s_{i,q}^c} \geq \dot{\beta}_i^c \wedge n_c(i+1) \leq N_c(i+1) \\ \textit{Reject}, & \text{otherwise} \end{cases} \\ &\{s_{i,q}^c\}_{q=1,\ldots,\rho_i^c} \end{aligned} \tag{2}$$

TABLE 1. Table of notations.

III. System Model[a]	
InP	
Variable	Definition
Y	# network nodes
i	Slice interval identifier from 1 up to Ψ
$t^0_{i,c}$	Initial instant of i-th slice interval for service class c
$T^{slicing}_{i,c}$	Duration of i-th slice interval for service class c
$T^{slicing}_{min}$	Min. timescale supported by InP for slicing
$n_c(i)$	# slices allocated to service class c at interval i
$\overline{\boldsymbol{n}}(i)$	Vector of slice allocation to each service class at i
$\boldsymbol{\mathcal{F}}$	Feasibility region for heterogeneous slices allocation
$N_c(i)$	Maximum # of slices for class c at interval i
ρ^c_i	# slice requests received for class c at interval i
$s^c_{i,q}$	Identifier of slice request q at interval i for class c
$\mathcal{P}^c_i$	Admission policy applied at the end of interval i for c
$\dot{\beta}^c_i$	Admission threshold applied by AT at the end of i for c
$n^a_c(i)$	# slice requests admitted at the beginning of i for c
SPs	
Variable	Definition
c	Identifier of supported service classes
$r^e_c(i)$	Requirements of class c at interval i for resource e
$\overline{\boldsymbol{r}}_c(i)$	Resource requirements vector for class c at interval i
$\lambda_c(i)$	Average arrival rate for class c at interval i
μ_c	Average departure rate for class c
τ_c	Max. waiting time (request to allocation) for class c
T_s	Exclusive holding time for a generic tenant s
T_c	Average holding time for class c
β^c_m, β^c_M	Min./Max. tariff for class c
$\beta_{s^c_{i,q}}$	Bid associated with slice request $s^c_{i,q}$
$\sigma^e_c(i)$	Sharing factor for class c at interval i over resource e
$\overline{\boldsymbol{\sigma}}_c(i)$	Resource sharing vector for class c at interval i

IV. System Analysis (for a generic class c)	
Variable	Definition
$\nu = (\lambda/\mu, N)$	State condition at a generic instant
$\mathcal{V}$	Set of test state conditions ν for strategy pre-computation
$\mathcal{V}'$	Union of ν from all network nodes' locations
$\xi_i = (\mathcal{P}_i, T^{slicing}_{i+1})$	Admission strategy applied by InP at the end of interval i
$\mathcal{W}$	Search space for admission strategies ξ
ξ^{opt}_ν	Optimal admission strategy for state condition ν
l, h	# values explored for $T^{slicing}$ and $\dot{\beta}$
f_β	Probability density function of β_s
A_i	Admission ratio at the beginning of interval i
$\bar{A}$	Average admission rate
C^{av}_i	Portion of network capacity available at interval i
U_i	Percentage of resource utilization at interval i
$\bar{U}$	Average percentage of resource utilization
R^{tot}_i	Aggregate revenue by all tenants at interval i
R^{tot}	Total aggregate revenue
R_β	Long term revenue rate for specific ν and ξ
$\bar{\beta}_s$	Average admitted bid
$\bar{\tau}_i$	Average waiting time from request until allocation
k, δ	# clusters and features considered for nodes clustering
K	Coefficient for crossfold validation of the NN
n_{HL}, s_{HL}	# hidden layers and neurons/layer considered for the NN
V. Results Evaluation	
Variable	Definition
T_{trace}	Network trace interval
N_{UE}	Average # of active UEs for a specific node
S	Maximum throughput of a given node considering all UEs
M	Total amount of data sent through a network node
γ	Maximum # iterations for clustering
α	Scaling factor for adapting 4G traces to 5G requirements

[a] Sub/superscripts c, i, and q are omitted when a generic service class, slice interval, and/or slice request are considered.

IV. SYSTEM ANALYSIS

In this section, we first introduce the metrics used for performance assessment, then we adapt and study the optimization problem introduced in [3] for offline pre-computation of optimal admission strategies. In particular, the approach proposed in [3] has to be implemented in parallel for each of the c service classes supported by the InP. However, as introduced in Section II, in this context the focus is on the timeliness of an admission control mechanism suitable for slice classes with strict requirements in terms of latency (e.g., short-lived uRLLC). Therefore, rather than studying the resource allocation and slice provision to different service classes, we study and provide performance results for the slice provision to SPs belonging to a specific service class (i.e., sub/superscript c is omitted in the following). Besides, in order to achieve the promptest solution in terms of slice provision, we adopt the periodic scheme proposed in [3] associated with the lowest complexity (i.e., SI strategies) and we perform offline pre-computation of the optimal admission strategies, which are enforced with adaptable timescales $T^{slicing}_i$.

With respect to the resource profile associated to this specific service class, we study a simplified model where only access network resources are considered for slice allocation (i.e., channel capacity C of the access link) because, due to their scarcity, they are the most valuable asset in the slice marketplace (see Section I) and, therefore, they represent the bottleneck in the E2E slice provision [3], [12]. For this specific case, the number $n(i)$ of slices in the system at i-th instant can take values between zero and $N(i) = \lfloor \sigma(i)\ C/r(i) \rfloor$ (sub/superscript e is omitted as only one resource kind is considered).[2]

We consider the highest levels of customization and isolation, that is, per-user slice allocation. This choice is due to two main reasons: i) we consider only access network resources, therefore, it is possible to enforce slice allocations by means of scheduling algorithms, thus removing the complexity deriving from a two-level resource allocation, and, ii) we want to provide a PoC for short-lived uRLLC services expecting timely slice allocations, avoiding the delays related to the aggregation of slice requests coming from multiple users.

Finally, we assume that slice requests arrivals can be modeled as a Poisson stochastic process with average rate $\lambda(i)$, and SPs' departures as a general stochastic process with average rate μ. With respect to the SPs' bidding strategy, we assume that bids β_s can be modeled as a random variable following a general distribution f_β over the sample space $[\beta_m, \beta_M]$.

[2] We remark that $r(i)$ only depends on the resource requirements of the considered slice class, while $\sigma(i)$ is obtained from the policy region defined by the InP and depends on the allocation state $\overline{\boldsymbol{n}}(i)$.

A. PERFORMANCE METRICS

The analytical definitions provided in [3] for the performance metrics of the on-demand case can be easily adapted to the periodic case and expressed as a function of the system model's variables introduced in Section III. In particular, assuming that the *admission strategy* ξ_i enforced at the end of slice interval i can be fully described by tuple $(\mathcal{P}_i, T_{i+1}^{slicing})$, we represent with $A_{i+1} = n^a(i+1)/\rho_i$ the *admission ratio* at the next slice interval, expressed as the ratio between slice requests admitted and total number of arrivals. Until its departure, an admitted slice s (received at slice interval i) implies a decrease of r_i from the available capacity C at slice interval $i+1$, and a contribution to InP's revenue equal to $\beta_s T_s$ (paid proportionally at each of the following slice intervals). Consequently, for a specific service class, if we represent with C_i^{av} the portion of network capacity available out of $\sigma(i)C$ at slice interval i, we define the *percentage of resource utilization* of the service class as $U_i = 1 - C_i^{av}/(\sigma(i)C)$. Finally, if R_i^{tot} represents the aggregate revenue paid by all tenants at a specific slice interval i, we can compute the total *revenue rate* as $R_i^{tot}/T_i^{slicing}$. An average or aggregate version of the same metrics is also provided over the whole observed time interval. In particular, the *average admission rate* $\bar{A}$, the *average percentage of resource utilization* $\bar{U}$, and *the average admitted bid* $\bar{\beta}_s$ are computed averaging over the Ψ slice intervals considered. On the other hand, the *total aggregate revenue* is provided as $R^{tot} = \sum_{i=1}^{\Psi} R_i^{tot}$. Finally, as a measure for the timeliness of the slice admission control method, we employ the average waiting time from the moment a slice request is received, until an admission decision is made,[3] that is, $\bar{\tau}_i = T_i^{slicing}/2$, which has to be lower than the maximum timescale τ accepted by SPs to meet latency requirements for the slice allocation.

B. OPTIMAL STRATEGY AND COMPLEXITY

For the pre-computation of the optimal admission strategies at specific network conditions, we follow the approach presented in [3], which aims at the maximization of InP's revenue rate. The maximization problem extended to the periodic and adaptive case can be defined as follows:

$$\begin{aligned} \xi_\nu^{opt} &= \arg\max_{\xi} R_\beta\Big(\nu, f_\beta, \xi\Big) \\ \textit{FCFS, BB: } \xi &\equiv T^{slicing} \\ \textit{AT: } \xi &\equiv (\dot{\beta}, T^{slicing}) \\ T^{slicing} &\in [T_{min}^{slicing}, \tau] \\ \dot{\beta} &\in [\beta_m, \beta_M] \end{aligned} \tag{3}$$

where R_β, represents the revenue rate that InP would obtain in the long term by enforcing a given admission strategy ξ over a network node with state condition $\nu = (\lambda/\mu, N)$. In particular, we remind from the system model that the triple $(\lambda/\mu, r, f_\beta)$ represents SPs' model, in terms of traffic load, resource requirements and bidding behavior. On the other hand, N is a measure of the maximum resource availability at a specific network location with respect to SPs' requirements at a specific time instant. Finally, ξ represents test strategies in the search space for the considered continuous optimization problem, which is mono-dimensional in the case of FCFS and BB strategies, where only the slicing timescale $T^{slicing}$ can be tuned, and bi-dimensional in the case of AT approach, where we can configure both slicing timescale and admission threshold for incoming bids β_s. Because we implement an offline strategy for the pre-computation of the admission policies, the optimization process is performed only once and its outcome can be used for building a lookup table that will be used on-the-fly for different network nodes and time instants. Justified by the computational power of current technologies, we explore in this study the exhaustive search of the optimal strategies, and we compare its performance with more computationally-efficient and flexible methods based on ML.

In order to limit the complexity of the offline pre-computation of optimal admission strategies, we discretize the search space independently over its dimensions, transforming the problem in (3) into a combinatorial optimization problem. More into detail, we assume that InP can arbitrarily choose for $T^{slicing}$ a finite number l of sample values in $[T_{min}^{slicing}, \tau]$. On the other hand, for AT strategies only, we consider a uniform discretization of $\dot{\beta}$ into a finite number h of intervals over the sample space $[\beta_m, \beta_M]$, that is, $\dot{\beta} = \beta_m + j(\beta_M - \beta_m)/h, \;\; j = 0, \ldots, h-1$. In conclusion, the candidate admission strategies ξ are defined over a space $\mathcal{W}$ of cardinality $|\mathcal{W}| = l$, or $|\mathcal{W}| = l \cdot h$ in FCFS and BB case, or in AT case, respectively. The discretization of the search space could lead to the curse of dimensionality, where a higher number of sample values is translated into increased complexity, although not necessarily associated with a better statistical significance. Therefore, the particular choice of the sample admission strategies (considering both cardinality $|\mathcal{W}|$ and selected values) could lead to very different performance and, in general, the adoption of a decomposition algorithm is recommended for the discretization of the sample space according to its most representative features. However, in this study, we decide to limit complexity by choosing few sample strategies, while relying on the NN for extending the admission strategies to unexplored regions of the sample space. Indeed, the NN is trained by using the input-target pairs (ν, ξ_ν^{opt}) for providing near-optimal strategies ξ in correspondence of generic state conditions ν.

As remarked in [3], the InP is responsible for pre-computing convenient strategies in correspondence of network conditions that are representative of real SPs' behavior and resource availability at different nodes of the network. Therefore, InP has to properly choose the tuples ν over the discrete sample set $\mathcal{V}$ to be used for the offline solution of problem in (3). In order to limit complexity

[3] According to [3], the properties of Poisson processes can be exploited for computing the average value for the arrival instant t_i^a within the i-th slice interval, that is, $\mathbf{E}[t_i^a] = T_i^{slicing}/2$. Then, $\bar{\tau}_i = T_i^{slicing} - \mathbf{E}[t_i^a]$.

while improving the versatility of pre-computed strategies, rather than solving the optimization problem separately for all possible conditions of different nodes at different times, we select sample state conditions that are statistically representative of the whole network (e.g., observing historical data gathered from different locations and time instants). Strategies need to be employed for mitigating the curse of dimensionality, which could lead to the overfitting of the neural network if the input state conditions selected for the initial training do not have statistical significance for all the network nodes in different hours. In this case, we first compute the union $\mathcal{V}'$ of all the tuples $(\lambda/\mu, N)$ obtained from network traces over different nodes' location. Afterwards, we perform an initial coarse and homogeneous sampling over $\mathcal{V}'$ and, finally, we run a fine scale sampling over the most occurring tuples.

1) EXHAUSTIVE SEARCH

The complexity of the offline pre-computation for a specific network condition by means of exhaustive search is linear with respect to the cardinality $|\mathcal{W}|$ of the search space for ξ (i.e., $\mathcal{O}(|\mathcal{W}|)$). The overall time required for the admission strategies' pre-computation strictly depends on the number of samples states considered (i.e., on the cardinality $|\mathcal{V}|$ of $\mathcal{V}$), which also determines the complexity of implementing the lookup table at runtime. In particular, for an arbitrary network condition ν_i at a specific node location and time interval i, we enforce the admission strategy ξ_i^{opt} corresponding to the tuple ν in $\mathcal{V}$ that minimizes the squared euclidean distance $d(\nu_i, \nu)^2$. Assuming that the minimization is performed by implementing the quicksort algorithm over the squared euclidean distances plus the selection of the smallest value, the average complexity is $\mathcal{O}(|\mathcal{V}|)$ independently from the admission strategy considered.

InP needs to implement the runtime process described above in parallel for all the Y network nodes, thus, with a network complexity at runtime equal to $\mathcal{O}(Y \cdot |\mathcal{V}|)$. As a possible solution for the reduction of the complexity over the network, we consider the approach introduced in Section I, that is, performing offline clustering of network nodes according to historical data, and applying optimal admission strategies of few candidates (i.e., nodes corresponding to clusters' centroids) to the rest of the nodes in the network. In particular, we perform clustering according to k-means implementation, that partitions Y nodes into k clusters based on δ-dimensional features extracted from network traces, while considering as objective function the global minimization of the squared euclidean distance to the clusters' centroid.

Although clustering requires an increase in the overall computational complexity, this process is performed only once offline, in exchange for a complexity reduction at runtime by a factor k/Y, which is the dominant component of adopting a policy-based solution on the long-term. However, in scenarios where the network is expected to experience drastic changes, clustering can be repeated according to a given periodicity in order to maintain an updated and accurate representation of clusters and centroids that fits the network.

2) ML-BASED SEARCH

As detailed in the previous subsection, the exhaustive search approach is used to generate a discrete solution set for different network conditions. This solution set is then used to train a neural network, which will be capable of providing effective strategies for new network conditions, not previously explored by the exhaustive search (i.e., $\nu \notin \mathcal{V}$). In particular, we consider the sample network conditions (i.e., $\nu \in \mathcal{V}$) as features, and the corresponding optimal strategies ξ_ν^{opt} as the labels in a supervised learning approach. Furthermore, we perform K-fold cross-validation for optimizing the NN training over the following hyperparameters: i) number of hidden layers n_{HL}, ii) number of neurons per layer s_{HL}, and, iii) training function. Besides, in the case of AT strategies, we compare the option where a single NN is used for computing both admission threshold and timescale, with the alternative approach where two parallel NNs are used for computing separately $\dot{\beta}_i$, and $T_i^{slicing}$. The outputs of the pre-trained NN represent a sub-optimal solution of the problem in (3) and are used for enforcing admission strategies at runtime in correspondence of untested conditions. We remind that the NN can be applied to any node in the network, because it is trained with sample conditions that represent statistically behaviors that could be observed throughout the whole network.

The complexity corresponding to the training phase of a NN depends on all the parameters introduced above, in addition to the stop criterion adopted. Finding a strict definition is out of the scope of this study because, similarly to the case of clustering, the training of the NN is performed offline only once. Besides, justified by the computational power offered by existing technologies, we neglect the corresponding increase in the overall complexity count. On the other hand, the enforcement of NN-based admission strategies at runtime at a specific node location and time interval i requires linear algebraic operations over the input network condition ν_i, whose complexity depends only on the NN's topology, that is, $\mathcal{O}(n_{HL} log(s_{HL}))$ [18]. In this work, we consider NN with reduced topology, therefore, the corresponding computational burden at runtime is expected to be lower when compared to the implementation of a lookup table over the pre-computed $|\mathcal{V}|$ admission strategies as described above (see Section V-A).

We remark that, in the case of BB admission strategy,[4] by definition, additional burden is required at runtime for the ordering of incoming slice requests with respect to bid values, when compared with FCFS and AT approaches. In particular, assuming that a quicksort algorithm is used, the average

[4] BB is the most greedy and unfair strategy from the InP's and SPs' perspectives, respectively.

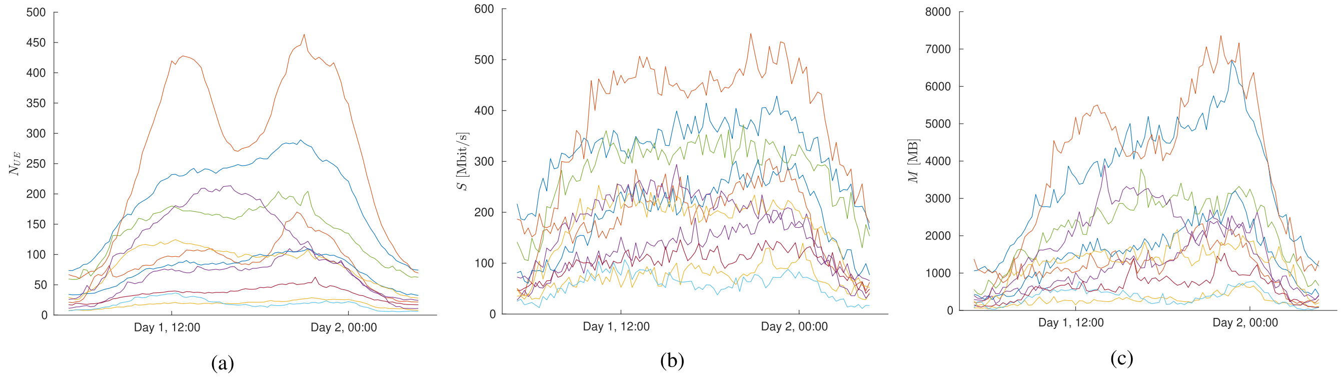


FIGURE 2. Daily averages of network traces from a real mobile operator from 5AM to 4:45AM of the next day: (a) average number of active UEs N_{UE}, (b) maximum throughput S of network nodes in downlink, and (c) aggregated data M sent to UEs in downlink. Different colors are used for different cells.

complexity associated with the BB's bid selection at the end of slice interval i is $\mathcal{O}(\rho_i)$.

V. RESULTS EVALUATION

In this section, we first describe the system setup, then we compare the performance obtained when different admission strategies are adopted. Finally, we study the case where ML strategies are employed for efficient computation of admission strategies, as well as the possible reduction in complexity offered by the offline clustering of network nodes.

A. SYSTEM SETUP

For the performance assessment, we consider the system setup described in the following. SPs slice request arrivals are realized according to a Poisson distribution with average arrival rate λ extracted from network traces, as explained below. On the other hand, for departures we consider an exponential distribution, with average service rate $\mu = 1/60$ set according to the upper limit on the holding time at link layer provided in [19]. SPs' bids follow a uniform distribution within the range $[\beta_m, \beta_M] = [0, 100]$. Finally, both channel capacity C of the access link and resource requirements r are extracted from network traces as explained next.

1) NETWORK TRACES

Network traces are provided by a mobile operator for an operational 4G network over a time interval of one week for eleven network nodes (i.e., $Y = 11$) at a regular periodicity, with trace intervals of size $T_{trace} = 900[s]$. For each network node, information is provided on the average number N_{UE} of active UEs, maximum throughput and aggregate amount of data exchanged with UEs. In the following, and without loss of generality, we only consider downlink resources. We represent with S the maximum throughput in $[Mbit/s]$ considering all UEs, and with M the total amount of data in $[MB]$ sent by the network node. In Figure 2, we provide the daily averages computed over the network traces, which clearly show that different nodes support diverse volumes of traffic, although with similar patterns, as it will be studied in detail in Section V-B.

2) CLUSTERING

We perform k-means clustering on the Y network nodes by considering different values of the number of clusters k and different combinations of δ-dimensional parameters from traces. The maximum number of iterations is set to $\gamma = 100$, and the algorithm is run ten times with random initial centroids in the attempt to filter out the dependence on the starting point. The highest separation in terms of squared euclidean distance between clusters is provided when $k = 2$ is used, and when clustering is performed over the average and the variance of N_{UE} computed over the week (i.e., $\delta = 2$), respectively, $\langle N_{UE} \rangle$ and $Var(N_{UE})$. This result shows a high correlation between N_{UE}, S and M. Resulting clusters are represented with different colors in Figure 3a, with triangle and circle star markers representing, respectively, network nodes and geometrical centroids for each cluster. In the following, we consider as centroids the network nodes in each cluster that minimize the squared euclidean distance to the geometrical centroid (i.e., *Centr1* and *Centr2*). Note that the coordinates for each node are normalized with respect to the network's mean value and standard deviation. We can conclude from Figure 3, that one particular cell in the studied data set shows a very unique behavior and is therefore isolated in its own cluster, perhaps corresponding to the macro cell over the considered geographical area. For cluster 1, we also compute over its nodes the average characterization in terms of $\langle N_{UE} \rangle$ and $Var(N_{UE})$ and we identify the network node that minimizes the squared euclidean distance to this coordinate (i.e., *avNode1*). Besides, we represent in Figure 3b the values of the silhouette coefficients for each network node, representing the similarity of nodes within a cluster, with respect to those in the other cluster. With a mean silhouette value in cluster 1 equal to 0.98 we are sure that a good similarity is achieved among nodes in that cluster, as well as an excellent separation with respect to *Centr2*.

3) ADAPTATION OF NETWORK TRACES TO THE SYSTEM MODEL

In order to adapt 4G network traces to network conditions that take into account the high traffic demands expected for

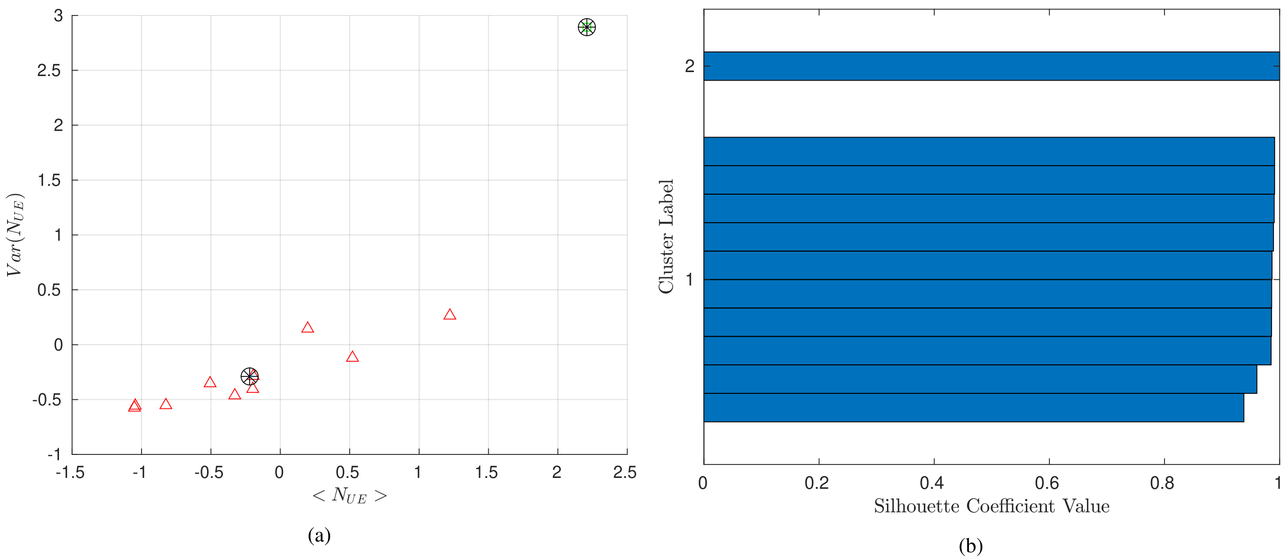


FIGURE 3. ***k*-means clustering on network nodes' traces with $k = 2$: a) clustering with respect to average and variance of N_{UE} over the week, and, b) silhouette plot. In a) coordinates are normalized with respect to mean and standard deviation computed over the network. Besides, nodes of cluster 1 and 2 are represented in red and green triangles, respectively, while a circle star marker is used for the centroids within each cluster.**

5G networks, we introduce a scaling factor $\alpha = 40$ such that $\lambda/\mu = \alpha N_{UE}$. Besides, for the resources available at a specific node's location, we assume that the channel capacity of the access link can be approximated by $C = \max_t S(t)$. Finally, SPs' resource requirements in $[Mbit/s]$ at a specific slice interval are computed as $r = 8 \cdot M/(N_{UE} \cdot T_{trace})$. For simplicity and without lack of generality we assume for access network resources a sharing factor $\sigma(i) = 1$ (i.e., $N(i) = \lfloor C/r(i) \rfloor$ and all the capacity reserved for the considered service class). Indeed, as introduced in Section IV, our focus is in the slice provision to SPs of the same service class, rather than between different service classes. However, the study of the adaptability of this approach to variable levels of resource availability is still guaranteed by the fluctuations of the resource requirements in time, according to the model defined above.

In Figure 4, we compare the levels of congestion[5] $(\lambda/\mu)/N$, for network conditions corresponding to traces of clusters 1 and 2, with respect to the values studied in [3]. In particular, we represent in colored lines the values of $(\lambda/\mu)/N$ when $\alpha = 40$, specifically, for centroids of cluster 1 and 2, as well as for the network node with average characterization within cluster 1. On the other hand, we represent in dashed lines the values of $(\lambda/\mu)/N \in \{0, 0833, 1.667, 16.667\}$ used for Figure 10 in [3], which define the higher limits for $(\lambda/\mu)/N$ when scaling factors $\alpha \in \{0.5, 10, 100\}$ are set, respectively. Therefore, the congestion levels considered in this study range between low (i.e., nowadays overscaled networks) and medium values.

For the discretization of the search space for optimal InP's admission strategies ξ_v^{opt}, we study and compare the performance offered by a fine, intermediate and coarse slicing timescale. In particular, we assume $l = 3$ possible values for the slice intervals $T^{slicing} \in \{0.1/\mu, 1/\mu, 3/\mu\}$, where the extreme values represent, respectively, $T^{slicing}_{min}$ and τ. Besides, for AT strategies, we consider $h = 4$ possible admission thresholds because, according to results in [3], it is sufficient for enabling the full potential in terms of revenue maximization for any network condition. For the selection of the state conditions to be used for the offline solution of the problem in (3), we first perform a coarse selection of 100 samples chosen homogeneously over $\mathcal{V}'$, that is, the union of the state conditions according to network traces of different nodes. Afterwards, we run a fine scale sampling over the most occurring state conditions and achieve a sample set with cardinality $|\mathcal{V}| = 268$.

4) OFFLINE PRE-COMPUTATION OF OPTIMAL ADMISSION STRATEGIES

For the offline pre-computation of the optimal admission strategies by means of exhaustive search, we develop in Matlab a simulator that generates instances of request arrivals, tenants' departure and bidding processes, on which it enforces FCFS, BB and AT admission strategies accordingly, making sure that at least 500 thousand arrivals are detected for each of the tested network conditions. To this aim, we employ an Intel(R) Core(TM) i9-7900X CPU @3.30GHz with 64GB of RAM. On the other hand, when a NN-based solution of the problem in (3) is performed, we reserve 20% of pre-computed strategies for final test while, at each fold of the K-fold cross-validation process, we use 70% and 10% of the pre-computed strategies for training and validation,

[5] The average traffic load λ/μ with respect to the maximum number of available slices N.

respectively. In other words, K-fold cross-validation with $K = 8$ is used for the optimization of the NN training over the following hyperparameters: i) number of hidden layers $n_{HL} \in \{1, 2\}$, and, ii) number of neurons per layer $s_{HL} \in \{5, 10\}$, and when the following training functions are tested: *Levenberg-Marquardt backpropagation*, *Bayesian Regularization*, and *Bayesian Regularization*.

In case of AT admission strategies, the outcome of cross-validation highlights that the best performance in terms of convergence time and output to target error minimization is obtained when two different NNs are used in parallel for computing independently ξ_v^{opt} components (i.e., $\dot{\beta}$ and $T^{slicing}$), both with $n_{HL} = 2$ hidden layers and, respectively, with $s_{HL} = 10$ and 5 neurons per hidden layer. Finally, *Levenberg-Marquardt backpropagation* training function is the one that provides the best performance in terms of convergence to error ratio. Optimal $T^{slicing}$ provided by the NN for AT are also applied for the cases of FCFS and BB.

5) RUNTIME ENFORCEMENT

For the performance assessment, we use a simulator similar to the one described above, with the main difference that optimal strategies are enforced this time over dynamic network conditions obtained from real traces. Given that those traces have a periodicity of $T_{trace} = 900$ seconds, we assume that network conditions remain constant within each trace interval. We remark that, due to time discretization, an intrinsic delay is introduced when enforcing periodic admission strategies with respect to on-demand ones. Indeed, optimal admission strategies ξ_i^{opt} are enforced at allocation interval i, over the vector $\{\beta_{s^c_{i-1,q}}\}$ of bids received during the previous allocation interval. Finally, because we implement slice allocation at fine timescale, we assume that network conditions remain approximately constant within a given slice interval, therefore, no traffic forecasting mechanisms are needed at instant t_i^0 (i.e., the beginning of i-th allocation interval) for guaranteeing the optimality of admission strategies within the slice interval.

B. PERFORMANCE EVALUATION

Below, we first present the admission strategies pre-computed by means of exhaustive search and NN-based search, as well as corresponding performance with respect to different admission strategies. Afterwards, we compare results with the case of admission strategies optimized on a per-node and a per-cluster basis.

1) OPTIMAL STRATEGIES

In Fig. 5, we can observe the fluctuation of the optimal strategies ξ_i^{opt} in time, expressed as: i) the timescale for slice allocation normalized to the average service time (i.e., $T_{opt}^{slicing}/(1/\mu)$), and, ii) the admission threshold for incoming bids when AT strategies are studied (i.e., $\dot{\beta}_{opt}$). Optimal strategies are provided over different network nodes' traces: a) *avNode1*, b) *Centr1*, and, c) *Centr2*. The strategies computed by means of exhaustive search and NN-based approach are provided in solid black line and discontinuous blue line, respectively.

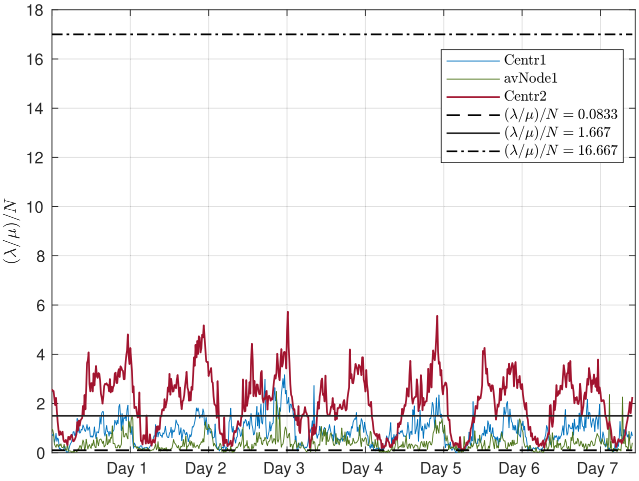


FIGURE 4. Values of $(\lambda/\mu)/N$ over the weekly traces in colored lines for *Centr1* and *Centr2*, as well as for *avNode1*. As a reference, the levels of congestion studied in [3] are also represented in dashed lines.

We can observe in the figure how the chosen admission strategies change in presence of different average levels of congestion, increasing from Fig. 5a to Fig. 5c. In particular, according to Fig. 5c, for high levels of congestion the recommendation to InPs is to adopt very fine timescales (i.e., small values for $T_i^{slicing}$) in such a way that more slice requests can be served in time. Furthermore, in the case of AT strategy, admission thresholds are set to the 25% of the bidding range for most of the time, while it mimics the FCFS scheme (i.e., the minimum admission threshold is adopted; $\dot{\beta}^{opt} = \beta_m$) only when very low congestion levels are perceived; note that the lowest threshold values $\dot{\beta}^{opt}$ observed in Fig. 5c correspond to nightly hours, during which the utilization of the network is low. On the other hand, according to Fig. 5a and 5c, more relaxed strategies are preferred when congestion levels get lower. Indeed, if arrivals are less frequent, less resolution is needed in time for serving all incoming slice requests, consequently coarser timescales can be adopted. Besides, in those cases, lower admission thresholds are preferred on average by the AT strategy.

Finally, we can observe that more flexible admission strategies are adopted by the NN-based approach, indeed, intermediate admission strategies are provided in the continuous domain for state conditions that were not explored for optimal strategy pre-computation. This phenomenon is particularly visible in the case of low congestion levels (see Fig. 5a), where coarser timescales are provided and combined, in the case of AT strategy, with lower values of the admission thresholds. However, we can see that NN's recommended strategies follow quite well those found by the exhaustive search approach. This means that the NN model is well adjusted to the training data provided by the ES study (cf. Section IV-B).

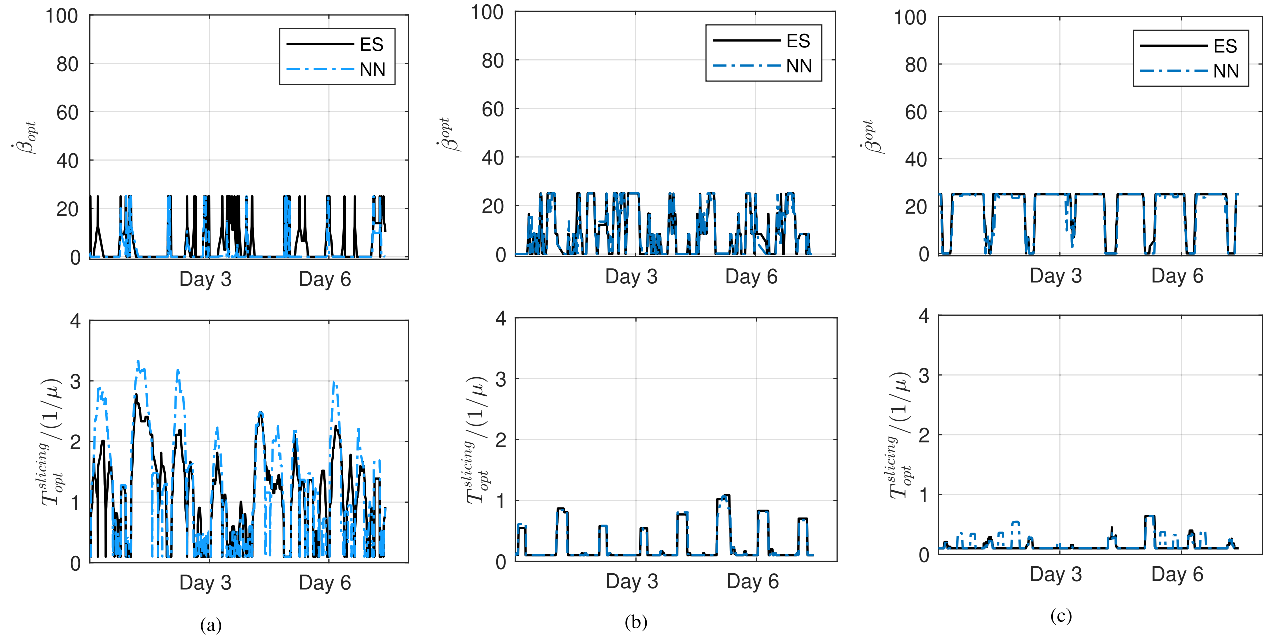


FIGURE 5. Optimal strategies ξ_i^{opt} (i.e., the timescale $T_i^{slicing}$ for slice allocation normalized to the average service time $1/\mu$ and, for AT strategies, admission threshold $\dot{\beta}_i$) computed over different network nodes' traces: (a) *avNode1*, (b) *Centr1*, and, (c) *Centr2*. Results are compared for ES and NN-based approaches in solid black line and discontinuous blue line, respectively. The moving average over one hour is used for a clearer representation.

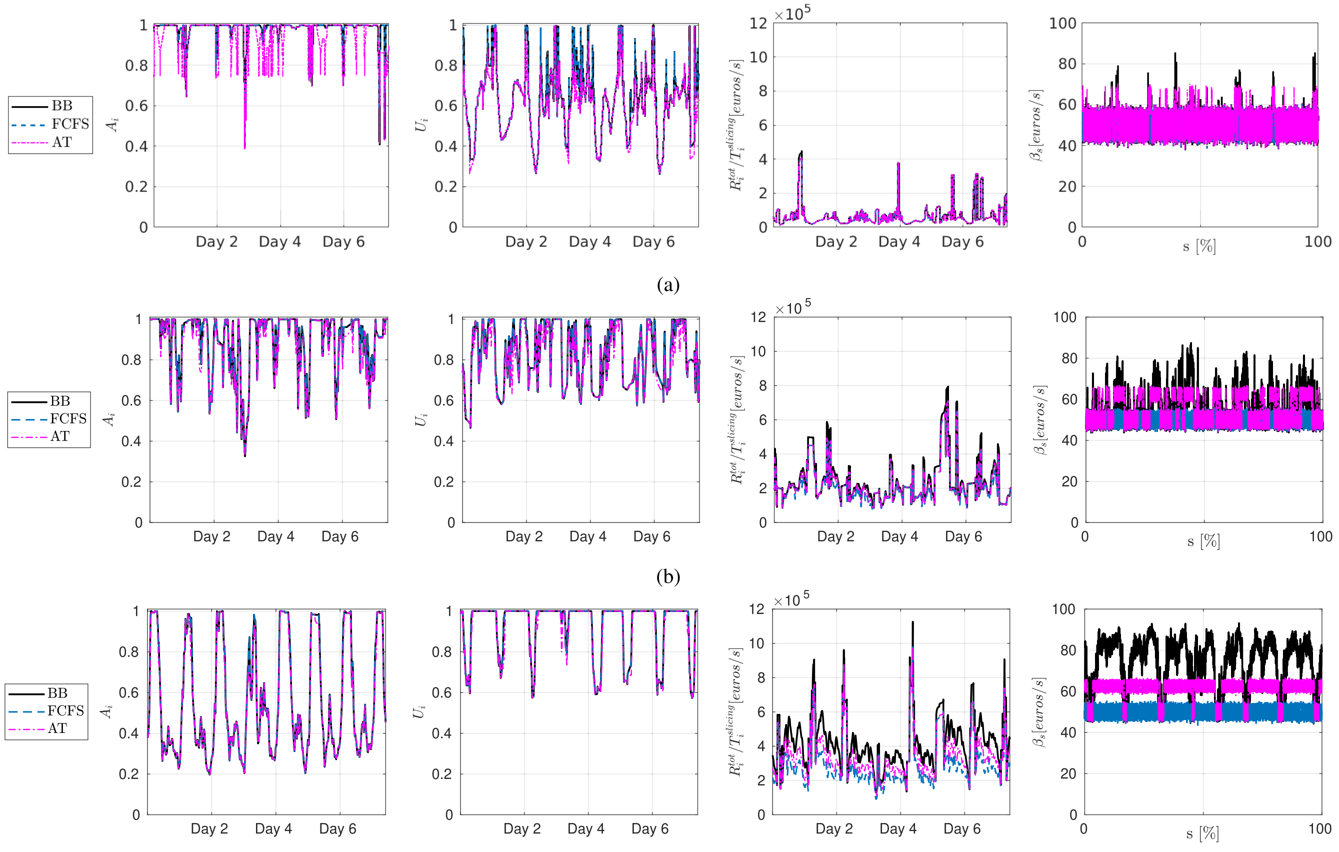
2) PERFORMANCE EVALUATION WITH EXHAUSTIVE SEARCH

As we introduced in Section III, a measure of the timeliness of a slice admission control method is provided by the average waiting time $\bar{\tau}_i$, equal to half of the admission timescale $T_i^{slicing}$. Besides, we remind that the minimum timescale $T_{min}^{slicing}$ allowed by InP is defined by technological and complexity factors, while its maximum value τ depends on SPs' latency constraints. In Fig. 5, we observed how the optimal admission strategies tend to provide relaxed timeliness for decreasing values of congestion level. Therefore, we remark the importance of defining in the SLA both τ and the penalty to the InP when this condition is not met, especially in the case of SPs with very strict requirements in terms of $\bar{\tau}_i$ and in presence of very low congestion levels (e.g., see Fig. 5a).

In Fig. 6, performance is assessed for different admission schemes (i.e., BB, FCFS, and AT) when strategies are computed by means of exhaustive search. In particular, different nodes' traces are considered: a) *avNode1*, b) *Centr1*, and, c) *Centr2*, and, from left to right, we represent the results for the rest of the performance metrics introduced in Section III: i) the admission ratio A_i, ii) the percentage of resource utilization U_i, iii) the revenue rate $R_i^{tot}/T_i^{slicing}$, and, iv) the accepted bids β_s. Finally, in order to have a quantitative measure of performance over the week, we provide in Fig. 7 the average admission rate $\bar{A}$, the average percentage of resource utilization $\bar{U}$, and the total aggregate revenue R^{tot}.

It can be observed in both figures that BB and FCFS always provide the same values for the admission rate, as they both allow slice allocation up to resource-exhaustion. On the other hand, the AT strategy reduces utilization by rejecting bids below a given threshold, which also corresponds to a lower admission ratio. This is particularly evident in the case of low congestion levels (e.g., *avNode1* according to Fig. 4), as the relative ratio of rejections increases with respect to the number of arrivals. Similar considerations hold for the percentage of resource utilization because, thanks to the lower number of admissions, less resources are used on average by the AT strategy when compared with FCFS and BB schemes.

In terms of revenue to the InP, any strategy provides similar revenues in case of low levels of congestion (e.g., *avNode1*). Indeed, because resources are overdimensioned with respect to the economic opportunities, each of the considered schemes tries to admit every incoming request. When congestion increases, the choice of the bids to admit becomes crucial for the revenue maximization, however, only AT and BB strategy can exploit the potential offered by the bigger number of incoming slice requests for achieving higher revenues. Comparing into more detail the revenue offered by different admission schemes, FCFS strategy provides the minimum revenue at zero complexity for its enforcement at runtime (i.e., it admits every new slice request up to resource saturation). On the other hand, BB approaches allow InPs to always select the highest bids at the cost of higher complexity in the long term, as explained in Section IV-A. Finally, AT approaches represent a tradeoff between FCFS and BB schemes in terms of revenue and complexity. Indeed, they always offer intermediate revenues between FCFS and BB schemes. Besides, strategies can



(a)

(b)

(c)

FIGURE 6. Performance assessed by adopting strategies ξ_i^{opt} computed by means of exhaustive search over different network nodes' traces: (a) *avNode1*, (b) *Centr1*, and, (c) *Centr2*. A comparison is provided between different admission strategies (i.e., BB, FCFS and AT) in terms of admission ratio A_i, percentage of resource utilization U_i, revenue rate $R_i^{tot}/T_i^{slicing}$ and accepted bids β_s. The moving average over one hour is used for a clearer representation.

be computed offline only once and enforced at runtime by comparing incoming bids with a threshold.

Together with complexity, admission rate, resource utilization and revenue, another term of comparison for the admission strategies is represented by the admitted bids β_s, which are shown in Fig. 6 with respect to the order of arrival s normalized to the total number of arrivals over the week. As explained for revenue, in case of low congestion levels, all admission schemes admit slice requests independently from the associated bids due to the scarcity of incoming revenue opportunities with respect to resources available. Consequently, according to the figure, the average admitted bid equals the mean value of β_s (i.e., $\bar{\beta}_s = (\beta_M - \beta_m)/2$ for a uniform bid distribution).[6] On the other hand, when the congestion level increases, FCFS does not change its admission strategy, while both AT and BB schemes become more selective and admit slice requests with higher associated bids.

The admission rate together with the average value for admitted bids can be interpreted as a measure of the fairness of InPs towards SPs accounting for: i) InP's greediness in resource usage for revenue maximization, and, ii) fair treatment of SPs' spending power, respectively. In conclusion, FCFS is the admission strategy with lower complexity and highest level of fairness, as it provides highest admission rates and lowest average values for the admitted bids. On the other hand, BB scheme maximizes revenues at the cost of increased complexity and lowest fairness towards SPs' spending power, as it sets the highest average value for the admitted bids. Finally, AT approach represents a tradeoff between the other considered schemes, as it provides slightly lower admission rates while requiring less resources. Besides, it is capable of providing higher revenues than FCFS strategy and, when compared with BB approach, it limits complexity and provides a more fair solution in terms of SPs' spending power, by setting lower average value for the admitted bids.

3) PERFORMANCE EVALUATION WITH ML-BASED STRATEGIES

In Sections I and II, we introduced the possibility of adopting ML-based solutions for providing near-optimal admission strategies for network conditions that have not been directly

[6] We remark that the moving average over one hour is used for a clearer representation in Fig. 6.

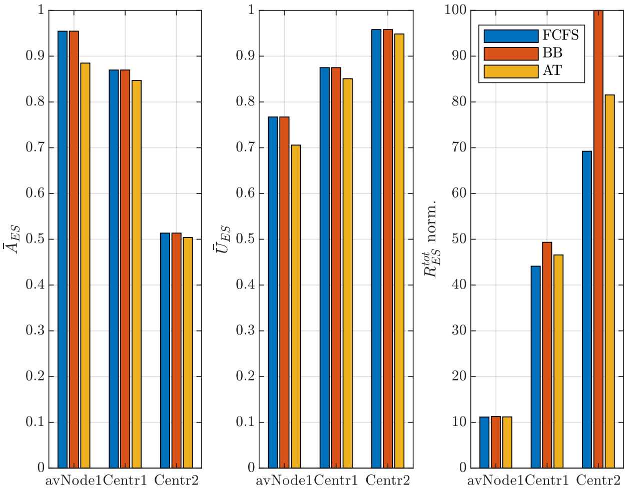


FIGURE 7. Performance over the week when adopting different admission schemes, with strategies computed for different network nodes by means of ES: average admission rate $\bar{A}$, average percentage of resource utilization $\bar{U}$, and total aggregate revenue R^{tot}. For R^{tot}, values are normalized to the maximum over the three strategies and network nodes.

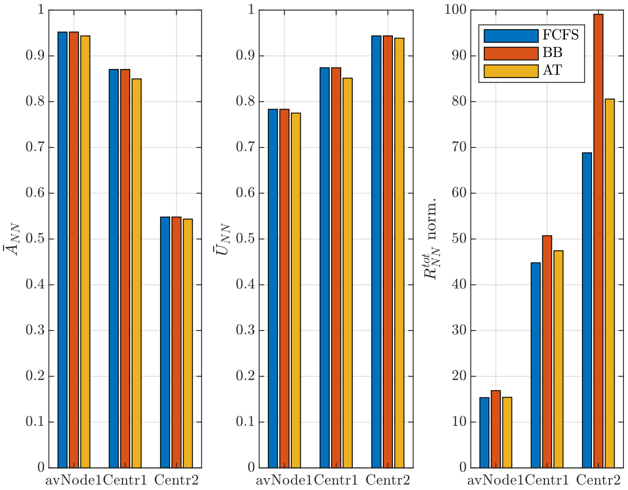


FIGURE 8. Performance over the week for different network nodes and admission schemes, with NN-based strategies. For R^{tot}, values are normalized to the maximum over the three schemes and network nodes when optimal strategies are adopted.

explored by the InP during the pre-computation phase. In particular, in Section IV-B, the advantages in terms of computational efficiency have been detailed for the case of a NN trained on the exhaustive search's output, thus providing custom admission strategies for different network nodes and congestion levels. In Fig. 8, we provide the performance study when strategies are chosen by means of a NN-based approach, which can be compared to that in Fig. 7 for optimal strategies.

In the case of network nodes with low congestion levels (e.g., *avNode1*), it can be observed that FCFS and BB strategies do not have much margin for improving the admission rate due to the very low number of slice requests arriving. This does not hold for the AT scheme as an increase in admission rate can still be achieved by adopting lower admission bids (see Fig. 5a). On the other hand, a great benefit in terms of revenue is offered by the adoption of a NN-based approach independently of the admission strategy. This can be explained by the better customization achieved in terms of admission timescales with respect to the input network conditions. The revenue increase comes at the cost of an increase in resource utilization for all admission schemes, which is more evident in the case of AT strategy because of the adoption of lower admission thresholds. More precisely, BB is the strategy experiencing the higher gain thanks to the increase in the average timescale adopted (see Fig. 5a), because a better opportunity is provided for selecting the highest bids among the arrivals. However, we remind that a more unfair behavior is experienced by SPs with respect to their spending power (see Fig. 6).

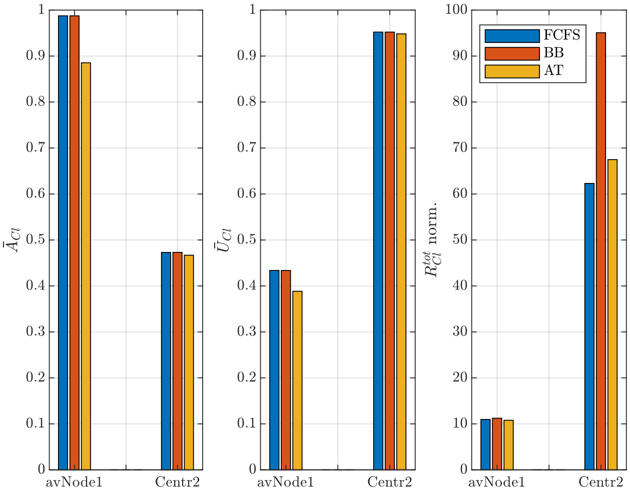


FIGURE 9. Performance over the week for different admission strategies, when *Centr1*'s optimal strategies are applied to *avNode1* and *Centr2*. For R^{tot}, values are normalized to the maximum over the three schemes and network nodes when optimal strategies are adopted.

When considering network nodes with medium congestion level (e.g., *Centr1*), the NN-based approach provides only a slight performance improvement with respect to the exhaustive search approach, which is confirmed by the fact that very similar strategies are adopted by the two approaches (see Fig. 5b). Finally, in the case of very high congestion levels (e.g., *Centr2*), admission rate can be improved by the better customization of admission timescales in time, although, with a slight increase in the average timescale used (see Fig. 5c). Consequently, a worse timeliness is achieved when serving incoming requests, which corresponds to a slightly lower resource utilization and revenues over the week.

In conclusion, the adoption of a NN-based solution for the computation of optimal admission strategies is recommended for network nodes with low or medium congestion levels. In particular, it can provide great gains in terms of revenue, mostly if a BB strategy is adopted and some flexibility exists in terms of fairness towards SPs. Besides, NN-based approaches are suitable for improving AT scheme's admission rate when fairness is preferred over revenue maximization. On the other hand, in case of high congestion levels, there is no incentive in adopting NN-based strategies due to the suboptimal nature of their solutions.

From a different perspective, the limited drop in performance when compared to optimal strategies motivates the adoption of a NN-based approach in case of lack of information on the precise statistics on the network conditions. Indeed, because the NN has been trained on a collection of state conditions from different network nodes with different congestion levels, it represents a suboptimal but more general solution for any network node under any circumstance. Consequently, the trained NN itself could be used by InPs as a computationally-efficient way to provide admission strategies for newly deployed nodes, or for adapting to changes in the congestion levels of already deployed nodes. On the other hand, it could also be used as a tradable asset leased among InPs, or, as a possible object of standardization for guaranteeing comparable performance across different InPs' networks.

4) PERFORMANCE EVALUATION WITH CLUSTERING

In Section I, we discussed the possible reduction in complexity offered by clustering solutions when performing the computation of admission strategies at a network level. In particular, in Section V-A2, we described the methodology for clustering network nodes according to traces, allowing the computation of the admission strategies only for one candidate within each cluster (i.e., the centroid of the cluster). Below, we assess the difference in performance obtained when the optimal strategies of one cluster's centroid are used both for a different node within the cluster and for a node belonging to another cluster. In Fig. 9, we show performance when *Centr1* strategies are enforced at different network locations (i.e., *avNode1* and *Centr2*), which can be compared to that in Fig. 7 for optimal strategies.

When applying *Centr1*'s strategies to the network node with average characterization within cluster 1 (i.e., *avNode1*), we can observe that the admission rate slightly increases for FCFS and BB strategy thanks to the average decrease in the admission timescales adopted (see Fig. 5a and 5b). This does not hold for AT approach as the improved timeliness is counterbalanced by the choice of higher average admission thresholds. On the other hand, resource utilization considerably decreases for all strategies, because of the choice of strategies that are not optimal for the low congestion levels typical of *avNode1*. For the same reason, a negligible reduction in revenue is also registered. Finally, the enforcement of *Centr1*'s strategies in presence of other clusters' conditions (i.e., *Centr2*) provides lower admission rates, resource utilization and revenues, as expected by observing the difference in the admission strategies represented in Fig. 5b and 5c.

In conclusion, adopting clustering strategies represents a valid option for reducing the complexity associated with the enforcement at runtime of optimal strategies over InPs' networks with centralized architectures. Moreover, it could be used in the case of network nodes with well-known statistics on congestion levels and uncertain information about current states. Indeed, in both cases, instead of monitoring and adapting optimal strategies independently for each network node, the InP can alternatively divide network nodes into clusters and apply the strategies that are optimal for a candidate node (e.g., *Centr1*) to the rest of the nodes within the cluster (e.g., *avNode1*), with a negligible difference in terms of performance. Besides, if InP's priority is placed on the maximum reduction in complexity, the same strategies could be also adopted for nodes belonging to other clusters (e.g., *Centr2*) with limited decrease in performance.

VI. CONCLUSION

In this work, we target the potential offered by 5G's marketplace both to network owners and SPs, in terms of revenue and QoS guarantees for services with strict latency constraints (e.g., uRLLC services). In particular, an intra-service reservation-based slicing mechanism has been defined for fine and adaptable timescales, with optimal strategies pre-computed offline for state conditions that are representative of both SPs' behavior, and resource availability in the network. A PoC on real network traces is implemented for studying and comparing complexity and performance of three reference admission strategies (i.e., FCFS, AT, and BB), the latter expressed in terms of efficiency in resource utilization, fairness to the SPs and InP's revenue. Finally, results obtained for optimal admission strategies are compared with those of more computationally efficient solutions.

In this context, our study proves that FCFS and BB strategies provide the minimum and maximum revenue to the InP, respectively, while the opposite holds true in terms of fairness towards SPs and complexity required for enforcement. On the other hand, the AT scheme provides a tradeoff in terms of complexity and performance, while reducing the average resource utilization when variable timescales are used. Furthermore, in case of low congestion levels, the improvement in terms of admission rate and revenue has been demonstrated when using ML-based solutions, at the cost of slightly higher resource utilization and lower fairness with respect to SPs' spending power.

Results show that, if InP's objective is a reduction in complexity, or, the computation of near-optimal strategies in absence of full information about network conditions, approaches based on ML and clustering are good solutions that come at the cost of a negligible or limited decrease in performance. In our future studies, we plan to extend our solution to include the case with different service classes, each with different resource requirements, and to model SPs as rational entities that can react to the fluctuations in price and admission rate by adapting their bidding strategy.

Besides, we foresee the implementation of the proposed methodology on real testbeds for proving the feasibility of adopting adaptive timescales in existing technology.

REFERENCES

[1] *Management and Orchestration; Concepts, Use Cases and Requirements*, document TS 28.530, Release 16, 3GPP, Jan. 2021.

[2] *Telecommunication Management; Study on Management and Orchestration of Network Slicing for Next Generation Network*, document TR 28.801, Release 15, 3GPP, Jan. 2018.

[3] M. Vincenzi, E. Lopez-Aguilera, and E. Garcia-Villegas, "Maximizing infrastructure providers' revenue through network slicing in 5G," *IEEE Access*, vol. 7, pp. 128283–128297, 2019.

[4] *Cisco Annual Internet Report (2018–2023)*, CISCO, San Jose, CA, USA, Mar. 2020.

[5] *Feasibility Study on New Services and Markets Technology Enablers; Stage 1*, document TR 22.891, Release 14, 3GPP, Sep. 2016.

[6] *Service Requirements for the 5G System; Stage 1*, document TS 22.261, Release 18, 3GPP, 2021.

[7] *System Architecture for the 5G System (5GS)*, document TS 23.501, Release 16, 3GPP, Jan. 2021.

[8] *Telecommunication Management; Life Cycle Management (LCM) for Mobile Networks That Include Virtualized Network Functions; Stage 2*, document TS 28.527, Release 16, 3GPP, 2020.

[9] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[10] *Release 15 Description; Summary of Rel-15 Work Items*, document TR 21.915, Release 15, 3GPP, Sep. 2019.

[11] A. Banchs, G. de Veciana, V. Sciancalepore, and X. Costa-Perez, "Resource allocation for network slicing in mobile networks," *IEEE Access*, vol. 8, pp. 214696–214706, 2020.

[12] S. Bakri, B. Brik, and A. Ksentini, "On using reinforcement learning for network slice admission control in 5G: Offline vs. online," *Int. J. Commun. Syst.*, vol. 34, no. 7, pp. 1–14, May 2021.

[13] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "How should I slice my network? A multi-service empirical evaluation of resource sharing efficiency," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM)*, Oct. 2018, pp. 191–206.

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

[15] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive network management in sliced 5G networks with deep learning," in *Proc. IEEE IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 280–288.

[16] 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.

[17] I. Vilà, O. Sallent, A. Umbert, and J. Pérez-Romero, "An analytical model for multi-tenant radio access networks supporting guaranteed bit rate services," *IEEE Access*, vol. 7, pp. 57651–57662, 2019.

[18] K. Bu, Y. Zhang, and Q. Luo, "Depth-width trade-offs for neural networks via topological entropy," 2020, *arXiv:2010.07587*. [Online]. Available: http://arxiv.org/abs/2010.07587

[19] A. Betzler, D. Camps-Mur, E. Garcia-Villegas, I. Demirkol, and J. J. Aleixendri, "SODALITE: SDN wireless backhauling for dense 4G/5G small cell networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1709–1723, Dec. 2019.

MATTEO VINCENZI received the M.Sc. degree in telecommunications engineering from the University of Bologna, in 2014. He is currently pursuing the Ph.D. degree with the Department of Network Engineering, Universitat Politècnica de Catalunya (UPC). He worked as a Researcher at Mavigex S.r.l. He participated as an Early Stage Researcher (ESR) in the European funded project Application-aware User-centric Programmable Architectures for 5G multi-tenant networks (5GAURA), an Innovative Training Network (ITN) of the Marie Skłodowska-Curie Actions (MSCA). His research interests include the area of programmable wireless communication systems, network sharing, and network slicing and pricing.

ELENA LOPEZ-AGUILERA received the M.Sc. degree in telecommunications engineering from the Universitat Politècnica de Catalunya, BarcelonaTech (UPC), in 2001, and the Ph.D. degree, in 2008. She is currently an Associate Professor and a member of the Wireless Networks Group (WNG) with the Networks Engineering Department, UPC. Her research interests include the study of IEEE 802.11 WLANs, the Internet of Things enabling technologies in heterogeneous scenarios, and 5G networks. Her experience also comprises QoS, security, radio resource management, location mechanisms, and wake-up radio systems.

EDUARD GARCIA-VILLEGAS received the M.Sc. and Ph.D. degrees from the Technical University of Catalonia, BarcelonaTech (UPC), in 2003 and 2010, respectively. He is currently an Associate Professor with UPC and a member of the Wireless Networks Group (WNG). He participates in the activities of the IEEE P802.11 WG as a Voting Member. He also participates in the research developed within the i2CAT Foundation. His research interests include IEEE 802.11 WLANs, radio resource management in wireless networks, the IoT enabling technologies, and 5G architecture.