Caballero, P., et al. Network slicing for guaranteed rate services: admission control and resource allocation games, in *2018 IEEE transactions on wireless communications, 17(10), Oct. 2018, pp. 6419-6432*

DOI: https://doi.org/10.1109/TWC.2018.2859918

Network Slicing for Guaranteed Rate Services: Admission Control and Resource Allocation Games

P. Caballero, A. Banchs, *Senior Member, IEEE,* G. de Veciana, *Fellow, IEEE,* X. Costa-Pérez, *Member, IEEE* and A. Azcorra, *Senior Member, IEEE*

***Abstract*—Technologies to enable network slicing are expected to be a key component of next generation mobile networks. Their promise lies in enabling tenants (such as mobile operators and/or services) to reap the cost and performance benefits of sharing resources while retaining the ability to customize their own allocations. When employing dynamic sharing mechanisms, tenants may exhibit strategic behavior, optimizing their choices in response to those of other tenants. This paper analyzes dynamic sharing in network slicing when tenants support inelastic users with *minimum rate requirements*. We propose a NEtwork Slicing (NES) framework combining (i) admission control, (ii) resource allocation and (iii) user dropping. We model the network slicing system with admitted users as a *network slicing game*; this is a new class of game where the inelastic nature of the traffic may lead to dropping users whose requirements cannot be met. We show that, as long as admission control guarantees that slices can satisfy the rate requirements of all their users, this game possesses a Nash Equilibrium. Admission control policies (a conservative and an aggressive one) are considered, along with a resource allocation scheme and a user dropping algorithm, geared at maintaining the system in Nash Equilibria. We analyze our NES framework's performance in equilibrium, showing that it achieves the same or better utility than static resource partitioning, and bound the difference between NES and the socially optimal performance. Simulation results confirm the effectiveness of the proposed approach.**

***Index Terms*—Wireless networks, Network slicing, Multi-tenant networks, Resource allocation, Guaranteed rate services, Inelastic Traffic.**

I. Introduction

It is widely agreed among the relevant industrial community [1] and ongoing standardization efforts [2] that enabling *network slicing* is a key technological requirement for 5G mobile networks. Such technology enables wireless infrastructure to be "sliced" into logical networks, which may be customized to support one or more specific services. This provides a basis for efficient infrastructure sharing among diverse entities, so-called *tenants*, each owning a slice. Tenants could be traditional or virtual mobile network operators acquiring a *network slice* from an infrastructure operator to support their business, as well as new players that simply view connectivity as a service, such as Over-The-Top (OTT) service providers which provision network slices to ensure quality of service to their end-customers.

A major element underlying network slicing is a mechanism for resource allocation amongst slices. One of the approaches considered in 3GPP suggests that base station resources could be statically partitioned based on fixed '*network shares*' [3]. However, given that slices' loads may be non-uniform accross space and varying in time, sharing gains can be achieved by *dynamically* allocating resources to slices based on their current needs (while respecting their overall network shares). At the same time, tenants should retain the ability to operate their slices autonomously and, in particular, to *customize* the allocation of resources to their users. This suggests the need for a flexible framework for resource sharing, wherein (i) tenants indicate their preferences to the infrastructure (e.g., by dynamically subdividing their network share amongst their users), and (ii) base station resources are allocated to slices according to such preferences (e.g., proportionally to the shares assigned to the users).

Under such a resource allocation model, it is to be expected that tenants might exhibit strategic behavior, adjusting their preferences to current demands at the different base stations so as to maximize their performance (subject to their share of the network). This could potentially have adverse effects on the network; e.g., the overall network efficiency might be harmed, or tenants' preferences (and the corresponding requests) might exhibit oscillations. While this problem has been studied in [4] for the case of elastic users, in many cases tenants' traffic will be *inelastic* in nature, wherein a user must either be guaranteed a minimum rate or her utility decreases sharply. When attempting to satisfy such user requirements, tenants' behavior may differ substantially from that in [4], affecting both network efficiency and stability. The focus of this paper is thus on the analysis of resource allocation for network slicing when tenants support inelastic users.

Related work: The resource allocation mechanism analyzed in this paper corresponds to a Fisher market, which is a

Manuscript received January 18, 2018; revised May 15, 2018; accepted July 8, 2018. Date of publication xxxx xx, xxxx; date of current version xxxx xx, xxxx. The work of University of Texas at Austin was supported in part by a gift from Cisco. The work of University Carlos III of Madrid was supported by the H2020 5G-MoNArch project (Grant Agreement No. 761445) and the 5GCity project of the Spanish Ministry of Economy and Competitiveness (TEC2016-76795-C6-3-R). The work of NEC Europe Ltd. was supported by the H2020 5G-Transformer project (Grant agreement no. 761536). The associate editor coordinating the review of this paper and approving it for publication was X. Wang. *(Corresponding author: Pablo Caballero.)*

P. Caballero and G. de Veciana are with The University of Texas at Austin, Austin, TX 78712 USA (e-mail: pablo.caballero@utexas.edu; gustavo@utexas.edu).

A. Banchs and A. Azcorra are with the University Carlos III of Madrid, 28911 Leganés, Spain, and also with the IMDEA Networks Institute, 28918 Leganés, Spain (e-mail: banchs@it.uc3m.es; azcorra@it.uc3m.es).

X. Costa-Pérez is with NEC Europe Ltd., 69115 Heidelberg, Germany (e-mail: xavier.costa@neclab.eu).

standard framework in economics. In such markets, buyers (in our case slices) have fixed budgets (in our case corresponding to pre-agreed network shares) and bid for resources within their budget (according to their preferences), which are then allocated to buyers proportionally to their bids [5]. Within the Fisher Market framework, our model falls in the category of buyers that anticipate the impact of their bids [6]. The analysis of Fisher markets under such price-anticipating buyers has been limited, so far, to the case of buyers with *linear* [6] or *concave* [4], [7] utility functions.

A related resource allocation model often considered in the networking field is the so-called 'Kelly's mechanism', which allocates resources to players proportionally to their bids [8]. This model has also been analyzed for price-anticipating players [9]. However, in Kelly's mechanism players respond to their payoff (given by the utility minus cost) whereas in our model tenants' behavior is only driven by their utilities (since they have a fixed budget, i.e., the network share). Moreover, Kelly's model has mainly been studied for concave utility functions.

The topic of network slicing is currently attracting substantial attention from the research community. One of the main issues investigated is the resource allocation across different slices, which is the focus of this paper. A number of works have been devoted to the resource allocation among different operators or tenants sharing the same wireless infrastructure (see e.g. [10]–[12]), and in [13], the authors focus on resource allocation of processing resources in network slicing in the context of C-RAN; see [14] for a survey on resource slicing in virtual wireless networks. In contrast to our paper, all these works have focused on elastic traffic.

In the context of network slicing, there are some works which have considered inelastic traffic. The algorithm proposed in [15] attempts to satisfy the demands of all slices but does not account for the resources each slice is entitled to. Similarly, [16]–[18] propose algorithms to meet requests from all tenants, but do not account for elastic demands and do not consider budget constraints. In [19], the authors propose an algorithm to trade resources among tenants, but their approach involves complex negotiations and relies on heuristic considerations rather than a well-established analytical framework. In contrast to all these works, our approach supports both elastic and inelastic services and is based on fixed budgets, corresponding to the *network shares*; this is in line with one of the scenarios considered in 3GPP [3] and does not involve pricing individual requests, which may represent an advantage in practical deployments.

In this work, we build on the Fisher Market mechanism for resource allocation across slices and analyze the game resulting from the interaction of several non-cooperative slices aiming to maximize their own network utility given a fixed budget. This problem has been addressed in the context of concave utility functions: [7] ensures the existence of Nash Equilibria (NE) for this type of utility functions, [20] proves the existence of a NE for price-taking players, [4] shows the convergence of Best Response Dynamics for certain classes of concave functions and [6] shows they may not converge for linear utilities. Much less attention has been paid to non-concave utility functions; among the few works on this topic it is worth mentioning [21], which uses potential games to prove convergence of Best Response Dynamics to a region around the NE for finite strategy games [22].

In the specific context of Fisher market-like frameworks, to the best of our knowledge our work is the first attempt to analyze resource allocation for inelastic traffic. In particular, this work addresses the following gap in the literature of resource allocation models: the analysis of *budget-constrained resource allocation* under *price-anticipating users* with *inelastic utilities*. The nature of inelastic utility functions leads to a new class of non-cooperative games, where a slice prefers to drop users whose rate requirements cannot be met, rather than allocating them insufficient resources. The nature of such games differs substantially from the ones previously analyzed in the literature for elastic traffic.

On the 5G standardization front, network slicing is currently being specified by 3GPP [2]. In particular, 3GPP's SA5 is working on the definition of a management and orchestration framework to support network slicing [23], [24]. While these efforts do not specifically address dynamic resource allocation, which is our focus here, the algorithms we propose are in line with this framework. One of the key features of our approach is the ability of tenants to customize their allocations; there is wide consensus in the standardization community that this is needed to efficiently satisfy their very diverse requirements (see, e.g., [25] for examples of possible vertical tenants).

Key contributions: The rest of the paper is organized as follows. In Section II we present our system model, and propose the Network Slicing (NES) framework to address resource allocation in such system. NES consists of three modules: admission control, weight allocation and user dropping. Section III focuses on the admission control module: it finds the requirements to ensure stability and proposes two policies, a conservative and an aggressive one, to perform admission control. Section IV presents the other two modules: a resource allocation mechanism and a strategy to drop users when rate guarantees are infeasible, and analyzes the convergence of the resulting dynamics. We then study in Section V the performance of NES versus two benchmark allocations: static resource partitioning and the social optimal. Throughout the paper, we present analytical results that support the design of NES, including (i) the existence of a Nash Equilibrium and the convergence of Best Response Dynamics, (ii) the effectiveness of admission control and protection from other slices, (iii) the user selection and weight allocation choices, and (iv) the gains over static slices and loss over social optimal. We further evaluate the performance of NES via simulation in Section VI, confirming that it provides substantial gains in terms of utility, throughput performance and reduced blocking probability while incurring an acceptable complexity.

II. Network Slicing model

We consider a wireless network consisting of a set of resources $\mathcal{B}$ (the base stations or sectors) shared by a set of network slices $\mathcal{O}$ (each operated by a different tenant). At a given point in time, the network supports a set of active users

$\mathcal{U}$ (the customers or devices), which can be subdivided into subsets $\mathcal{U}_b^o$, $\mathcal{U}_b$ and $\mathcal{U}^o$, corresponding to the users of slice o at base station b, the users at base station b, and the users of slice o, respectively. We consider that the association of users with base stations is fixed (e.g., by a pre-specified user association policy) and let $b(u)$ denote the base station that user u is (currently) associated with.

A. Resource allocation model

Following a similar approach as [4], [10], in our model each slice o is allocated a network share s_o (corresponding to its budget) such that $\sum_{o\in\mathcal{O}} s_o = 1$. The slice is at liberty to distribute its share amongst its users, assigning them non-negative weights (corresponding to the bids):

$$w_u \text{ for } u \in \mathcal{U}^o, \text{ such that } \sum_{u\in\mathcal{U}^o} w_u \leq s_o.$$

We let $\mathbf{w}^o = (w_u \colon u \in \mathcal{U}^o)$ be the weights of slice o, $\mathbf{w} = (w_u \colon u \in \mathcal{U})$ those of all slices and $\mathbf{w}^{-o} = (w_u \colon u \in \mathcal{U} \setminus \mathcal{U}^o)$ the weights of all users excluding those of slice o. We further let $l_b(\mathbf{w}) = \sum_{u\in\mathcal{U}_b} w_u$ denote the load at base station b, $d_b^o(\mathbf{w}^o) = \sum_{u\in\mathcal{U}_b^o} w_u$ the aggregate weight of slice o at b, and $a_b^o(\mathbf{w}^{-o}) = \sum_{u\in\mathcal{U}_b\setminus\mathcal{U}_b^o} w_u$ the aggregate weight of all other slices (excluding o) at b. We shall allocate each user a fraction of the base station's resources in proportion to her weight w_u.

We let c_u denote the achievable rate for user u, defined as the product of (i) the *average* rate per resource unit achieved by the user, and (ii) the total amount of resources available at the base station. Note that this depends on the modulation and coding scheme selected for the current radio conditions, which accounts for noise as well as the interference from the neighboring base stations. Following similar analyses in the literature [10], [26], [27], we shall assume that c_u is fixed for each user at a given time.

We further let r_u denote the rate allocated to user u. Under our model, r_u is given by c_u times the fraction of the base station's resources allocated to the user. Given that users are allocated a fraction of resources proportional to their weights, we have that r_u is a function of the weights $\mathbf{w}$ given by:

$$r_u(\mathbf{w}) = \frac{w_u}{\sum_{v\in\mathcal{U}_{b(u)}} w_v} c_u = \frac{w_u}{l_{b(u)}(\mathbf{w})} c_u. \tag{1}$$

When implementing the proposed resource allocation mechanism, a slice may assign a non-zero weight to some users while others may be dropped. To decide the setting of the users' weights, we assume that each slice o is aware of the aggregate weight of the other tenants at each base station, i.e., $a_b^o(\mathbf{w}^{-o})$. It is worth noting that for the mechanism under study we have that (i) a slice only sees the aggregate weight of the other slices, and hence can learn very limited information about the other slices; in particular, the weights of each tenant are not disclosed, and (ii) the mechanism needs to store very limited data; indeed, it is sufficient to keep the total load of each base station, as a tenant can obtain $a_b^o(\mathbf{w}^{-o})$ by simply subtracting its weight from the base station's load. Such information is already considered within the network slicing management system defined by 3GPP [24], and hence should be readily available.

In order to avoid the indeterminate form resulting from having all the weights at a base station equal to 0 in (1), we will require weights to exceed a fixed lower bound (i.e., $w_u \geq \delta,\ \forall u$). This bound can be arbitrarily small; indeed, in practice it should be set as small as possible, to allow slices the highest possible flexibility while avoiding zero weights. Accordingly, in the rest of the paper we assume that δ is so small that its effect can be neglected, except for Theorem 2, where this assumption is required to prove the existence of a Nash Equilibrium.

In the case where a slice o is the only one with users at a given base station b, such a slice would simply set w_u to the minimum possible value for these users, allowing them to receive all the resources of this base station while minimizing the consumed share. To avoid dealing with this special case, hereafter we shall assume that all base stations have users from at least two slices. Note that this assumption is made to simplify the expressions and discussion, and does not limit the generality of our analysis and algorithm, which indeed supports base stations with all users from the same slice.

B. Slice utility

Network slices may support services and customers with different needs, or may wish to differentiate the service they provide from competing slices. To that end, we assume that each slice has a *private* utility function, U^o, that reflects the slice's performance according to the preferences and needs of its users. The slice utility consists of the sum of the individual utilities of its users, U_u, i.e.,

$$U^o(\mathbf{w}) = \sum_{u\in\mathcal{U}^o} U_u(r_u(\mathbf{w})).$$

For inelastic traffic, we assume each user u requires a guaranteed rate γ_u, hereafter referred to as the user's minimum *rate requirement*. Following standard practice, we shall model inelastic traffic utility functions as[1]

$$U_u(r_u(\mathbf{w})) = \phi_u f_u(r_u(\mathbf{w})), \text{ for } r_u(\mathbf{w}) \geq \gamma_u,$$

where $f_u(\cdot)$ is a concave[2] utility function associated with the user, and ϕ_u is the relative priority of user u (where $\phi_u \geq 0$ and $\sum_{u\in\mathcal{U}^o} \phi_u = 1$). The relative priorities reflect the importance that users are given by the tenant of their slice; they drive, jointly with the load at the respective base stations, the weights assigned to the users, which in turn determine the rate allocation.

Note that the above utility function is only defined for rates above the minimal requirement, as performance degrades drastically if this guarantee is not met. Note also that the above definition includes elastic traffic, which corresponds to

[1]Inelastic traffic utility functions are typically modeled as a discontinuous function [28] or a sigmoidal one [29]. In this paper we adopt the former model, which aims at providing users with a guaranteed rate, and thus is aligned with the Guaranteed Bit Rate (GBR) class of 3GPP [30].

[2]Note that, even when $f_u(\cdot)$ is concave, we are dealing with non-concave utilities, due to the minimum rate requirement.

the special case $\gamma_u = 0$; thus, the results of this paper apply to mixes of elastic and inelastic traffic.

While most of our results hold for arbitrary $f_u(\cdot)$ functions, in some cases we will focus on the following widely accepted family of utility functions (see α-fairness, [31]):

$$f_u(r_u) = \begin{cases} \frac{(r_u)^{1-\alpha_o}}{(1-\alpha_o)}, & \alpha_o \neq 1 \\ \log(r_u), & \alpha_o = 1, \end{cases} \quad (2)$$

where the α_o parameter sets the level of concavity of the user utility functions, which in turn determines the underlying resource allocation criterion of the slice. Particularly relevant cases are $\alpha_o = 0$ (maximum sum), $\alpha_o = 1$ (proportional fairness), $\alpha_o = 2$ (minimum potential delay fairness) and $\alpha_o \to \infty$ (max-min fairness).

In our model for slice behavior, a tenant proceeds as follows to optimize its performance. First, it maximizes the number of users that see their rate requirement met, selecting as many users as can be possibly served. Second, it maximizes the utility $U^o(\mathbf{w})$ obtained from the users that have been selected.

Note that the above framework is sufficiently flexible to accommodate different network slicing models, including those under study in 3GPP [24]. For instance, in the case where tenants are Mobile Virtual Network Operators (MVNOs), the users of a tenant may have different service demands (e.g., elastic and inelastic users). Alternatively, we can also support a model where different slices are deployed for specific services; in this case, we may have some slices with only elastic users and others with only inelastic users.

C. Baseline allocations

Below we introduce two approaches to resource allocation that we will use as benchmarks to assess the performance of the proposed framework. For now, we shall assume the users' rate requirements can be met, and thus focus on the weight allocation that maximizes the slice's utility.

a) Socially Optimal Allocation (SO): If slices were to share their utility functions with a central authority, one could in principle consider a (share-constrained) allocation of weights (and resources) that optimizes the overall performance of the network, expressed in terms of the *network utility* $U(\mathbf{w})$ defined as the sum of the slices' utilities (see [4], [10]):

$$U(\mathbf{w}) := \sum_{o \in \mathcal{O}} U^o(\mathbf{w}).$$

The above is referred to as the socially optimal allocation, which is given by the following maximization:

$$\begin{aligned} \max_{\mathbf{w} \geq 0} \quad & U(\mathbf{w}) \\ \text{s.t.} \quad & \sum_{u \in \mathcal{U}^o} w_u = s_o, , \quad \forall o \in \mathcal{O}, w_u \geq \delta, \\ & r_u(\mathbf{w}) \geq \gamma_u, \quad \forall u \in \mathcal{U}. \end{aligned}$$

We shall denote the resulting optimal weights and resource allocation in the socially optimal setting by $\mathbf{w}^*$ and $\mathbf{r}^* = (r_u^*(\mathbf{w}^*) : u \in \mathcal{U})$, respectively.

b) Static Slicing Allocation (SS): By static slicing (also known as static splitting [32]) we refer to a complete partitioning of resources based on the network shares s_o, $o \in \mathcal{O}$. In this setting, each slice o receives a fixed fraction s_o of each resource, which is shared among its users proportionally to their weights,

$$r_u^{ss}(\mathbf{w}^o) = \frac{w_u}{\sum_{v \in \mathcal{U}^o_{b(u)}} w_v} s_o c_u, \quad \forall u \in \mathcal{U}^o, \quad \forall o \in \mathcal{O}, \quad (3)$$

where we note that, in this case, the rate of a user depends only on the weights of the other users in her slice, i.e., $\mathbf{w}^o$. A slice can then unilaterally optimize its weight allocation as follows:

$$\begin{aligned} \max_{\mathbf{w}^o \geq 0} \quad & U^o(\mathbf{w}^o) \\ \text{s.t.} \quad & \sum_{u \in \mathcal{U}^o} w_u = s_o, \quad r_u^{ss}(\mathbf{w}) \geq \gamma_u, \quad \forall u \in \mathcal{U}^o. \end{aligned}$$

where we have abused notation to indicate that in this case the slice's utility, given by $U^o(\mathbf{w}^o) = \sum_{u \in \mathcal{U}^o} U_u(r_u^{ss}(\mathbf{w}^o))$, depends only on $\mathbf{w}^o$. We shall denote the resulting optimal weights resulting from static slicing by $\mathbf{w}^{o,ss}$.

D. Network slicing framework

In this paper, we introduce our NEtwork Slicing (NES) framework to address the resource allocation problem in the context of the above system. NES manages both users and resources in network slices, as mobile users come and go. The proposed framework comprises the following modules:

1) *Admission control*: the purpose of this module is to ensure that admitted users will see their rate requirements met during their lifetime with a sufficiently high probability, even after there are changes in the network.
2) *Weight allocation*: this module determines how to allocate weights to the users, with the goal of maximizing the slice's utility.
3) *User dropping*: while admission control aims at ensuring that all rate requirements are always met, when users re-associate or see a change in their radio conditions, or when other slices admit more users, it could happen that a slice can no longer keep all its users while meeting their requirements; in that case, this module decides which users to drop.

The design of the admission control module is presented in Section III, while that of the weight allocation and user dropping modules is presented in Section IV.

In order to analyze the stability of the NES framework, we assume that slices are *competitive* (strategic and selfish), i.e., each attempts to unilaterally optimize its own utility, and model the behavior of the *weight allocation* and *user dropping* modules as a non-cooperative game. Note that this game only considers admitted users, i.e., admission control is not part of the game. It may be played at a point in time when admitted users may have re-associated or seen a change in their radio conditions, or new users may have been admitted; as a result, when playing the game we may not be able to meet all rate requirements. Thus, the game involves slices deciding

(i) which set of users to serve when the rate requirements of all users cannot be met, and (ii) how to allocate weights amongst the slice's users, in response to other slices' decisions. Hereafter we refer to this game as the *network slicing game*; its formal definition is stated as follows:

Definition 1. *Consider a set of slices $o \in \mathcal{O}$, each with a set of admitted users $u \in \mathcal{U}^o$. In the network slicing game, each slice selects which subset of users to serve within the set $\mathcal{U}^o$ and their associated weight allocation $\mathbf{w}^o$ such that (i) as many users as possible are served (meeting their rate requirements), and (ii) the slice's utility U^o is maximized for the selected subset of users.*

III. Admission control for sliced networks

In order to meet user rate requirements, NES needs to apply admission control on new users, rejecting them when the slice cannot guarantee with a very high probability that it will be able to satisfy the rate requirements of all its users during their lifetime. Note that this only applies to new users; in case the user rate requirements can no longer be satisfied as a result of users moving, or other tenants changing their allocations, this is handled by the *user dropping* module described in Section IV-A.

In the following, we analyze the implications of applying admission control on the system stability, and propose two different admission control algorithms, *Worst-case admission control* (WAC) and *Load-driven admission control* (LAC). These two algorithms correspond to different trade-offs between slice isolation and efficiency: while WAC provides perfect isolation, guaranteeing that a slice will never need to drop users because of changes in the other slices' loads, LAC achieves a higher efficiency at the cost of providing more relaxed guarantees on isolation (yet ensuring that the probability of dropping a user remains sufficiently low).

A. Nash Equilibrium existence

A critical question is whether the *network slicing game* defined in Section II-D possesses a Nash Equilibrium (NE), i.e., there exists a choice of users and associated weight allocation $\mathbf{w}$ such that no slice can unilaterally modify its choice to improve its utility. In the following, we analyze the requirements on admission control policies in order to ensure that a NE exists *after* admission control is applied. Note that, if the game does not have a NE, strategic slice behavior may lead to system instability affecting the practicality of the proposed approach.

The following theorem shows that if admission control cannot ensure that slices can satisfy the rate requirements of all their users, the network slicing game may not have a NE. The proof of the theorem exhibits a case where instability arises when there is no weight allocation such that the rate requirements of all the users of a given slice are met given feasible allocations for the other slices. Note that in a dynamic setting such a situation could arise, when a slice initially admits users for which the requirements are feasible, and subsequently other slices admit additional users to their slice, making some of the users in the first slice infeasible (see the Appendix for the proof of all the theorems).

Theorem 1. *When slices cannot satisfy all of their users' rate requirements, the existence of a NE cannot be guaranteed for the network slicing game.*

The problem identified by the above theorem can be overcome by applying an admission control scheme that avoids such situations. According to the following theorem, a NE exists as long as admission control is able to guarantee that a slice can satisfy the rate requirements of all its users under any feasible weight allocation of the other slices (including *future* allocations when possibly new users may have been admitted). Note that in this case the resulting game focuses on maximizing slice utilities while meeting the rate requirements of all users. This result implies that, as long as proper admission control is implemented and ensures that rate requirements can always be satisfied, the stability of the system can be guaranteed.

Theorem 2. *Suppose admission control ensures that, for any feasible weight allocation of the other slices, each slice o has a weight allocation $\mathbf{w}^o$ such that its users' rate requirements are met. Then, the network slicing game has a (not necessarily unique) NE.*

Note that the above theorem guarantees the existence of a NE when all slices are elastic; indeed, elastic slices have a rate requirement equal to 0, and therefore their rate requirements can always be satisfied. This leads to the following result.

Corollary 1. *When all slices are elastic, the network slicing game has a NE.*

In the following, we propose two alternative admission control policies (one more aggressive and one more conservative) that aim at ensuring that the conditions given by Theorem 2 are met. Note that it is ultimately up to the tenant to choose and customize its admission control strategy, and hence each tenant may independently apply its *own* admission control policy.

B. Worst-case admission control (WAC)

The WAC policy is devised to ensure that the rate requirements of all users are always met, independently of the behavior of the other tenants. To that end, under the WAC policy a slice admits users as follows: it conservatively assumes it has access to only a fraction s_o of resources at each base station, and admits users only if their requirements can be satisfied with these resources. Given that a user needs a fraction γ_u/c_u of the base station's resources to meet her rate requirement, this policy imposes that for slice o the following constraint is satisfied at each base station b:

$$\sum_{u \in \mathcal{U}_b^o} \frac{\gamma_u}{c_u} \leq s_o. \tag{4}$$

The WAC policy aims at ensuring that (4) is satisfied at all times. However, even if this condition holds when a new user is admitted, it may be subsequently violated upon changes in the slice, e.g., due to mobility of users or changes in their c_u.

To provide robustness against such changes, we follow the approach in [33] for single-tenant networks. Specifically, we add a guard band to (4) aimed at ensuring that the condition will continue to hold with high probability after any changes. Thus, a slice admits a new user request as long as the following holds

$$\sum_{u \in \mathcal{U}_b^o} \frac{\gamma_u}{c_u} \leq \rho_w \cdot s_o,$$

where $\rho_w < 1$ parametrizes the guard band: the smaller this parameter, the larger the guard band. In practice, this parameter may be set to different values by different slices based on the slice specifics, such as the fluctuations of c_u or user association (where larger fluctuations will require a larger guard band) or the desired level of assurance to its users (stricter guarantees will require a larger guard band). The reader is referred to [33] for a discussion on how to set this parameter.

In the following, we analyze the properties of WAC under the assumption that (4) is satisfied with this policy. The theorem below shows that, as long as this condition is satisfied, a slice will always be able to meet its users' rate guarantees independent of the setting of the other slices. Thus, a high degree of protection to the choices and changes in other slices is provided. The theorem also shows that if the slice deviates from the proposed policy, it is not protected from the other slices' choices, implying that this policy represents a necessary condition to provide protection.

Theorem 3. *Consider a slice o with users having rate requirements $\boldsymbol{\gamma}^o = (\gamma_u : u \in \mathcal{U}^o)$, then the following hold:*

1) *If* (4) *is satisfied, there exists at least one weight allocation $\mathbf{w}^o$ such that $\forall u \in \mathcal{U}^o$ $r_u(\mathbf{w}) \geq \gamma_u$, for any feasible allocation of the other slices' aggregate weights $\mathbf{a}^o$.*
2) *If* (4) *is not satisfied, slice o is not protected, as there is a feasible $\mathbf{a}^o$ allocation such that slice o is not able to meet the rate requirements of its admitted users.*

Note that combining this result with Theorem 2, it follows that a NE exists when all slices run WAC. Indeed, the above theorem ensures that a slice can find an allocation that meets the rate requirements of all its users for any feasible $\mathbf{a}^o$, which comprises all the possible allocations of the other slices $\mathbf{w}^{-o}$. Theorem 2 guarantees that when this holds, a NE exists. Thus, we have the following corollary:

Corollary 2. *If* (4) *is satisfied by all slices, then the network slicing game has a NE.*

Note that Corollary 2 imposes more conservative conditions than Theorem 2; for instance, if a slice never has users at a given base station, according to Theorem 2 such a slice cannot place any weight on this base station; in contrast, the arguments behind (4) account for, and protect the slice against, such possibility.

C. Load-driven admission control (LAC)

While the WAC policy protects a given slice from the others, it may be overly conservative in some cases where base stations are lightly loaded or where some slices are unlikely to use resources at certain base stations. In those cases, one may opt to be more aggressive in admitting users without running significant risks. To this end, we propose the Load-driven Admission Control (LAC) policy, where a slice measures the current load across base stations and performs admission control decisions based on the measured loads (assuming that they will not change significantly).[3]

The following theorem provides a basis for the design of the LAC policy. It gives a necessary and sufficient condition that has to be satisfied to meet the rate requirements of the slice's users, given the current weight allocations of the other slices. This constraint is shown to be less restrictive than the one imposed by (4), implying that LAC (potentially) allows the admission of more users than WAC.

Theorem 4. *Consider a slice o comprising users with rate requirements $\boldsymbol{\gamma}^o = (\gamma_u : u \in \mathcal{U}^o)$, and suppose the aggregate weight of the other slices is given by $\boldsymbol{a}^o$. Then, a weight allocation $\mathbf{w}^o$ that meets slice o's rate requirements exists if and only if the following is satisfied:*

$$\sum_{b \in \mathcal{B}} \frac{\sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u}{1 - \sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u} a_b^o \leq s_o. \tag{5}$$

where $\mathcal{U}_b^o$ is the subset of users of slice o associated with base station b, according to the given user association policy.

Moreover, if the rate requirements satisfy (4)*, then the above condition is satisfied.*

The central idea of the LAC policy is as follows. Upon receiving a request of a new user u with a rate requirement γ_u, slice o assesses the current $\boldsymbol{a}^o$ values in the network and checks whether (5) would be satisfied with the new user. According to the theorem, as long as (5) is satisfied, the rate requirements can be met *if* the $\boldsymbol{a}^o$ values do not change. However, in practice $\boldsymbol{a}^o$ may change due to the response of the other slices to slice o, or to changes in the other slices (e.g., the admission of new users). We shall address this uncertainty by following a similar approach to WAC: when admitting a new user, we verify that (5) is satisfied with a sufficiently large guard band, i.e.,

$$\sum_{b \in \mathcal{B}} \frac{\sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u}{1 - \sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u} a_b^o \leq \rho_l \cdot s_o, \tag{6}$$

where $\rho_l < 1$ is the parameter providing the guard band for LAC. Note that, in addition to other considerations, in this case the setting of ρ_l will need to account for observed statistical fluctuations of $\boldsymbol{a}^o$, larger fluctuations requiring a larger guard band.

The following theorem shows that, as long as the chosen value for ρ_l is sufficiently conservative, LAC is effective in guaranteeing that the rate requirements of all users are met.

Theorem 5. *There exists a ρ_l value sufficiently small such that the rate requirements of all the users of slice o can be met independent of how the other slices change their weights.*

[3]Note that many similar (load-driven) admission control algorithms have been proposed in the literature [34], [35] in the context of single-tenant networks. In this paper, we apply this concept to a network slicing setting.

The following corollary follows from the above result and Theorem 3. Indeed, as long as every slice satisfies either (4) and (6), Theorems 3 and 5 guarantee that all slices can choose a weight allocation that satisfies the rate requirements of all their users. Furthermore, Theorem 2 guarantees that when this holds there exists a NE. These implies that, as long as all slices run either WAC or LAC, the system can be expected to be stable.

Corollary 3. *If either* (4) *or* (6) *holds for every slice (the latter with a sufficiently small ρ_l), then there exists a NE.*

IV. Weight allocation and user dropping for Network Slicing

Once a slice decides which users to admit, possibly following one of the admission control policies presented above, it needs to determine the weight allocation of the admitted users. In NES, this is determined based on a sequence of best responses, where in each round a slice chooses its best response given the choices of the other slices. A slice's best response involves the following two steps: (i) *user subset selection*, to determine which subset of users to serve, and (ii) *weight allocation*, to set the weights of the users in the selected subset. In the following, we first present the algorithms to perform the user subset selection and weight allocation, and then analyze the convergence of the sequence of best responses.

A. User subset selection

When a slice cannot satisfy the rate requirements of all its users, it needs to decide which subset to serve. Note that, while admission control aims at ensuring that rate requirements of all users can always be satisfied, in practice this can only be ensured with a (very) high probability due to the unpredictable nature of the mobile network; thus, in some unlikely cases it may happen that the rate requirements of some users cannot be met. When this happens, the slice has to drop those users. Note that this yields a novel paradigm for managing the resources of a slice, where changes in one part of the network may lead to dropping users in another part.

Below we present the algorithms for two possible approaches for user selection: (i) *MaxSubsetSelection*, which maximizes the cardinality of the subset of served users (thus minimizing user dropping); and (ii) *PriorityUserSelection*, which uses a priority ordering on a slice's users (enabling a slice to customize its users' service).

To realize *MaxSubsetSelection* we use a greedy algorithm which at each step adds the user which needs the smallest additional weight to meet the selected users' rate requirements. To that end, let $\tilde{\mathcal{U}}^o$ be a candidate subset of the admitted users by slice o, $\mathcal{U}^o$, and let $\omega_b^o(\tilde{\mathcal{U}}^o)$ be the minimum aggregate weight required to satisfy the rate requirements the candidate subset's users on base station b, $\tilde{\mathcal{U}}_b^o$. The value of $\omega_b^o(\tilde{\mathcal{U}}^o)$ can be computed as follows. The minimum weight w_u needed to satisfy the rate requirement of user $u \in \tilde{\mathcal{U}}_b^o$ must satisfy $w_u c_u / l_b = \gamma_u$; summing these over $u \in \tilde{\mathcal{U}}_b^o$ and isolating $\sum_{u\in\tilde{\mathcal{U}}^o} w_u$ yields

$$\omega_b^o(\tilde{\mathcal{U}}^o) = a_b^o(\mathbf{w}^{-o}) \frac{\sum_{u\in\tilde{\mathcal{U}}_b^o} \gamma_u/c_u}{1 - \sum_{u\in\tilde{\mathcal{U}}_b^o} \gamma_u/c_u}.$$

where we are assuming $\sum_{u\in\tilde{\mathcal{U}}^o} \gamma_u/c_u \leq 1$ (otherwise we let $\omega_b^o(\tilde{\mathcal{U}}^o) = \infty$).

We further let $\omega^o(\tilde{\mathcal{U}}^o) = \sum_{b\in\mathcal{B}} \omega_b^o(\tilde{\mathcal{U}}^o)$ denote the aggregate minimal weight requirement for the slice, and for any user $u' \in \mathcal{U}^o$ we define the marginal aggregate weight of the user u' given candidate subset $\tilde{\mathcal{U}}^o$ as

$$\Delta\omega^o(\tilde{\mathcal{U}}^o, u') = \omega^o(\tilde{\mathcal{U}}^o \cup \{u'\}) - \omega^o(\tilde{\mathcal{U}}^o).$$

Building on the above notation, we present a greedy solution in Algorithm 1, which provides as output the set of selected users $\tilde{\mathcal{U}}^o$. The following theorem confirms the effectiveness of this algorithm.

Algorithm 1: MaxSubset Algorithm.

Initialize: $\tilde{\mathcal{U}}^o = \emptyset$
while $\tilde{\mathcal{U}}^o \neq \mathcal{U}^o$ **do**
  $u^* = \text{argmin}_{u'}\{\Delta\omega^o(\tilde{\mathcal{U}}^o, u') \mid u' \in \mathcal{U}^o \setminus \tilde{\mathcal{U}}^o\}$
  if $\omega^o(\tilde{\mathcal{U}}^o \cup \{u^*\}) \leq s_o$ **then** $\tilde{\mathcal{U}}^o := \tilde{\mathcal{U}}^o \cup \{u^*\}$;
  else return;
end

Theorem 6. *The MaxSubsetSelection algorithm results in a subset of users that maximizes the number of users the slice can serve and still meet their minimal rate requirements.*

Alternatively, slices might apply a *PriorityUserSelection* algorithm to customize their user subset selection policy by assigning users a priority order. Such an ordering may depend, e.g., on the users' traffic class, the revenue they generate, how long users have been in the system, and/or their current signal to noise ratio, among other factors. To this end, the algorithm simply adds users sequentially to the subset to be served in order of decreasing priority until no more can be added, i.e., $\omega^o(\tilde{\mathcal{U}}^o \cup \{u^*\}) > s_o$.

B. Weight allocation

Once a slice has selected a set of users whose requirements can be satisfied, it sets their weights as follows. Given the aggregate weights of the other slices, $a_b^o(\mathbf{w}^{-o})$, a slice chooses $\mathbf{w}^o$ such that the its utility is maximized, i.e.,

$$\begin{aligned}\mathbf{w}^o = \arg\max_{\mathbf{w}'^o} & \sum_{u\in\tilde{\mathcal{U}}^o} U^o(\mathbf{w}'^o, \mathbf{w}^{-o}),\\ \text{s.t.: } & \frac{w'_u}{a_b^o(\mathbf{w}^{-o}) + l_b^o(\mathbf{w}'^o)} \geq \frac{\gamma_u}{c_u},\ \forall u \in \tilde{\mathcal{U}}^o,\\ & w'_u \geq \delta,\ \forall u \in \tilde{\mathcal{U}}^o,\ \sum_{u\in\tilde{\mathcal{U}}^o} w'_u \leq s_o.\end{aligned}$$

where, for convenience, we write $U^o(\mathbf{w}'^o, \mathbf{w}^{-o}) = U^o(\mathbf{w})$ to highlight dependencies on other slices weights.

Note that as long as utility functions $f_u(\cdot)$ are concave in the allocated user rates, the above maximization corresponds to a (computationally tractable) convex optimization problem.

C. Convergence of best response dynamics

With NES, we determine users' weight allocation based on a sequence of best responses. The proposed algorithm implements the best response computed above in rounds: slices update the weight allocation of their users $\mathbf{w}^o$, sequentially, one at a time and in the same fixed order, in response to the other slices weights $\mathbf{a}^o$. Following standard game theory terminology, we refer to this iterative process as *Best Response Dynamics*.

The following theorem shows that the above dynamics may not converge. In particular, the proof of the theorem considers an instance satisfying the conditions of Theorem 2, i.e., a feasible instance under admission control, and shows that, even though a NE is guaranteed to exist under such conditions, Best Response Dynamics do not converge.

Theorem 7. *Consider a game instance such that, for each slice $o \in \mathcal{O}$ there exists an allocation satisfying the rate requirements of all its users for any possible allocation of the other slices. Even though a NE is guaranteed to exist under these conditions, Best Response Dynamics may not converge.*

While the above theorem shows that convergence cannot be ensured, our simulation results show that in practice Best Response Dynamics converge quickly to a region close to the NE, and hence we can simply force the system to halt after a number of best response rounds and use the weights obtained in the last round. Specifically, following the results provided in Section VI-D, in our simulations we halt the system after 7 rounds.

From the above, it can be seen that NES incurs an acceptable computational load, as its execution involves solving a sequence of convex optimization problems (each of which scales with the number of users of the slice and number of base stations) for a limited number of times (namely, the number of slices in the network multiplied by 7). Moreover, the above computations may be possibly performed at centralized controllers, as the resource allocation does not need to be implemented in the base stations before the sequence of optimizations converges or stops. Also, resources may be reallocated only periodically to alleviate the overhead associated to the reconfiguration of base stations. Quantitative results on the computational load are provided in Section VI-E.

V. Analysis of the NES framework

In the following, we analyze the performance achieved by the NES approach proposed above as compared to the two baseline allocations given in Section II-C: (i) the socially optimal allocation, and (ii) static slicing. Our analysis assumes that NES reaches a Nash equilibrium.

A. Gain over static slicing

The result below shows that NES outperforms static slicing.

Theorem 8. *For the same set of admitted users, the utility achieved by an operator under NES is never lower than the utility that this operator would obtain under static slicing.*

While the theorem assumes the same set of admitted users for static slicing and NES, we argue that the result holds in general. Indeed, a tenant is free to choose any admission control policy, including that employed by static slicing, and it is to be expected that it will apply the policy that maximizes its utility. Thus, it follows that the level of satisfaction of the tenant will be greater with NES, under the chosen admission policy, than with static slicing.

B. Loss over the socially optimal allocation

We now study the difference in the utility achieved under socially optimal resource allocation vs. that achieved under NES. We focus on the case where $f_u(\cdot)$ follows (2) for $\alpha_o = 1$ and $\alpha_o = 2$, which are two highly relevant settings in practice (corresponding to proportional and minimum delay potential fairness, respectively). To perform the comparison, we define the Loss over the Social Optimal (LSO) as follows. For $\alpha_o = 1$ we define $LSO \doteq U(\mathbf{w}^*) - U(\hat{\mathbf{w}})$, where $\mathbf{w}^*$ is the socially optimal weight allocation and $\hat{\mathbf{w}}$ is the weight allocation with NES, while for $\alpha_o = 2$ we define it as $LSO \doteq \frac{U(\hat{\mathbf{w}})}{U(\mathbf{w}^*)}$. Note that these definitions are adjusted to the type of utility function: for $\alpha_o = 1$, utilities are logarithmic in the rate, and hence by subtracting utilities we capture the ratio between rates, while for $\alpha_o = 2$ utilities are inversely proportional to the rates, and hence the ratio between rates is obtained by dividing utilities.

The following theorem provides a bound on the LSO and gives an instance for which the LSO is close to this bound, showing that the bound is tight.

Theorem 9. *Let user utilities $f_u(\cdot)$ follow* (2)*, $\underline{\gamma}_u$ be the minimum rate guarantee in the network, $\bar{c}_u$ be the largest possible achievable rate and $\varepsilon = \underline{\gamma}_u / \bar{c}_u$. Under a given set of admitted users, we have that:*

1) *If $\alpha_o = 1 \ \forall o \in \mathcal{O}$, then $LSO \leq -\log(\varepsilon)$ and there is an instance for which $LSO \geq -\frac{1}{2}\log(2\varepsilon)$.*
2) *If $\alpha_o = 2 \ \forall o \in \mathcal{O}$, then $LSO \leq \frac{1}{\varepsilon}$ and there is an instance for which $LSO \geq \frac{1}{3\varepsilon}$.*

Note that, according to the above results, the bound on the LSO relaxes as we decrease the minimum rate requirement in the network, and becomes unbounded in the case where we have elastic traffic with no rate guarantees, i.e., $\gamma_u = 0$. However, in a well provisioned network all users should experience a sufficiently large rate, and in this case the LSO should be low according to the above result. This is corroborated by our simulation results, which show that in practice NES performance is close to optimal and LSO is very small.

VI. Performance evaluation

We next evaluate the performance of NES via simulation. Unless otherwise stated, the mobile network setup of our simulator follows the IMT-A evaluation guidelines for dense 'small cell' deployments [36], considering a network with 19 base stations disposed in a hexagonal grid layout with 3 sectors, i.e., $|\mathcal{B}| = 57$. User mobility follows the Random Waypoint (RWP) model. The users arrive to the network following a Poisson Process with intensity λ arrivals/sec, and their holding times are exponentially distributed. Users' SINR

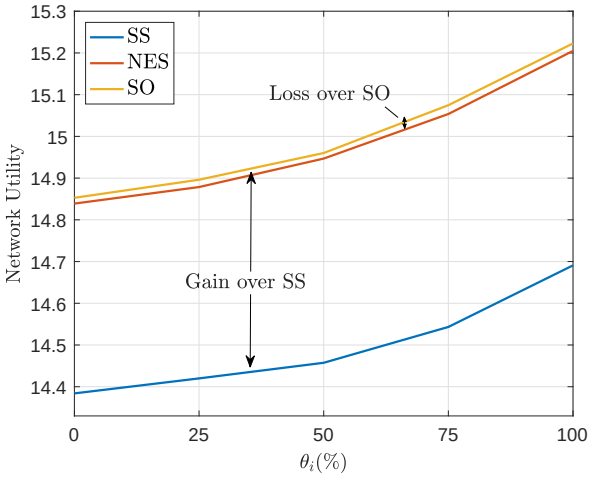

Fig. 1. Performance of NES in terms of network utility as compared to the two benchmark allocations (SS and SO).

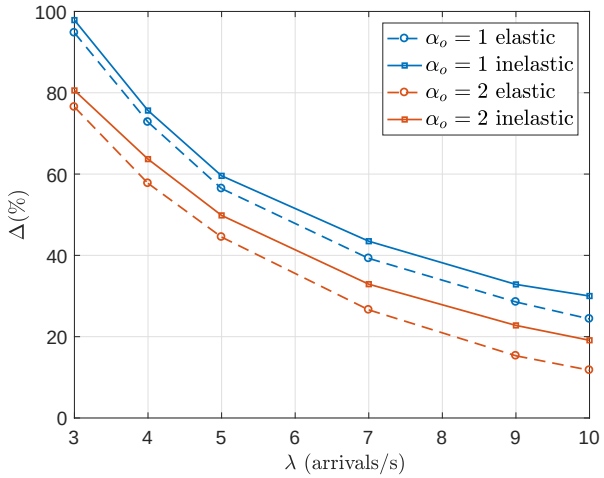

Fig. 2. Throughput gains over SS for different traffic types (elastic, inelastic), utility functions (α_o) and network load (λ).

is computed based on physical layer network model specified in [36] (which includes path loss, shadowing, fast fading and antenna gain) and user association follows the strongest signal policy. The achievable rate for a user u, c_u, is determined based on the thresholds reported in [37]. Unless otherwise stated, the rate requirement of the inelastic users is set to $\gamma_u = 0.5$ Mbps, we have $\alpha_o = 1$ for all slices, there are 5 slices in the network with equal shares, the arrival rate is $\lambda = 5$ (equally split among slices) and the average holding time is 1 minute. In the simulations, we consider both slices with mixed traffic of different types (Sections VI-A and VI-C) as well as slices dedicated to one specific traffic type (Section VI-B). All confidence intervals are below 1%.

A. Network utility

We first analyze the network utility achieved by NES as compared to the two benchmark solutions presented in Section II-C (namely, SS and SO). To ensure that the rate requirements of admitted users are always met, we adopt the WAC admission control policy with $\rho_w = 1$ and suppress user movements yielding changes in base station associations and/or c_u values. To analyze the impact of inelastic traffic, we vary the fraction of inelastic traffic arrivals, θ, yielding an arrival rate of $\theta\lambda$ for inelastic users and of $(1-\theta)\lambda$ for elastic ones. The results, depicted in Fig. 1, show that (i) NES outperforms very substantially SS, providing very high gains, and (ii) it performs almost optimally, very close to the SO. Moreover, this holds independently of the mix of elastic and inelastic users present in the network.

B. Throughput gains

To give a more intuitive measure of the gains achieved by NES, we define the throughput gain over SS, Δ, as follows: it is the value such that, if we increase the rate of all users in SS by Δ, we reach the same network utility as NES (e.g., $\Delta = 100\%$ means that SS achieves the same utility as NES when multiplying all user rates by 2). Fig. 2 illustrates the throughput gains for (i) $\alpha_o = 1$ and $\alpha_o = 2$, which are the two most relevant α_o values in practice, (ii) elastic and inelastic slices, where all users are either elastic and inelastic, and (iii) different arrival rates λ, yielding different network loads. We conclude from the results that (i) gains are very substantial, ranging from 100% to 20%, (ii) they decrease with the load, as already observed in [4], and (iii) they are fairly insensitive to the fraction of inelastic traffic and choice of utility function.

C. Blocking probability

In addition to improving the performance of admitted users, one of the key advantages of the dynamic resource allocation implemented by NES is that it allows admitting more users while meeting their rate requirements. In order to assess the achieved improvement, we evaluate the blocking probability (i.e., the probability that a new user cannot admitted) under NES versus SS. For NES, we consider the two admission policies proposed in Section III (WAC and LAC), while for SS we apply the policy given in [33]. For all settings, we drop users based on the *MaxSubsetSelection* algorithm, and adjust the guard bands to ensure that the probability of dropping an admitted user is no more than 1%. To increase the offered load sufficiently so that we can observe the behavior of the blocking probability, we set $\gamma_u = 1$ Mbps and an average holding time of 2 minutes. The results are given in Fig. 3 as a function of the fraction of inelastic user arrivals (θ). They show very high gains over SS for both approaches (WAC and LAC), and confirm that, by behaving more aggressively, LAC is able to admit many more users than WAC.

D. Convergence to the NE

To better understand the dynamics of NES, we have evaluated a very large number of randomly generated scenarios (namely 10^4 scenarios) with the following settings: (i) a uniform number of slices between 2 and 10, i.e., $|\mathcal{O}| \sim U(2,10)$, (ii) a number of users per slice of $|\mathcal{U}^o| \sim U(0,350)$, (iii) inelasticity level $\theta \sim U(0,100)(\%)$, (iv) minimum rate requirements $\gamma_u \sim U(0,3)$ Mbps, and (v) the shares s_o proportional to the number of users. We have found that a

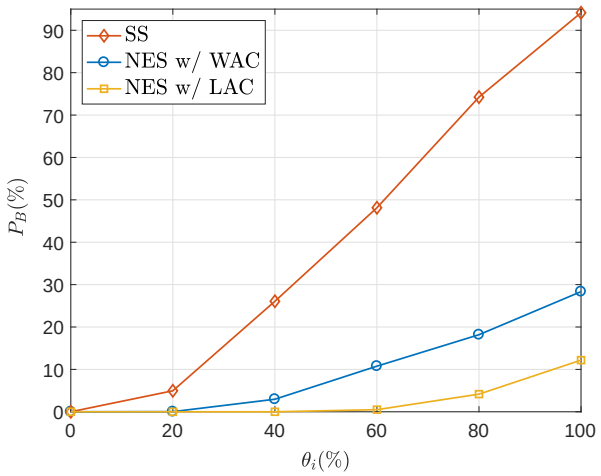

Fig. 3. Blocking probability for new arrivals for the two policies proposed and the SS benchmark.

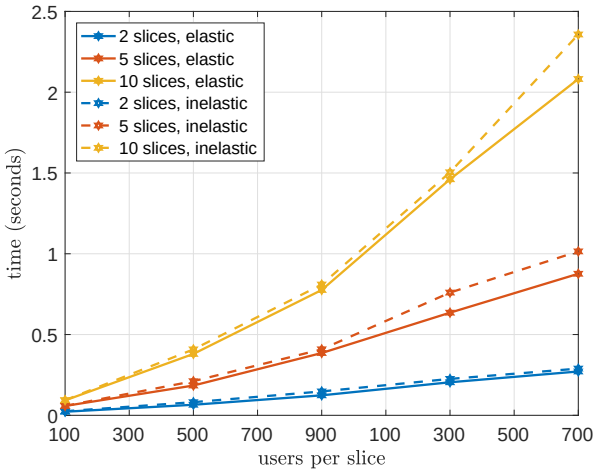

Fig. 5. Computational times of the proposed approach as a function of the number of slices and users in the network.

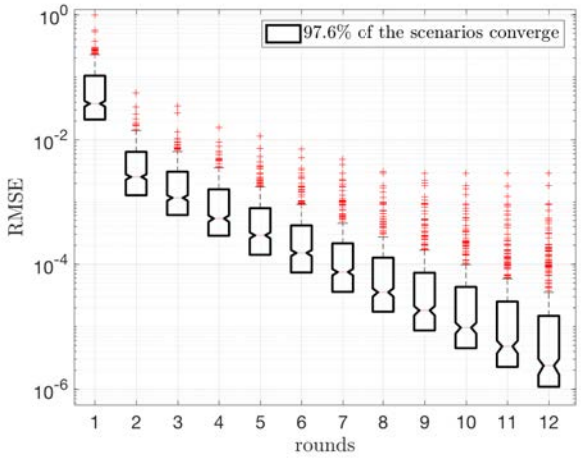

Fig. 4. Box plot for the RMSE of the weight allocation at a given round with respect to the NE weight allocation.

vast majority (97.6%) of scenarios converge to the NE after 100 rounds. For such scenarios, Fig. 4 shows the difference between the weight allocation at a given round and the one at the NE in terms of mean squared error (RMSE), providing a box plot with the median (red), 95% percentile (box), 99% percentile (whisker) and outliers (red crosses). We observe that the RMSE decreases exponentially in the number of rounds. After 7 rounds we are already very close to the NE (the median is below 10^{-4}), which justifies our choice in Section IV-C. Additional results, not included for space reasons, show that user rates exhibit a very similar behavior to the weights.

E. Computational load

Next we evaluated the computational complexity of the NES algorithm when the system halts after 7 rounds (as given by the configuration chosen in this paper). Fig. 5 shows the computational times for a dual-core 2.9GHz i7 processor for elastic and inelastic traffic and different numbers of slices and users, when the number of base stations is scaled with the number of users and admission control is adjusted to ensure that dropping probabilities below 1%. Results confirm that NES can be applied to practical settings, as complexity is roughly linear with the size of the network and computational times remain low even for large size problems; for instance, for a network with 9000 users the time falls below 2.5 seconds. We further observe that inelastic traffic slightly increases complexity but does not challenge the practicality of the approach. Finally, we note that the computational time values provided here could be further improved by optimizing the code, parallelizing tasks and/or increasing the machine computational power.

F. Slice differentiation

We next analyze the ability of NES to deploy slices providing a customized service. To this end, we consider a scenario with 4 slices with different requirements: (i) slice 1 provides rate requirements of $\gamma_u = 1$ Mbps with WAC, (ii) slice 2 provides $\gamma_u = 0.5$ Mbps with WAC, (iii) slice 3 provides $\gamma_u = 0.5$ Mbps with LAC, and (iv) slice 4 provides no minimum rate requirements. All slices have the same share, the arrival rate is of $\lambda = 10$ equally split among the slices, and admission control is configured to provide dropping probabilities below 1%. Fig. 6 shows the empirical CDF of the user rates for each slice as well as the blocking probabilities ($\approx 47.2\%, 16.7\%, 3.58\%$ and 0%, respectively). We observe that (i) the minimum rate requirements are satisfied for all slices; (ii) as the rate requirements increase, so does the blocking probability, yielding an overall improvement of the user rate distribution, and (iii) by employing LAC, we achieve a dramatic reduction of the blocking probability while paying a very small prices in terms of user rate distribution. We conclude that NES is effective in enabling slice differentiation.

VII. Conclusions

In this paper we proposed and analyzed a framework for network slicing that relies on network shares and allows slices to customize resource allocations to their users. This framework results in a *network slicing game* where each slice unilaterally reacts to the settings of the others. While this

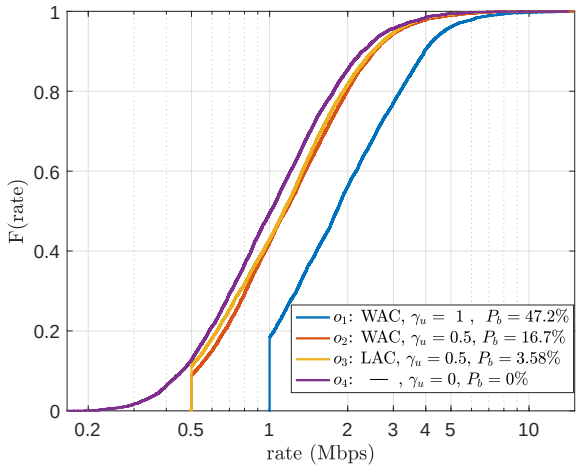

Fig. 6. Blocking probability and empirical CDF of the user rates for a scenario of 4 slices with different requirements.

game has been previously studied for elastic traffic, the slices' behavior changes substantially when users have minimum rate requirements, and so does the outcome of the game. Indeed, we have shown that (in contrast to the elastic case) this game may not have a Nash Equilibrium and, even when it has a NE, Best Response Dynamics may not converge to the equilibrium. In spite of this (apparently) negative result, we have shown that as long as admission control is applied (which is to be expected under inelastic traffic), we can guarantee that a NE exists. We have proposed algorithms for admission control, weight allocation and user dropping, which jointly bring the system to a NE. We have further analyzed performance at the equilibrium, showing that it is close to the social optimal and provides substantial gains over static slicing. Based on these results, our main conclusion is that the proposed NES framework provides an *effective* and *implementable* scheme for dynamically sharing resources across slices, both for elastic and for inelastic traffic.

APPENDIX: PROOFS OF THE THEOREMS

Proof of Theorem 1

Consider a setting with two base stations (a and b) and two slices (1 and 2), each slice with one user associated to base station a and another user associated to base station b. We refer to these users as $\mathcal{U} = \{1a, 1b, 2a, 2b\}$. Let the rate requirements of slice 1 be $\gamma_{1a} = \gamma_{1b} = 2C/3$, the users of slice 2 have no minimum rate requirements, and $s_1 = s_2 = 1/2$. We show that this game has no NE by contradiction. We necessarily have that either $w_{2a} \leq 1/4$ or $w_{2b} < 1/4$. Let us assume that $w_{2a} < 1/4$ and $w_{2b} > 1/4$. Since in this case slice 1 can only meet the rate requirements of user 1a, its best response will concentrate its weight on this user, $w_{1a} = 1/2$. However, the best response of slice 2 to such allocation of slice 1 is to concentrate its share on user 2a. Thus, $w_{2a} > 1/4$, which contradicts the initial assumption. Following a similar argument, it can be seen that if we assume $w_{2a} = 1/4$ or $w_{2a} > 1/4$, we also reach a contradiction. □

Proof of Theorem 2

Let $\mathcal{W}$ be the convex and compact set of feasible weights $\mathbf{w}$ satisfying (i) $w_u \geq \delta$ $\forall u$, and (ii) $\sum_{u \in \mathcal{U}_o} w_u = s_o$ $\forall o$ and let us consider the mapping $\mathbf{w} \to \tilde{\mathbf{w}} = \Gamma(\mathbf{w})$, where $\tilde{\mathbf{w}}^o$ is the best response of slice o to $\mathbf{w}^{-o}$. We next show that this mapping satisfies the conditions of Kakutani's theorem: $i)$ $\Gamma(\mathbf{w})$ is non-empty, $ii)$ $\Gamma(\mathbf{w})$ is a convex-valued correspondence, and $iii)$ $\Gamma(\mathbf{w})$ has a closed graph. Conditions $i)$ and $ii)$ follow from the fact that the best response of a slice to $\mathbf{w}^{-o}$ is a unique allocation $\tilde{\mathbf{w}}^o$. This implies that that $\tilde{\mathbf{w}}$ exists and is a single point (and hence a convex set). Condition $iii)$ is shown by proving that $\tilde{\mathbf{w}}^o$ is a continuous function of $\mathbf{w}^{-o}$ for all slices. Consider the set of users for which $r_u > \gamma_u$ and the set for which $r_u = \gamma_u$. As long as these sets do not change, $\tilde{\mathbf{w}}^o$ can be expressed as a continuously differentiable function of $\{\tilde{\mathbf{w}}^o, \mathbf{w}^{-o}\}$, and it follows from the implicit function theorem that $\tilde{\mathbf{w}}^o$ is a continuous function of $\mathbf{w}^{-o}$. When some user moves from set $r_u > \gamma_u$ to $r_u = \gamma_u$ (or viceversa), such user satisfies both the equation for $r_u = \gamma_u$ and the one for $r_u > \gamma_u$, providing continuity over the transitions. Since all the conditions of Kakutani's theorem are satisfied, we have that the mapping Γ has at least one fixed point, which implies that at least one NE exists.

To show that the NE is not necessarily unique, we provide an example with multiple NEs. Consider a scenario with three slices (1,2,3) and three base stations (a,b,c). Let the first slice have users in base stations a and c (users $1a$, $1c$), the second slice in a and b ($2a$, $2b$) and the third slice in b and c ($3b$, $3c$). Let $\phi_{1a} = \phi_{1b} = 1/2$, $\phi_{2a} = \phi_{3c} = 1$ and $\phi_{2b} = \phi_{3b} = 0$. Also, let $\gamma_u = 1/2$ for users $2b$ and $3b$, $\gamma_u = 0$ for all other users and $c_u = 1$ for all users. It can be seen that all the weight allocations satisfying $w_{1a} = w_{1b} = 1/6$, $w_{2b} = w_{3b} = w$ and $w_{2a} = w_{3c} = 1/3 - w$ for $w \in [\delta, 1/3 - \delta]$ correspond to a NE, which shows that multiple NE exist for this example. □

Proof of Theorem 3

The result of 1) follows directly from Lemma 1 in [4]. If users are admitted at base stations such that under static slicing their rate guarantees are met, i.e. $r_u^{ss} \geq \gamma_u$, then it follows by the above mentioned lemma that there exists an allocation satisfying $r_u \geq r_u^{ss} \geq \gamma_u$, which proves the first part of the theorem.

To prove 2), we proceed as follows. Suppose slice o admits users are such that their associated rate requirements violate (4) at some base station b, i.e., $\sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u > s_o$. If all other slices place their entire share at that base station, we have

$$\sum_{u \in \mathcal{U}_b^o} \frac{r_u}{c_u} = \frac{\sum_{u \in \mathcal{U}_b^o} w_u}{\sum_{u \in \mathcal{U}_b^o} w_u + 1 - s_o} \leq s_o,$$

which implies $\sum_{u \in \mathcal{U}_b^o} r_u / c_u < \sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u$ and hence necessarily $r_u < \gamma_u$ for some u, proving the second part of the theorem. □

Proof of Theorem 4

Recall that the rate of user u is given by $r_u = w_u c_u / l_{b(u)}$. If we add the rates of the users of slice o at a given base station b and isolate $\sum_{u\in\mathcal{U}_b^o} w_u$, we obtain

$$\sum_{u\in\mathcal{U}_b^o} w_u = \frac{\sum_{u\in\mathcal{U}_b^o} r_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o} r_u/c_u} a_b^o.$$

By summing the above over all base stations and noting that $\sum_{u\in\mathcal{U}^o} w_u = s_o$, we obtain

$$\sum_{b\in\mathcal{B}} \frac{\sum_{u\in\mathcal{U}_b^o} r_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o} r_u/c_u} a_b^o = s_o. \tag{7}$$

We now prove that as long as (5) is satisfied, there exists a weight allocation $\mathbf{w}^o$ that meets the rate requirements of all users. Let us consider the weight allocation satisfying[4]

$$w_u = \frac{(\gamma_u/c_u) l_{b(u)}}{\sum_{v\in\mathcal{U}^o} (\gamma_v/c_v) l_{b(v)}} s_o, \ \forall u \in \mathcal{U}^o. \tag{8}$$

Note that with the above weight allocation, the rates r_u are proportional to γ_u, which means that either we have $r_u \geq \gamma_u \ \forall u$ or $r_u < \gamma_u \ \forall u$. The latter yields a contradiction; indeed, if $r_u < \gamma_u \ \forall u$ it follows that

$$\sum_{b\in\mathcal{B}} \frac{\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u} a_b^o > \sum_{b\in\mathcal{B}} \frac{\sum_{u\in\mathcal{U}_b^o} r_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o} r_u/c_u} a_b^o = s_o,$$

which contradicts (5). Hence, it follows that $r_u \geq \gamma_u \ \forall u$.

We next prove that if (5) is not satisfied, then there exists no weight allocation meeting the rate requirements. The proof goes by contradiction. Assume (5) is not satisfied but $r_u \geq \gamma_u \ \forall u$. From the latter, it follows that $\sum_{u\in\mathcal{U}_b} r_u/c_u \geq \sum_{u\in\mathcal{U}_b} \gamma_u/c_u \ \forall b$. Combining this with (7) yields

$$\sum_{b\in\mathcal{B}} \frac{\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u} a_b^o \leq s_o,$$

which contradicts that assumption that (5) is not satisfied.

Finally, we show that if the rate requirements satisfy (4), then they surely satisfy (5). The lhs of (5) increases with $\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u$. As long as this value is no larger than s_o, we have that the following equation gives a sufficient condition for (5) to be satisfied: $\frac{s_o}{1-s_o}\sum_{b\in\mathcal{B}} a_b^o \leq s_o$.

The above is surely satisfied since $\sum_{b\in\mathcal{B}} a_b^o = 1 - s_o$. As (4) imposes $\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u \leq s_o$, it follows that as long as (4) is satisfied, (5) is also satisfied. □

Proof of Theorem 5

Let us take $\rho_l = \min_b \frac{a_b^o}{1-\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u}$. Then, from (6) it follows that $\sum_{b\in\mathcal{B}}\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u \leq s_o$. From this, we have that condition (4) is satisfied. According to Theorem 3, as long as this condition is satisfied, there exists a choice of $\mathbf{w}^o$ that satisfies the rate requirements of all users of slice o independent of the weight setting of the other slices, which completes the proof. □

[4]The existence of such an allocation follows from applying Brouwer fixed-point theorem to the function $\mathbf{f}: \mathcal{W} \to \mathcal{W}$, where $w_u = f_u(\mathbf{w})$ is given by (8) and $\mathcal{W}$ is the set of weights satisfying $\sum_{u\in\mathcal{U}^o} w_u = s_o$ and $w_u \geq (\gamma_u/c_u) a_b^o s_o / \sum_{v\in\mathcal{U}^o} (\gamma_v/c_v)$ (recall that $a_b^o \neq 0 \ \forall b$, as weights cannot be zero).

Proof of Theorem 6

The proof goes by contradiction. Let $\tilde{\mathcal{U}}^o$ be the set of users selected by the *MaxSubsetSelection* algorithm, and let us assume that there exists an alternative feasible user selection $\hat{\mathcal{U}}^o$ such that $|\hat{\mathcal{U}}^o| > |\tilde{\mathcal{U}}^o|$. If we take the set $\hat{\mathcal{U}}^o$ and substitute each user by another one in the base station with smaller γ_u/c_u, the resulting set $\bar{\mathcal{U}}^o$ is feasible and has the same number of users as the original one. Note that set $\bar{\mathcal{U}}^o$ necessarily has some base station b with more users than set $\tilde{\mathcal{U}}^o$ – otherwise $|\hat{\mathcal{U}}^o| > |\tilde{\mathcal{U}}^o|$ would not hold. Let us assume that there exists some other base station b' with fewer users. In this case, let us remove user u from one of the base stations with more users, b, and add user u' in one of the base stations with fewer users, b'. The resulting set remains feasible, as $\Delta\omega_{b'}^o(\bar{\mathcal{U}}^o, u') \leq \Delta\omega_{b'}^o(\bar{\mathcal{U}}^o, u)$ – otherwise *MaxSubsetSelection* would have chosen a different subset of users. We can do this until there are no base station with fewer users than in $\tilde{\mathcal{U}}^o$. The result of these operations is a feasible set where all base stations have as many users or more than $\tilde{\mathcal{U}}^o$, and overall it has more users. However, this yields a contradiction: if such set was feasible, the *MaxSubsetSelection* algorithm would have selected more users. □

Proof of Theorem 7

Let us consider a scenario with three base stations (a,b,c) and three slices (1,2,3), with $s_1 = s_2 = s_3 = 1/3$ and any arbitrary $\alpha_1, \alpha_2, \alpha_3$ values. Let slice 1 have two users associated to base stations a and b (u_{1a},u_{1b}), slice 2 two users associated to base stations b and c (u_{2b},u_{2c}) and slice 3 two users associated to base stations a and c (u_{3a},u_{3c}). Let $c_u = 1 \ \forall u$, $\gamma_{1a} = \gamma_{2b} = \gamma_{3c} = 1/2$, $\gamma_{1b} = \gamma_{2c} = \gamma_{3a} = 0$, $\phi_{1a} = \phi_{2b} = \phi_{3c} \to 0$ and $\phi_{1b} = \phi_{2c} = \phi_{3a} \to 1$. The NE of this instance is $w_u = 1/6 \ \forall u$. However, if we start with $w_{3c} = w < 1/6$ and $w_{3a} = 1/3 - w$, and perform a best response cycle starting starting with slice 1 followed by 2 and 3, it can be seen that this leads to an endless cycle where each slice takes a weight allocation of either $\{w, 1/3-w\}$ or $\{1/3-w, w\}$ at each step (none of which corresponds to the NE). Hence, Best Response Dynamics do not converge for this instance of the game. □

Proof of Theorem 8

The proof follows from Lemma 1 of [4], which shows that, given a slice o and a feasible weight allocation $\mathbf{w}^{-o}$ for the other slices, there exists a weight allocation $\mathbf{w}^o$ for slice o, possibly dependent on $\mathbf{w}^{-o}$, such that the resulting weight allocation $\mathbf{w}$ satisfies $r_u(\mathbf{w}) \geq r_u^{ss}$ for all $u \in \mathcal{U}^o$. Therefore, there exists a weight allocation that provides the same utility as static slicing. Since the weight allocation chosen by NES is the one that maximizes the slice's utility, it surely provides a utility no smaller than that under static slicing. □

Proof of Theorem 9

We start for $\alpha_o = 1$. To prove the bound on the LSO, we first note that

$$U(\mathbf{w}^*) = \sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o \phi_u \log\left(\frac{w_u^*}{\sum_{u'\in U_{b(u)}} w_{u'}^*} c_u\right)$$

$$\leq \sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u \log\left(\bar{c}_u\right).$$

Furthermore, from the minimum rate constraint it follows that

$$\begin{aligned} U(\hat{\mathbf{w}}) &= \sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u \log\left(\frac{\hat{w}_u}{\sum_{u'\in U_{b(u)}}\hat{w}_{u'}}c_u\right) \\ &\geq \sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u \log\left(\underline{\gamma}_u\right). \end{aligned}$$

Combining the above two equations, we obtain $U(\mathbf{w}^*) - U(\hat{\mathbf{w}}) \leq \log(\bar{c}_u/\underline{\gamma}_u) = -\log(\varepsilon)$, which completes the first part of the proof.

To show that the above bound is tight, we consider the following network instance. We have two slices with shares $s_1 = s_2 = 1/2$ and two base stations. The first slice has two users in the first base station (weights w_{11} and w_{12}) and the second slice has one user in the first base station (w_{21}) and another one in the second base station (w_{22}). All users have $c_u = \bar{c}_u$, and the rate requirements are $\gamma_{11} = \bar{c}_u(1/2 - \varepsilon)$ for the first user and $\gamma_u = \underline{\gamma}_u = \bar{c}_u\varepsilon$ for the other ones. Furthermore, let $\phi_{11} \to 0$, $\phi_{12} \to 1$, $\phi_{21} \to 0$ and $\phi_{22} \to 1$. In the allocation employed by NES (which corresponds to the NE) we have $w_{11} = 1/2 - \varepsilon$, $w_{12} = \varepsilon$, $w_{21} \to 1/2$ and $w_{22} \to 0$, which yields $U(\hat{\mathbf{w}}) = \frac{1}{2}\log(\varepsilon\bar{c}_u) + \frac{1}{2}\log(\bar{c}_u)$. In the social optimal, we have the following weight allocation: $w_{11} = \left(\frac{1}{2} - \varepsilon\right)\left(\frac{1}{2} + \frac{\varepsilon}{2(1-\varepsilon)}\right)$, $w_{12} = 1/2 - w_{11}$, $w_{21} = \frac{\varepsilon}{2(1-\varepsilon)}$ and $w_{22} = 1/2 - w_{21}$, from which $U(\mathbf{w}^*) = \frac{1}{2}\log\left((1/2)\bar{c}_u\right) + \frac{1}{2}\log(\bar{c}_u)$. This yields $U(\mathbf{w}^*) - U(\hat{\mathbf{w}}) = -\frac{1}{2}\log\left(2\varepsilon\right)$, which terminates the proof for $\alpha_o = 1$.

To prove the LSO bound for $\alpha_o = 2$, we note that

$$U(\mathbf{w}^*) \geq -\sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u\frac{1}{\bar{c}_u}$$

and

$$U(\hat{\mathbf{w}}) =\leq -\sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u\frac{1}{\underline{\gamma}_u}.$$

Combining these two equations we obtain $\frac{U(\hat{\mathbf{w}})}{U(\mathbf{w}^*)} \leq \frac{1}{\varepsilon}$, which completes the first part of the proof. The tightness of the bound is proven by considering the same network instance as for $\alpha_o = 1$:

$$\frac{U(\hat{\mathbf{w}})}{U(\mathbf{w}^*)} = \frac{-\frac{1}{2}\frac{1}{\varepsilon\bar{c}_u} - \frac{1}{2}\frac{1}{\bar{c}_u}}{-\frac{1}{2}\frac{1}{(1/2)\bar{c}_u} - \frac{1}{2}\frac{1}{\bar{c}_u}} = \frac{\frac{1}{\varepsilon}+1}{\frac{1}{1/2}+1} \geq \frac{1}{3\varepsilon}. \qquad \square$$

REFERENCES

[1] NGMN Alliance, "Description of Network Slicing Concept," NGMN 5G P1, Jan. 2016.

[2] 3GPP, "Study on Architecture for Next Generation System," TR 23.799, v0.5.0, May 2016.

[3] ——, "Study on Radio Access Network (RAN) sharing enhancements," TS 22.101 V15.1.0, Jun. 2017.

[4] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez, "Network Slicing Games: Enabling Customization in Multi-Tenant Networks," in *Proc. of IEEE INFOCOM*, May 2017.

[5] L. Zhang, "Proportional response dynamics in the Fisher market," *Theoretical Computer Science*, vol. 412, no. 24, pp. 2691–2698, May 2011.

[6] M. Feldman, K. Lai, and L. Zhang, "The Proportional-Share Allocation Market for Computational Resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1075–1088, Aug. 2009.

[7] J. B. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games," *Econometrica*, vol. 33, no. 3, pp. 520–534, Jul. 1965.

[8] F. P. Kelly, A. K. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, Mar. 1998.

[9] R. Johari and J. N. Tsitsiklis, "Efficiency Loss in a Network Resource Allocation Game," *Mathematics of Operations Research*, vol. 29, no. 3, pp. 407–435, Aug. 2004.

[10] R. Mahindra *et al.*, "Radio Access Network sharing in cellular networks," in *Proc. of IEEE ICNP*, Oct. 2013.

[11] J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs, "Statistical multiplexing and traffic shaping games for network slicing," in *Proc. of WiOpt 2017*, May 2017.

[12] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Prez, "Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3044–3058, Oct 2017.

[13] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic Network Slicing for Multitenant Heterogeneous Cloud Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, Apr. 2018.

[14] R. Matias, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–466, Sep. 2016.

[15] J. Kwak, J. Moon, H. W. Lee, and L. B. L, "Dynamic network slicing and resource allocation for heterogeneous wireless services," in *Proc. of IEEE PIMRC*, Oct. 2017.

[16] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A Service-oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory," *IEEE Access*, vol. 6, pp. 19 691–19 701, Apr. 2018.

[17] O. Narmanlioglu, E. Zeydan, and S. S. Arslan, "Service-Aware Multi-Resource Allocation in Software-Defined Next Generation Cellular Networks," *IEEE Access*, vol. 6, pp. 20 348–20 363, Mar. 2018.

[18] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in *Proc. of European Wireless*, May 2016.

[19] O. U. Akguel, I. Malanchini, V. Suryaprakash, and A. Capone, "Service-Aware Network Slice Trading in a Shared Multi-Tenant Infrastructure," in *Proc. of IEEE GLOBECOM*, Dec. 2017.

[20] L. Zhang, "Proportional response dynamics in the Fisher market," *Theoretical Computer Science*, vol. 412, no. 24, pp. 2691–2698, May 2011.

[21] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[22] O. Candogan, A. Ozdaglar, and P. A. Parrilo, "Dynamics in near-potential games," *Games and Economic Behavior*, vol. 82, pp. 66 – 90, 2013.

[23] 3GPP, "Study on management and orchestration of network slicing for next generation network (Release 15)," TR 28.801 V1.2.0, May 2017.

[24] 3GPP, "Management of 5G networks and network slicing; Concepts, use cases and requirements (Release 15)," TS 28.530, v0.6.0, Apr. 2018.

[25] 5GPPP White paper, "5G Empowering vertical industries," 2016.

[26] A. Gudipati, L. E. Li, and S. Katti, "RadioVisor: A Slicing Plane for Radio Access Networks," in *Proc. of HotSDN*, 2014.

[27] V. Sciancalepore *et al.*, "Interference coordination strategies for content update dissemination in LTE-A," in *Proc. of IEEE INFOCOM*, May 2014.

[28] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sep. 2006.

[29] P. Hande, S. Zhang, and M. Chiang, "Distributed rate allocation for inelastic flows," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1240–1253, Dec. 2007.

[30] 3GPP, "Technical Specification Group Services and System Aspects; Policy and charging control architecture," 3GPP TS 23.203, Jun. 2016.

[31] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, Oct. 2000.

[32] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 27–35, Jul. 2013.

[33] R. Ramjee, D. Towsley, and R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks," *Wireless Networks*, vol. 3, no. 1, pp. 29–41, Mar. 1997.

[34] J. Kim and A. Jamalipour, "Traffic management and QoS provisioning in future wireless IP networks," *IEEE Personal Communications*, vol. 8, no. 5, pp. 46–55, Oct. 2001.
[35] R. D. Callaway, M. Devetsikiotis, and C. Kan, "Design and implementation of measurement-based resource allocation schemes within the realtime traffic flow measurement architecture," in *Proc. of IEEE ICC*, Jun. 2004.
[36] ITU-R, "Report ITU-R M.2135-1, Guidelines for evaluation of radio interface technologies for IMT-Advanced," Technical Report, Dec 2009.
[37] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," TS 36.213, v12.5.0, Rel. 12, Mar. 2015.

Pablo Caballero received his B.S. in telecommunications and his M.S. in telematics engineering respectively from the University Carlos III of Madrid in 2013 and 2015, respectively. In 2015, he joined the Wireless Networking and Communications Group at the University of Texas at Austin to pursue his Ph.D. under the supervision of Profs. Gustavo de Veciana and Albert Banchs. Previously, Pablo worked as Research Assistant at IMDEA Networks Institute and as Research Intern at NEC Laboratories Europe. His research interests lie in the design and performance evaluation of communication networks, game theory and algorithm analysis.

Albert Banchs (M'04-SM'12) received his M.Sc. and Ph.D. degrees from the Polytechnic University of Catalonia (UPC-BarcelonaTech) in 1997 and 2002, respectively. He is currently a Full Professor with the University Carlos III of Madrid (UC3M), and has a double affiliation as Deputy Director of the IMDEA Networks institute. Before joining UC3M, he was at ICSI Berkeley in 1997, at Telefonica I+D in 1998, and at NEC Europe Ltd. from 1998 to 2003. Prof. Banchs is Editor of IEEE Transactions on Wireless Communications and IEEE/ACM Transactions on Networking. His research interests include the performance evaluation and algorithm design in wireless and wired networks.

Gustavo de Veciana (S'88-M'94-SM'01-F'09) received his B.S., M.S, and Ph.D. in electrical engineering from the University of California at Berkeley in 1987, 1990, and 1993 respectively, and joined the Department of Electrical and Computer Engineering where he is currently a Cullen Trust Professor of Engineering. His research focuses on the analysis and design of communication and computing networks; data-driven decision-making in man-machine systems, and applied probability and queueing theory. Dr. de Veciana served as editor and is currently serving as editor-at-large for the IEEE/ACM Transactions on Networking. In 2009 he was designated IEEE Fellow for his contributions to the analysis and design of communication networks. He currently serves on the board of trustees of IMDEA Networks Madrid.

Xavier Costa-Pérez (M'01) is Head of 5G Networks R&D at NEC Laboratories Europe, where he manages several projects focused on 5G mobile core, backhaul/fronthaul and access networks. His team contributes to NEC projects for products roadmap evolution, to European Commission R&D collaborative projects as well as to open-source projects and related standardization bodies, and has received several R&D Awards for successful technology transfers. Dr. Costa-Pérez has served on the Program Committees of several conferences and holds multiple patents. He received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia (UPC-BarcelonaTech) and was the recipient of a national award for his Ph.D. thesis.

Arturo Azcorra (SM'02) received his M.Sc. degree in Telecommunications from UPM in 1986 and his Ph.D. in 1989. In 1993, he obtained an MBA with honors. He is IEEE Senior Member and ACM SIGCOMM Member. He has been visiting researcher at MIT and UC Berkeley. He has participated in 49 research projects since the first Framework Programme of the EU. Azcorra has been the project coordinator of 5G-TRANSFORMER, 5G-Crosshaul, CARMEN, CONTENT and E-NEXT. More information in: http://en.wikipedia.org/wiki/Arturo_Azcorra