Constrained Network Slicing Games: Achieving service guarantees and network efficiency

Jiaxiao Zheng*, Gustavo de Veciana*, Albert Banchs†
*The University of Texas at Austin, TX
†University Carlos III of Madrid & IMDEA Networks Institute, Spain

***Abstract*—Network slicing is a key capability for next generation mobile networks. It enables one to cost effectively customize logical networks over a shared infrastructure. A critical component of network slicing is resource allocation, which needs to ensure that slices receive the resources needed to support their services while optimizing network efficiency. In this paper, we propose a novel approach to slice-based resource allocation named *Guaranteed seRvice Efficient nETwork slicing* (GREET). The underlying concept is to set up a *constrained* resource allocation game, where (i) slices unilaterally optimize their allocations to best meet their (dynamic) customer loads, while (ii) constraints are imposed to guarantee that, if they wish so, slices receive a pre-agreed share of the network resources. The resulting game is a variation of the well-known Fisher market, where slices are provided a budget to contend for network resources (as in a traditional Fisher market), but (unlike a Fisher market) prices are constrained for some resources to provide the desired guarantees. In this way, GREET combines the advantages of a share-based approach (high efficiency by flexible sharing) and reservation-based ones (which provide guarantees by assigning a fixed amount of resources). We characterize the Nash equilibrium, best response dynamics, and propose a practical slice strategy with provable convergence properties. Extensive simulations exhibit substantial improvements over network slicing state-of-the-art benchmarks.**

I. Introduction

There is consensus among the relevant industry and standardization communities that a key element in 5G mobile networks is *network slicing*. This technology allows the network infrastructure to be "sliced" into logical networks, which are operated by different entities and may be tailored to support specific mobile services. This provides a basis for efficient infrastructure sharing among diverse entities, such as mobile network operators relying on a common infrastructure managed by an infrastructure provider, or new players that use a *network slice* to run their business (e.g., an automobile manufacturer providing advanced vehicular services, or a city hall providing smart city services). In the literature, the term *tenant* is often used to refer to the owner of a network slice.

A network slice is a collection of resources and functions that are orchestrated to support a specific service. This includes software modules running at different locations as well as the nodes' computational resources, and communication resources in the backhaul and radio network. By tailoring the orchestration of resources and functions of each slice according to the slice's needs, network slicing enables tenants to share the same physical infrastructure while customizing the network operation according to their market segment's characteristics and requirements.

One of the key components underlying network slicing is the underlying framework for *resource allocation*: we need to decide how to assign the underlying infrastructure resources to each slice at each point in time. When taking such decisions, two major objectives are pursued: (i) meeting the customers' needs specified by slice-based Service Level Agreements (SLAs), and (ii) realizing efficient infrastructure sharing by maximizing the overall level of satisfaction across all slices. Recently, several efforts have been devoted to this problem. Two different types of approaches have emerged in the literature:

Reservation-based schemes [1]–[8] where a tenant issues a reservation request with a certain periodicity or on demand. Each request involves a given allocation for each resource in the network (where a resource can be a base station, a cloud server or a transmission link).

Share-based schemes [9]–[15] where a tenant does not issue reservation requests for individual resources, but rather purchases a share of the whole network. This share is then mapped dynamically to different allocations of individual resources depending on the tenants' needs at each point in time.

These approaches have advantages and disadvantages. Reservation-based schemes are in principle able to guarantee that a slice's requirements are met, but to be efficient, require constant updating of the resource allocations to track changing user loads, capacities and/or demands. The overheads of doing so at a fine granularity can be substantial, including challenges with maintaining state consistency to enable admission control, modifying reservations and addressing handoffs. Indeed these overheads are already deemed high for basic horizontal and/or vertical handoffs. As a result, resource allocations need to be done at a coarser granularity and slower time-scales resulting in reduced overall efficiency and performance.

In contrast to the above, in share-based approaches a slice is given a coarse grain share of the network resources which combined with a fine grain dynamic policy can track rapid changes in a slices' load distributions. Indeed, as these schemes do not involve explicit per resource reservation requests, they can more rapidly adapt allocations to the demand variations of network slices (see, e.g., [16]). Their main drawback, however, is that tenants do not have a guaranteed allocation at individual resources, and as a consequence one cannot ensure that slices' requirements will always be met.

Key contributions: In this paper, we propose a novel approach to resource allocation among network slices named *Guaranteed seRvice Efficient nETwork slicing* (GREET). GREET combines the advantages of the above two approaches while avoiding their drawbacks. The key idea is that a slice is guaranteed a given allocation at each individual resource, as long as the slice needs such an allocation, while the remaining resources are flexibly and efficiently shared. In this way, GREET is able to provide guarantees and thus meet the SLA requirement of each slice, and at the same time it provides a flexible sharing of resources across slices that leads to an overall optimal allocation. Our key contributions are as follows:

- We propose the *GREET slice-based resource allocation framework*, which relies on a *constrained* resource allocation game where slices can unilaterally optimize their allocations under some constraints which guarantee that slices are entitled to a pre-agreed amount of the individual network resources specified in their SLAs (Section II).
- We analyze the resulting network slicing game when slices contend for resources to optimize their performance. We show that the game has a Nash Equilibrium (NE) but unfortunately the Best Response Dynamics (BRD) may not converge to this equilibrium (Section III).
- We propose a *GREET strategy for individual slices* that complements our resource allocation framework. The proposed strategy is simple and provides a good approximation to the slice's best response. We show conditions for convergence with the proposed strategy (Section IV).
- We perform a simulation-based evaluation confirming that GREET combines the best features of reservation-based approaches, providing service guarantees while maximizing overall performance (Section V).

Due to space constraints, we refer the reader to [17] for the proofs of the theoretical results as well as for some additional results.

II. Resource Allocation Approach

In this section we introduce both the system model and the resource allocation framework proposed in this paper.

A. System model

We consider a set of resources $\mathcal{B}$ shared by a set of slices $\mathcal{V}$, with cardinalities B and V, respectively. $\mathcal{B}$ may denote a set of base stations as well as any other sharable resource type, e.g., servers providing compute resources. While our analysis can be applied to different resource types, in what follows we focus on radio resources and refer to $b \in \mathcal{B}$ as a base station.

We assume that each network slice supports a collection of mobile users, possibly with heterogeneous requirements, each of which is associated with a single base station. The overall set of users on the network is denoted by $\mathcal{U}$, those supported by slice v are denoted by $\mathcal{U}^v$, those associated with base station b are denoted by $\mathcal{U}_b$, and we define $\mathcal{U}_b^v := \mathcal{U}_b \cap \mathcal{U}^v$. The set of active slices at base station b, corresponding to those that have at least one user at b, is denoted by $\mathcal{V}_b$ (i.e., $|\mathcal{U}_b^v| > 0$ holds for $v \in \mathcal{V}_b$).

The goal in this paper is to develop a mechanism to allocate resources amongst slices. To that end, we let f_b^v denote the fraction of resources at base station b allocated to slice v. We adopt a generic formulation based on divisible resources that can be applied to a variety of technologies. The specific resource notion will depend on the underlying technology; for instance, in OFDM resources refer to physical resource blocks, in FDM to bandwidth and in TDM to the fraction of time.

The resources of a base station allocated to a slice are subdivided among the slice's users at the base stations, such that a user $u \in \mathcal{U}_b^v$ receives a fraction f_u of the resource, where $\sum_{u \in \mathcal{U}_b^v} f_u = f_b^v$. With such an allocation, user u achieves a service rate $r_u = f_u \cdot c_u$, where c_u is the user's achievable rate, defined as the rate that the user would see if she had the entire base station provisioned to herself. Note that c_u depends on the modulation and coding scheme selected for the user given the current radio conditions, which accounts for noise as well as the interference from the neighboring base stations. Following similar analyses in the literature (see e.g., [18]), we shall assume that c_u is fixed for each user at a given time.

The focus of this paper is on *slice-based resource allocation*: our problem is to decide which fraction of the overall resources we allocate to each slice (e.g., the number of resource blocks of each base station). In order to translate slice-based allocations to specific user-level allocations, the system will further need to decide (i) which specific resources will be assigned to each slice, and (ii) in turn, the assignment of slice resources to active users. This corresponds to a user-level scheduling problem which is not in the scope of this paper, but may impact the users' achievable rates c_u (this problem has been addressed, for instance, in [19]–[21]).

In line with standard network slicing frameworks [22], the approach studied in this paper can be flexibly combined with different algorithms for user-level allocations. The specific mechanism to assign resources to slices is the responsibility of the infrastructure provider, which may take into account, e.g., the latency requirements of the different slices. The sharing of the resources of a slice amongst its users is up to the slice, and different slices may run different scheduling algorithms depending on the requirements of their users. For instance, slices with throughput-driven services may opt for opportunistic schedulers [23]–[25] while other slices with latency requirements may opt for delay-sensitive schedulers [26].

Depending on its type of traffic, a slice may require different allocations. For instance, a URLLC slice with high reliability and/or low latency requirements may require a resource allocation much larger than its average load, to make make sure sufficient resources are available and/or delays are low. By contrast, a slice with eMBB traffic may not require guarantees at each individual base station, but may only need a certain average fraction of resources over time for its users (i.e., f_u).

B. GREET: Slice-based Resource Allocation

Below, we propose a slice-based resource allocation scheme that, on the one hand, ensures that each slice is guaranteed, *as needed*, a pre-agreed fraction of the resources at each individual base station, and, on the other hand, enables slices to contend for spare resources. Such division into guaranteed resources and extra ones is in line with current cloud models [27]–[29]. In order to regulate the resources to which a network slice is entitled, as well as the competition for the 'excess' resources, we rely on the different types of *shares* defined below. Such shares are specified in the slices' SLAs.

Definition 1. *For each slice v, we define the following pre-agreed* static *shares of the network resources.*

1) *We let the* ***guaranteed (resource) share*** *s_b^v denote the fraction of b's resources guaranteed to slice v, which must satisfy $\sum_{v \in \mathcal{V}} s_b^v \leq 1$ in order to avoid over-commitment.*
2) *We let e^v denote the* ***share of excess resources*** *which slice v can use to contend for the spare network resources.*
3) *We let s^v denote the slice v's* ***overall share****, given by $s^v = \sum_{b \in \mathcal{B}} s_b^v + e^v$.*

After being provisioned a fraction of network resource, each slice v has the option to divide its own share to its individual users. This can be done by designating a weight w_u for user

$u \in \mathcal{U}^v$. We let $\mathbf{w}^v = (w_u, u \in \mathcal{U}^v)$ denote the weight allocation of Slice v such that $\|\mathbf{w}^v\|_1 \leq s^v$. The set of feasible weight allocations is given by $\mathcal{W}^v := \{\mathbf{w}^v : \mathbf{w}^v \in \mathbb{R}_+^{|\mathcal{U}^v|}$ and $\sum_{u\in\mathcal{U}^v} w_u \leq s^v\}$. Then, we'll have $l_b^v = \sum_{u\in\mathcal{U}_b^v} w_u$ as the slice v's aggregate dynamic *local* bid to BS b, which is determined by its user distribution and must satisfy that $\sum_{b\in\mathcal{B}} l_b^v \leq s^v$. We further let $l_b := \sum_{v\in\mathcal{V}_b} l_b^v$ denote the overall bid at resource b and $l_b^{-v} := \sum_{v'\neq v} l_b^{v'}$ such bid excluding slice v. We define $\Delta_b^v := (l_b^v - s_b^v)_+$ as the excessive bid per BS of slice v. Then, our proposed resource allocation mechanism works as follows.

Definition 2. *(**GREET slice-based resource allocation**) We determine the fraction of each resource b allocated to slice v, $(f_b^v, v \in \mathcal{V}, b \in \mathcal{B})$, as follows. If $l_b \leq 1$, then*

$$f_b^v = \frac{l_b^v}{l_b}, \tag{1}$$

and otherwise

$$f_b^v = \begin{cases} l_b^v, & l_b^v < s_b^v, \\ s_b^v + \frac{\Delta_b^v}{\sum_{v'\in\mathcal{V}_b} \Delta_b^{v'}} \left(1 - \sum_{v'\in\mathcal{V}_b} \min\left(s_b^{v'}, l_b^{v'}\right)\right), & l_b^v \geq s_b^v. \end{cases} \tag{2}$$

The rationale underlying the above mechanism is as follows. If $l_b \leq 1$, then (1) ensures that each slice gets a fraction of resources f_b^v exceeding its local bid l_b^v at resource b. If $l_b > 1$, then (2) ensures that a slice whose local bid at b is less than its guaranteed share, i.e., $l_b^v \leq s_b^v$, receives exactly its local bid, and a slice with a local bid exceeding its guaranteed share, i.e., $l_b^v > s_b^v$, receives its guaranteed share s_b^v plus a fraction of the *extra resources* proportional to the excessive bid Δ_b^v. The extra resources here correspond to those not allocated based on guaranteed resource shares. As a slice can always choose a local-bid allocation at resource b, l_b^v, exceeding its guaranteed share, s_b^v, this ensures that, if it so wishes, a slice can always attain its guaranteed resource shares.

The above specifies the slice allocation per resource. Based on the w_u's, the slices then allocate base stations' resources to users in proportion to their weights, i.e., $f_u = \frac{w_u}{\sum_{u'\in\mathcal{U}_b^v} w_{u'}} f_b^v$, where f_u is the fraction of resources of base station b allocated to user $u \in \mathcal{U}_b^v$.

One can think of the above allocation in terms of market pricing schemes as follows. The share s^v can be understood the budget of player v and the local bid l_b^v as the bid that this player places on resource b. Then, the case where $l_b \leq 1$ corresponds to the well-known Fisher market [30], where the price of the resource is set equal to the aggregate bids from slices, making allocations proportional to the slices' bids. GREET deviates from this when $l_b \geq 1$ by modifying the 'pricing' as follows: for the first s_b^v bid of slice v on resource b, GREET sets the price to 1, to ensure that the slice budget suffices to buy the guaranteed resource shares. Beyond this, the remaining resources are priced higher, as driven by the corresponding slices' excess bids.

In summary, the proposed slice-based resource allocation scheme is geared at ensuring a slice will, if it wishes, be able to get its guaranteed resource shares, s_b^v, but it also gives a slice the flexibility to contend for excess resources, by shifting portions of its overall share s^v (both from the guaranteed and excess shares) across the network resources, to better meet its current users' requirements by aligning with its user traffic. Such a slice-based resource sharing model provides the benefit of protection guarantees as well as the flexibility to adapt to user demands.

III. Network Slicing Game Analysis

Under the GREET resource allocation scheme, each slice must choose how to subdivide its overall share amongst its users. Then, the network decides how to allocate base station resources to slices. This can be viewed as a *network slicing game* where, depending on the choices of the other slices, each slice chooses an allocation of local bid to base stations that maximizes its utility. In this section, we study the behavior of this game; we first provide a model for the utility of a slice and then analyze the resulting game.

A. Slice and Network Utilities

Note that the users' rate allocations, $(r_u : u \in \mathcal{U})$, can be expressed as a function of the overall slice weight assignments across the network, $\mathbf{w} = (w_u : u \in \mathcal{U})$. Indeed, the weights provide the local bid of each slice at each base station, which determine the resources of each slice, as well as the division of such resources across the slice's users at the base station. Accordingly, in the sequel we focus the game analysis on the weights and express the resulting user rates as $r_u(\mathbf{w})$.

We assume that each slice has a *private* utility function, denoted by U^v, that reflects the slice's preferences based on the needs of its users. We suppose the slice utility is simply a sum of its users individual utilities, U_u, i.e., $U^v(\mathbf{w}) = \sum_{u\in\mathcal{U}^v} U_u(r_u(\mathbf{w}))$.

Following standard utility functions [31] [32], we assume that for some applications, a user $u \in \mathcal{U}^v$ may require a guaranteed rate γ_u, hereafter referred to as the user's *minimum rate requirement*. We model the utility functions for rates above the minimum requirement as follows:

$$U_u(r_u(\mathbf{w})) = \begin{cases} \phi_u F_u(r_u(\mathbf{w}) - \gamma_u), & r_u(\mathbf{w}) > \gamma_u, \\ -\infty & \text{otherwise}, \end{cases}$$

where $F_u(\cdot)$ is the utility function associated with the user, and ϕ_u reflects the *relative priority* that slice v wishes to give user u, with $\phi_u \geq 0$ and $\sum_{u\in\mathcal{U}^v} \phi_u = 1$.

For $F_u(\cdot)$, we consider the following widely accepted family of functions, referred to as α-fair utility functions [33]:

$$F_u(x_u) = \begin{cases} \frac{(x_u)^{1-\alpha^v}}{(1-\alpha^v)}, & \alpha^v \neq 1 \\ \log(x_u), & \alpha^v = 1, \end{cases}$$

where the α^v 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 $\alpha^v = 0$ (maximum sum), $\alpha^v = 1$ (proportional fairness), $\alpha^v = 2$ (minimum potential delay fairness) and $\alpha^v \to \infty$ (max-min fairness).

Note that the above utility is flexible in that it allows slice utilities to capture users with different types of traffic:

- *Elastic traffic* ($\gamma_u = 0$ and $\phi_u > 0$): users with no minimum rate requirements and a utility that increases with the allocated rate, possibly with different levels of concavity given by α^v.
- *Inelastic traffic* ($\gamma_u > 0$ and $\phi_u = 0$): users that have a minimum rate requirement but do not see any utility improvement beyond this rate.

- *Rate-adaptive traffic* ($\gamma_u > 0$ and $\phi_u > 0$): users with a minimum rate requirement which see a utility improvement if they receive an additional rate allocation above the minimum.

Following [9], [10], [12]–[14], [34], we define the overall (network) utility as the sum of the individual slice utilities weighted by the respective overall shares,

$$U(\mathbf{w}) = \sum_{v \in \mathcal{V}} s^v U^v(\mathbf{w}), \tag{3}$$

and the social optimal weight allocation $\mathbf{w}^{\text{so}}$ as the allocation maximizing the overall utility $U(\mathbf{w})$, i.e.,

$$\mathbf{w}^{\text{so}} = \underset{\mathbf{w}}{\text{argmax}}\ U(\mathbf{w}). \tag{4}$$

B. Network Slicing Resource Allocation Game

Next we analyze the network slicing game resulting from the GREET resource allocation scheme and the above slice utility. We formally define the network slicing game as follows, where $\mathbf{w}^v$ denotes slice v users' weights.

Definition 3. ***(Network slicing game)*** *Suppose each slice v has access to the guaranteed shares and the local bid allocations of the other slices, i.e., $s_b^{v'}, l_b^{v'}, v' \in \mathcal{V} \setminus \{v\}, b \in \mathcal{B}$. In the network slicing game, slice v chooses its own user weight allocation $\mathbf{w}^v$ in its strategic space $\mathcal{W}^v$ so as to maximize its utility, given that the network uses a GREET slice-based resource allocation. This choice is known as slice v's Best Response (BR).*

In the sequel we consider scenarios where the guaranteed shares suffice to meet the minimal rate requirements of all users. The underlying assumption is that a slice would provision a sufficient shares and/or perform admission control so to limit the number of users. We state this formally as follows:

Assumption 1. ***(Well dimensioned shares)*** *We assume that the minimum rate requirements of the users of all slices can be met with the slices' guaranteed share at each base station. In particular, we assume that $\sum_{u \in \mathcal{U}_b^v} \underline{f}_u \leq s_b^v$ for all $v \in \mathcal{V}$ and $b \in \mathcal{B}$, where $\underline{f}_u = \frac{\gamma_u}{c_u}$ is the minimum fraction of resources required by user u to meet the minimum rate requirement γ_u. When this assumption holds, we say that the (guaranteed) shares of all slices are well dimensioned.*

The following lemma clarifies that, when the above assumption holds, a slice's best response is determined as the solution to a convex problem and meets the minimum rate requirements of all its users. Thus, this result guarantees that, as long as the shares of a slice are properly provisioned, the proposed scheme meets the slice's requirements.

Lemma 1. *When Assumption 1 holds, computing the Best Response under GREET-based resource allocation is a convex optimization problem. Furthermore, the minimum rate requirements of all the slice's users are satisfied by the Best Response.*

To characterize the system, it is desirable to determine the existence of a NE. The result below shows that, when the slice shares are well dimensioned, if we impose that weights have to be above some value δ (which can be arbitrarily small), the existence of a NE is guaranteed. However, if we do not impose such lower bound on the weights, a NE may not exist.

Theorem 1. *Suppose that Assumption 1 holds and that we constrain user weights to be positive, i.e., for all $u \in \mathcal{U}$ $w_u \geq \delta$ for some $\delta > 0$. Then, a NE exists. However, if we do not impose this constraint on the weights, an NE may not exist.*

Beyond the existence of equilibria, it is also desirable to have a dynamic behavior that leads to an equilibrium. Below, we analyze the Best Response Dynamics (BRD), where slices update their Best Response sequentially, one at a time, in a Round Robin manner. Ideally, we would like this process to converge after a sufficiently large number of rounds. However, the following result shows that this need not be the case.

Theorem 2. *Suppose that Assumption 1 holds and that we constrain user weights to be positive, i.e., for all $u \in \mathcal{U}$ $w_u \geq \delta$ for some $\delta > 0$. Then, even though a NE exists, the Best Response Dynamics may not converge.*

IV. GREET Slice Strategy

In addition to the equilibrium and convergence issues highlighted in Theorems 1 and 2, a drawback of the Best Response algorithm analyzed in Section III is its complexity. Indeed, to determine its best response, a slice needs to solve a convex optimization problem. This strays from the simple algorithms, both in terms of implementation and understanding, that get adopted in practice and tenants tend to prefer. In this section, we propose an alternative slice strategy to the best response, which we refer to as the *GREET share allocation policy*. This policy complements the resource allocation mechanism proposed in Section II, leading to the overall GREET framework consisting of two pieces: the resource allocation mechanism and the share allocation policy.

A. Algorithm definition and properties

The GREET resource allocation given in Section II depends on the bid that slices allocate at each base station. In the following, we propose the *GREET share allocation policy* to determine how each slice allocates its share across its users and resources. Our proposal works on the basis of user weights, corresponding to the share fraction allocated to individual users: we first determine the weights of all the users of the slice, and then compute the local bid by summing the weights of all the users at each base station, i.e., $l_b^v = \sum_{u \in \mathcal{U}_b^v} w_u$.

Under the proposed GREET share allocation, slices decide the weight allocations of their users based on two parameters: one that determines the minimum allocation of a user (γ_u) and another one that determines how extra resources should be prioritized (ϕ_u). A slice first assigns each user u the weight needed to meet its minimum rate requirement γ_u. Then, the slice allocates its remaining share amongst its users in proportion to their priority ϕ_u. The algorithm is formally defined below. Note that this algorithm does not require revealing each slices' local bids to the others but only aggregates, which discloses very limited information about slices' individual sub-shares and leads to low signaling overheads.

Definition 4. ***(GREET Share Allocation)*** *Suppose that each slice v has access to the following three aggregate values for each base station: l_b^{-v}, $\sum_{v' \in \mathcal{V}_b \setminus \{v\}} \Delta_b^v$ and $\sum_{v' \in \mathcal{V}_b \setminus \{v\}} \min(s_b^{v'}, l_b^{v'})$. Then, the GREET share allocation is given by the weight computation determined by Algorithm 1.*

Algorithm 1 GREET share allocation round for slice v

1: **for** user $u \in \mathcal{U}^v$ **do** set $\underline{f}_u \leftarrow \frac{\gamma_u}{c_u}$
2: **for** each base station $b \in \mathcal{B}$ **do** set $\underline{f}_b^v \leftarrow \sum_{u \in \mathcal{U}_b^v} \underline{f}_u$
3: **for** user $u \in \mathcal{U}^v$ **do**
4: **if** $l_b^{-v} + \underline{f}_b^v \leq 1$ **then** set $\underline{w}_u \leftarrow \frac{\underline{f}_u}{1-\underline{f}_b^v} l_b^{-v}$
5: **else**
6: **if** $s_b^v \geq \underline{f}_b^v$ **then** set $\underline{w}_u \leftarrow \underline{f}_u$
7: **else** set $\underline{w}_u \leftarrow$ expression given by (5)
8: **if** $\sum_{u \in \mathcal{U}^v} \underline{w}_u \leq s^v$ **then**
9: **for** user $u \in \mathcal{U}^v$ **do**
10: set $w_u \leftarrow \underline{w}_u + \phi_u \left(s^v - \sum_{u' \in \mathcal{U}^v} \underline{w}_{u'}\right)$
11: **else**
12: **while** $\sum_{u \in \mathcal{U}^b} w_u \leq s^v$ **do**
13: select users in order of increasing $\underline{w}_u$
14: set $w_u \leftarrow \underline{w}_u$

Algorithm 1 realizes the basic insight presented earlier. The slice, say v, first computes the minimum resource allocation required to satisfy the minimum rate requirement of each user, denoted by $\underline{f}_u$. These are then summed to obtain the minimum aggregate requirement at each base station, denoted by $\underline{f}_b^v$ (see Lines 1-2 of the algorithm).

Next, it computes the minimum weight for each user to meet the above requirements, denoted by $\underline{w}_u$. If $l_b^{-v} + \underline{f}_b^v \leq 1$, the GREET resource allocation is given by (1), and slice v's minimum local bid at base station b, $\underline{l}_b^v$, should satisfy $\frac{\underline{l}_b^v}{\underline{l}_b^v + l_b^{-v}} = \underline{f}_b^v$. Hence, the minimum share for user u at base station b is given by $\underline{w}_u = \frac{\underline{f}_u}{\underline{f}_b^v} \underline{l}_b^v = \frac{\underline{f}_u}{1-\underline{f}_b^v} l_b^{-v}$ (Line 4).

If $l_b^{-v} + \underline{f}_b^v > 1$, the GREET resource allocation is given by (2) and two cases need to be considered. In first case, where the minimum resource allocation satisfies $\underline{f}_b^v \leq s_b^v$, it suffices to set $\underline{l}_b^v = \underline{f}_b^v$ and $\underline{w}_u = \underline{f}_u$ and GREET resource allocation will make sure the requirement is met (Line 6). In the second case, where $\underline{f}_b^v > s_b^v$, in order to meet the minimal rate requirements under the GREET allocation given by (2), the minimum local bid allocation $\underline{l}_b^v$ must satisfy

$$s_b^v + \frac{(\underline{l}_b^v - s_b^v)\left(1 - s_b^v - \sum\limits_{v' \in \mathcal{V}_b \setminus \{v\}} \min\left(s_b^{v'}, l_b^{v'}\right)\right)}{\underline{l}_b^v - s_b^v + \sum\limits_{v' \in \mathcal{V}_b \setminus \{v\}} \Delta_b^{v'}} = \underline{f}_b^v.$$

Solving the above for $\underline{l}_b^v$ and allocating user weights in proportion to $\underline{f}_u$ gives the following minimum weights (Line 7):

$$\underline{w}_u = \frac{\underline{f}_u}{\underline{f}_b^v}\left(s_b^v + \frac{(\underline{f}_b^v - s_b^v)\sum_{v' \in \mathcal{V}_b \setminus \{v\}} \Delta_b^{v'}}{1 - \underline{f}_b^v - \sum_{v' \in \mathcal{V}_b \setminus \{v\}} \min(s_b^{v'}, l_b^{v'})}\right). \quad (5)$$

Once we have computed the minimum weight requirement for all users, we proceed as follows. If the slice's overall share s^v suffices to meet the requirements of all users, we divide the remaining share among the slice's users proportionally to their ϕ_u (Line 10). Otherwise, we assign weights such that we maximize the number of users that see their minimum rate requirement met, selecting users in order of increasing $\underline{w}_u$ and providing them with the minimum weight $\underline{w}_u$ (Lines 13-14).

The lemma below lends support to the GREET share allocation algorithm. It shows that, under some relevant scenarios, this algorithm captures the character of social optimal slice allocations. Furthermore, in a network with many slices where the overall share of an individual slice is very small in relative terms, GREET is a good approximation to a slice's best response, suggesting that a slice cannot gain (substantially) by deviating from GREET. This result thus confirms that, in addition to being simple, GREET provides close to optimal performance both at a global level (across the whole network) as well as locally (for each individual slice).

Lemma 2. *The weight allocations provided by the GREET share allocation policy satisfy the following properties:*

1) *Suppose that the users of all slices are elastic. Then, GREET provides all users with the same rate allocation as the social optimal weights, i.e., $r_u(\mathbf{w}^g) = r_u(\mathbf{w}^{so}), \forall u$, where $\mathbf{w}^{so}$ is the (not necessarily unique) social optimal weight allocation and $\mathbf{w}^g$ is the weight allocation under GREET.*
2) *Suppose that all the users of a slice are either elastic or inelastic and Assumption 1 holds. Further, suppose that $s^v / l_b^{-v} < \epsilon \ \forall b$. Then, the following holds for all u:*

$$\frac{w_u^{br,v}(\mathbf{w}^{-v})}{1+\epsilon} < w_u^{g,v}(\mathbf{w}^{-v}) < (1+\epsilon) w_u^{br,v}(\mathbf{w}^{-v}),$$

where $\mathbf{w}^{br,v}(\mathbf{w}^{-v})$ is the best response of slice v to the other slices' weights $\mathbf{w}^{-v}$ and $\mathbf{w}^{g,v}(\mathbf{w}^{-v})$ is slice v's response under GREET.

One of the main goals of the GREET resource allocation model proposed in Section II, in combination with the GREET share allocation policy proposed in this section, is to provide guarantees to different slices, so that they can in turn ensure that the minimum rate requirements of their users are met. The lemma below confirms that, as long as slices are well dimensioned, GREET will achieve this goal.

Lemma 3. *When Assumption 1 holds, the resource allocation resulting from combining the GREET resource allocation model with the GREET share allocation policy meets all users' minimum rate requirements.*

B. Convergence of the GREET algorithm

A key desirable property for a slice-based share allocation policy is convergence to an equilibrium. Applying a similar argument to that of Theorem 2, it can be shown that the GREET share allocation algorithm need not converge. However, below we will show sufficient conditions for convergence.

We let $\mathbf{w}(n)$ be the overall weight allocation for update round n. Our goal is to show that the weight sequence $\mathbf{w}(n)$ converges when $n \to \infty$. The following theorem provides a sufficient condition for geometric convergence to a unique equilibrium. According to the theorem, convergence is guaranteed as long as (i) slice shares are well dimensioned, and (ii) the guaranteed fraction of resources for a given slice at any base station is limited. The second condition essentially says there should be quite a bit of flexibility when managing guaranteed resources, leaving sufficient resources not committed to any slice. In practice, this may indeed make sense in networks supporting slices with elastic traffic (which need non-committed resources), inelastic traffic (which may require some safety margins), or combinations thereof.

Theorem 3. *Suppose that Assumption 1 holds and the maximum aggregate resource requirement per slice, $f_{\max}$, satisfies*

$$f_{\max} := \max_{v\in\mathcal{V}} \max_{b\in\mathcal{B}} \underline{f}_b^v < \frac{1}{2|\mathcal{V}|-1}. \tag{6}$$

Then, if slices perform GREET-based updates of their share allocations according to Algorithm 1, either in Round Robin manner or simultaneously, the sequence of weight vectors $(\mathbf{w}(n) : n \in \mathbb{N})$ converges to a unique fixed point, denoted by $\mathbf{w}^*$, irrespective of the initial share allocation $\mathbf{w}(0)$. Furthermore the convergence is geometric, i.e.,

$$\max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(n) - l_b^{v,*}| \leq \xi^n \max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(0) - l_b^{v,*}| \tag{7}$$

where $\xi := \frac{2(|\mathcal{V}|-1)f_{\max}}{1-f_{\max}}$ and $\mathbf{l}^{v,*}$ corresponds to slice v's per resource local bid at the fixed point $\mathbf{w}^*$. Note that, by (6)*, we have $\xi < 1$.*

This convergence result can be further generalized under the asynchronous update model in continuous time [35]. Specifically, without loss of generality, let n index the sequence of times $(t_n, n \in \mathbb{N})$ at which one or more slices update their share allocations and let $\mathcal{N}^v$ denote the subset of those indices where slice v performs an update. For $n \in \mathcal{N}^v$, slice v updates its share allocations based on possibly outdated weights for other slices, denoted by $(\mathbf{w}^{v'}(\tau_{v'}^v(n)) : v' \neq v)$, where $0 \leq \tau_{v'}^v(n) \leq n$ indexes the update associated with the most recent slice v' share weight updates available to slice v prior to the n^{th} update. As long as the updates are performed according to the assumption below, one can show that GREET converges under such asynchronous updates.

Assumption 2. ***(Asynchronous updates)*** *We assume that asynchronous updates are performed such that, for each slice $v \in \mathcal{V}$, the update sequence satisfies (i) $|\mathcal{N}^v| = \infty$, and (ii) for any subsequence $\{n_k\} \subset \mathcal{N}^v$ that tends to infinity, then $\lim_{k\to\infty} \tau_{v'}^v(n_k) = \infty,\ \forall v' \in \mathcal{V}$.*

Theorem 4. *Under Assumption 1, if slices perform GREET-based updates of their share allocations asynchronously but satisfying Assumption 2, and if* (6) *holds, then the sequence of weight updates $(\mathbf{w}(n) : n \in \mathbb{N})$ converges to a unique fixed point irrespective of the initial condition.*

While the above results provide some sufficient conditions for convergence, in the simulations performed in Section V we observed that, beyond these sufficient conditions, the algorithm always converges quite quickly under normal circumstances (within a few rounds). Based on this, we adopt an approach for the GREET share allocation algorithm where we let the weights to be updated by each slice for a number of rounds, and stop the algorithm if it has not converged upon reaching this number (which is set to 7 in our simulations).

V. Performance Evaluation

In this section we present a detailed performance evaluation of GREET versus two representative slice-based resource allocation approaches in the literature: one reservation- and the other share-based.

A. Mobile Network Simulation Setup

Simulation model: We simulate a dense 'small cell' wireless deployment following the IMT-Advanced evaluation guidelines [36]. The network consists of 19 base stations in a hexagonal cell layout with an inter-site distance of 20 meters and 3 sector antennas; thus, $\mathcal{B}$ corresponds to 57 sectors. Users associate to the sector offering the strongest SINR, where the downlink SINR between base station b and user u is modeled as in [37]: $\text{SINR}_{bu} = \frac{P_b G_{bu}}{\sum_{k\in\mathcal{B}\setminus\{b\}} P_k G_{ku} + \sigma^2}$, where, following [36], the noise σ^2 is set to -104dB, the transmit power P_b is equal to 41dB and the channel gain between BS sector b and user u, denoted by G_{bu}, accounts for path loss, shadowing, fast fading and antenna gain. The path loss is defined as $36.7\log_{10}(d_{bu}) + 22.7 + 26\log_{10}(f_c)$dB, where d_{bu} denotes the current distance in meters from the user u to sector b, and the carrier frequency f_c is equal to 2.5GHz. The antenna gain is set to 17 dBi, shadowing is updated every second and modeled by a log-normal distribution with standard deviation of 8dB [37]; and fast fading follows a Rayleigh distribution depending on the mobile's speed and the angle of incidence. The achievable rate c_u for user u at a given point in time is based on a discrete set of modulation and coding schemes (MCS), with the associated SINR thresholds given in [38]. This MCS value is selected based on the average $\overline{\text{SINR}}_{bu}$, where channel fast fading is averaged over a second. For user scheduling, we assume that resource blocks are assigned to users in a round-robin manner proportionally to the allocation determined by the resource allocation policy under consideration. For user mobility, we consider two different mobility patterns: Random Waypoint model (RWP) [39], yielding roughly uniform load distributions, and SLAW model [40], typically yielding clustered users and thus non-uniform load distributions.

Performance metrics: Recall that our primary goal is to give slices flexibility in meeting their users' minimum rate requirements while optimizing the overall network efficiency. To assess the effectiveness of GREET in achieving this goal, we focus on the following two metrics:

- Outage probability $P(\text{outage})$: this is the probability that a user does not meet its minimum rate requirement. In order for a slice to provide a reliable service, this probability should be kept below a certain threshold.
- Overall utility U: this is given by (3) and reflects the overall performance across all slices.

State-of-the-art approaches: In order to show the advantages of GREET, we will compare it to the following benchmarks:

- Reservation-based approach: with this approach, each slice v reserves a local share at each base station b, denoted by $\hat{s}_b^v$. The resources at each base station are then shared among the *active* slices (having at least one user) in proportion to the local shares $\hat{s}_b^v$. This is akin to setting weights for a Generalized Processor Sharing in a resource [41] and is in line with the spirit of reservation-based schemes in the literature [1]–[8].
- Share-based approach: with this approach, each slice gets a share $\tilde{s}^v$ of the overall resources, as in [9]–[15]. Specifically, resources at each base station are shared according to SCPF as proposed in [10], whereby each slice $v \in \mathcal{V}$ distributes its share $\tilde{s}^v$ equally amongst all its active users $u \in \mathcal{U}^v$, such that each user u gets a weight $\tilde{w}_u = \tilde{s}^v/|\mathcal{U}^v|$,

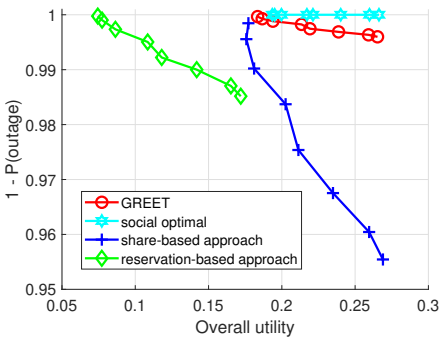

Fig. 1: Comparison of GREET against the benchmark approaches in terms of the overall Utility U and the outage probability $P(\text{outage})$.

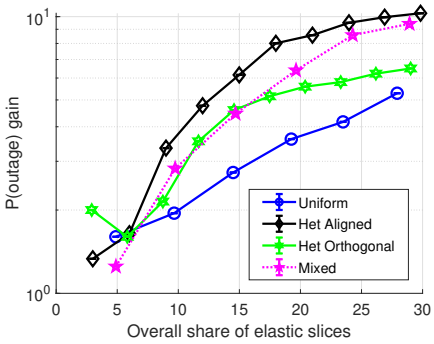

Fig. 2: Gain in $P(\text{outage})$ over share-based approach, measured as the ratio of $P(\text{outage})$ under the share-based approach over that under GREET.

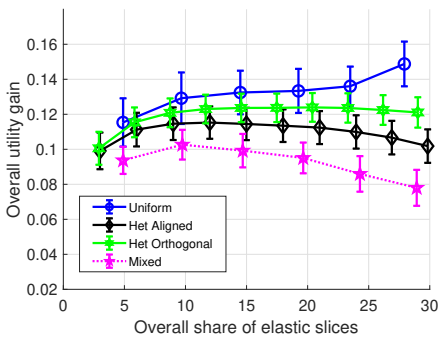

Fig. 3: Gain in utility over reservation-based approach, measured as the utility under GREET minus that under the reservation-based scheme.

and then, at each base station $b \in \mathcal{B}$ the resources are allocated in proportion to users' weights.

- Social optimal: this scheme corresponds to the social optimal weight allocation $\mathbf{w}^{\text{so}}$ given by (4) under GREET resource allocation.

In order to meet the desired performance targets, the shares employed in the above approaches are dimensioned as follows. We consider two types of slices: (i) those which provide their users with minimum rate requirements, which we refer to as *guaranteed service slices*, and (ii) those which do not provide minimum rate requirements, which we refer to as *elastic service slices*. In GREET, for guaranteed service slices, we define a maximum acceptable outage probability P_{max} and determine the necessary share at each base station, s_b^v, such that $P(\text{outage}) \leq P_{max}$, assuming that the number of users follow a Poisson distribution whose mean is obtained from the simulated user traces; for these slices, we set $e^v = 0$. For elastic service slices, we set $s_b^v = 0 \ \forall b$ and e^v to a value that determines the mean rate provided to elastic users. For the reservation-based approach, we set $\hat{s}_b^v = s_b^v$ for guaranteed service slices, to provide the same guarantees as GREET; for elastic service slices, we set $\hat{s}_b^v$ such that (i) their sum is equal to e^v, to provide the same total share as GREET, (ii) the sum of the $\hat{s}_b^v$'s at each base station does not exceed 1, to preserve the desired service guarantees, and (iii) they are as much balanced as possible across all base stations, within these two constraints. Finally, for the share-based approach we set $\tilde{s}^v = s^v$ for all slice types, i.e., the same shares as GREET.

B. Comparison with state-of-the-art benchmarks

Fig. 1 exhibits the performance of GREET versus the above benchmarks in terms of $P(\text{outage})$ and overall utility U for the following scenario: (i) we have two guaranteed service and two elastic service slices; (ii) the share of elastic service slices is increased within the range $s^v \in [2, 19]$; (iii) the minimum rate requirement for users on the guaranteed service slices is set to $\gamma_u = 0.2\,\text{Mbps}\ \forall u$; (iv) the shares of guaranteed service slices are dimensioned to satisfy an outage probability threshold P_{max} of 0.01; (v) for all slices, the priorities ϕ_u of all users are equal; and, (vi) the users of the elastic service slices follow the RWP model, leading to roughly uniform spatial loads, while the users of the guaranteed service slices have non-uniform loads as given by the SLAW model. Since user utilities are not defined below the minimum rate requirements, the computation of the overall utility only takes into account the users whose minimum rate requirements are satisfied under all schemes.

The results show that GREET outperforms both the share- and reservation-based approaches. While the share-based approach can flexibly shift resources across base stations, leading to a good overall utility, it is not able to sufficiently isolate slices from one another, resulting in large outage probabilities, $P(\text{outage})$, as the share of elastic service slices increase. By contrast, the reservation-based approach is effective in keeping $P(\text{outage})$ under control (albeit a bit above the threshold due to the approximation in the computation of s_b^v). However, since it relies on local decisions, it cannot globally optimize allocations and is penalized in terms of the overall utility. GREET achieves the best of both worlds: it meets the service requirements, keeping $P(\text{outage})$ well below the P_{max} threshold, while achieving a utility that matches that of the share-based approach. Moreover, it performs very close to the social optimal, albeit with somewhat larger $P(\text{outage})$ due to the fact that the social optimal imposes the minimum rate requirements as constraints, forcing each slice to help the others meeting their minimum rate requirements, while in GREET each slice behaves 'selfishly'.

C. Outage probability gains over the share-based scheme

One of the main observations of the experiment conducted above is that GREET provides substantial gains in terms of outage probability over the shared-based scheme. In order to obtain additional insights on these gains, we analyze them for a variety of scenarios comprising the following settings:

- *Uniform*: we have two guaranteed service slices and two elastic service slices; the users' mobility on all slices follow the RWP model and have the same priority ϕ_u.
- *Heterogeneous Aligned*: the users of all slices are distributed non-uniformly according to SLAW but they all follow the same distribution (i.e., has same hotspots).
- *Heterogeneous Orthogonal*: all slices are distributed according to SLAW model but each slice follows a different distribution (i.e., has different hotspots).
- *Mixed*: we have the same scenario as in Fig. 1, with the only difference that for one of the guaranteed service slices we have that all users are inelastic, i.e., the priority ϕ_u of all of them is set to 0.

For the above network configurations, we vary the share s^v of elastic service slices while keeping the shares for the guaranteed service slices fixed. Fig. 2 shows the ratio of the

$P(\text{outage})$ of the share-based approach over that of GREET as a function of the overall share of elastic slices, i.e., $\sum_{v \in \mathcal{V}_e} s^v$, where $\mathcal{V}_e$ is the set of elastic service slices. Results are given with 95% confidence intervals but they are so small that can barely be seen. We observe that GREET outperforms the share-based approach in all cases, providing $P(\text{outage})$ values up to one order of magnitude smaller. As expected, the gain in $P(\text{outage})$ grows as the the share of elastic service slices increases; indeed, as the share-based approach does not provide resource guarantees, it cannot control the outage probability of guaranteed service slices.

D. Utility gains over the reservation-based scheme

In order to gain additional insight on the utility gains over the reservation-based scheme, in Fig. 3 we analyze them for the scenarios introduced above. Results show that GREET consistently outperforms the reservation-based scheme across all approaches and share configurations, achieving similar gains in terms of overall utility in all cases. This confirms that, by providing the ability to dynamically adjust the overall resource allocation to the current user distribution across base stations, GREET can achieve significant utility gains over the reservation-based approach.

VI. Conclusions

GREET provides a flexible framework for managing heterogeneous performance requirements for network slices supporting dynamic user populations on a shared infrastructure. It is a practical approach that provides slices with sufficient resource guarantees to meet their requirements, and at the same time it allows them to unilaterally and dynamically customize their allocations to their current users' needs, thus achieving a good tradeoff between isolation and overall network efficiency. We view the GREET approach proposed here as a component of the overall solution to network slicing. Such a solution should include interfaces linking the resource allocation policies proposed here to lower level resource schedulers, which may possibly be opportunistic and delay-sensitive. Of particular interest will be the interfaces geared at supporting ultra-high reliability and with ultra-low latency services.

References

[1] S. Vassilaras and et. al., "The Algorithmic Aspects of Network Slicing," *IEEE Comm. Mag.*, vol. 55, no. 8, pp. 112–119, Aug. 2017.

[2] G. W. et al., "Resource Allocation for Network Slices in 5G with Network Resource Pricing," in *Proc. IEEE GLOBECOM*, Dec. 2017.

[3] V. Sciancalepore and et. al., "Mobile Traffic Forecasting for Maximizing 5G Network resource Utilization," in *Proc. IEEE INFOCOM*, May 2017.

[4] M. Leconte and et. al., "A resource allocation framework for network slicing," in *Proc. IEEE INFOCOM*, Apr. 2018.

[5] D. Bega and et. al., "Deepcog: Cognitive network management in sliced 5g networks with deep learning," in *Proc. IEEE INFOCOM*, Apr. 2019.

[6] ——, "Optimising 5G Infrastructure Markets: The Business of Network Slicing," in *Proc. IEEE INFOCOM*, May 2017.

[7] ——, "Mobile Traffic Forecasting for Maximizing 5G Network resource Utilization," in *IEEE Tran. Mobile Comp.*, to appear.

[8] X. Foukas and et. al., "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. ACM MOBICOM)*, Oct. 2017.

[9] J. Zheng and et. al., "Statistical multiplexing and traffic shaping games for network slicing," in *Proc. WiOPT*, Paris, France, May 2017.

[10] ——, "Statistical multiplexing and traffic shaping games for network slicing," *IEEE/ACM Trans. Networking*, vol. 26, no. 6, pp. 2528–2541, Dec 2018.

[11] P. Caballero and et. al., "Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads," *IEEE/ACM Trans. Networking*, vol. 25, no. 5, pp. 3044–3058, Oct 2017.

[12] ——, "Network Slicing Games: Enabling Customization in Multi-Tenant Networks," in *Proc. IEEE INFOCOM*, May 2017.

[13] ——, "Network Slicing Games: Enabling Customization in Multi-Tenant Mobile Networks," *IEEE/ACM Trans. Networking*, vol. 27, no. 2, pp. 662–675, Apr. 2017.

[14] ——, "Network Slicing for Guaranteed Rate Services: Admission Control and Resource Allocation Games," *IEEE Tran. Wireless Comm.*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018.

[15] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-complexity distributed radio access network slicing: Algorithms and experimental results," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2815–2828, Dec. 2018.

[16] 3GPP, "Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)," TS 28.628, Jun. 2013.

[17] J. Zheng, G. de Veciana, and A. Banchs, "Constrained network slicing games: Achieving service guarantees and network efficiency," 2020. [Online]. Available: http://arxiv.org/abs/2001.01402

[18] R. Mahindra, M. A. Khojastepour, H. Zhang, and S. Rangarajan, "Radio Access Network sharing in cellular networks," in *Proc. of IEEE ICNP*, Oct. 2013.

[19] S. D'Oro and et. al., "The Slice Is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems," in *Proc. IEEE INFOCOM*, Apr. 2019.

[20] S. Mandelli and et. al., "Satisfying Network Slicing Constraints via 5G MAC Scheduling," in *Proc. IEEE INFOCOM)*, Apr. 2019.

[21] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Comm. Mag.*, vol. 55, no. 6, pp. 102–108, Jun. 2017.

[22] M. Richart and et. al., "Resource slicing in virtual wireless networks: A survey," *IEEE Tran. Network Service Management*, vol. 13, no. 3, pp. 462–476, Sept 2016.

[23] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Comm. Surveys and Tutorials*, vol. 15, no. 4, pp. 1671–88, 2013.

[24] S. Borst, N. Hegde, and A. Proutiere, "Mobility-Driven Scheduling in Wireless Networks," in *Proc. IEEE INFOCOM*, Apr. 2009.

[25] H. J. Kushner and P. A. Whiting, "Convergence of Proportional-Fair Sharing Algorithms Under General Conditions," *IEEE Trans. Wireless Comm.*, vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

[26] M. Kalil, A. Shami, and A. Al-Dweik, "QoS-Aware Power-Efficient Scheduler for LTE Uplink," *IEEE Trans. Mobile Comp.*, vol. 14, no. 8, pp. 1672–1685, Aug. 2015.

[27] M. Mattess, C. Vecchiola, and R. Buyya, "Mobility-Driven Scheduling in Wireless Networks," in *Proc. IEEE HPCC*, Sep. 2010.

[28] Amazon, "EC2 Spot Instances," https://aws.amazon.com/ec2/spot/.

[29] Google Cloud, "Preemptible Virtual Machines," https://cloud.google.com/preemptible-vms/.

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

[31] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sep. 1995.

[32] 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.

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

[34] P. Caballero and et. al., "Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads," *IEEE/ACM Trans. Networking*, vol. 25, no. 5, pp. 3044–3058, Oct. 2017.

[35] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989.

[36] ITU-R, "Report ITU-R M.2135-1, Guidelines for evaluation of radio interface technologies for IMT-Advanced," Technical Report, Dec 2009.

[37] Q. Ye and et. al., "User Association for Load Balancing in Heterogeneous Cellular Networks," *IEEE Trans. Wireless Comm.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[38] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," TS 36.213, v12.5.0, Rel. 12, Mar. 2015.

[39] E. Hyytia, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 680–694, June 2006.

[40] K. Lee and et. al., "SLAW: self-similar least-action human walk," *IEEE/ACM Trans. Networking*, vol. 20, no. 2, pp. 515–529, Apr. 2012.

[41] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344–357, Jun. 2004.