Constrained Network Slicing Games: Achieving service guarantees and network efficiency

J. Zheng, A. Banchs *Senior Member, IEEE,* and G. de Veciana, *Fellow, IEEE,*

Abstract—**Network slicing is a key capability for next generation mobile networks. It enables infrastructure providers 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 ensure 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 future 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 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 tenants' 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 typically 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-grained share of the network resources which combined with a fine-grained 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 they 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*

J. Zheng and G. de Veciana are with the University of Texas at Austin. A. Banchs is with the University Carlos III of Madrid and with the Institute IMDEA Networks, Spain.

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 each individual network resource, 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 slice 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 and share-based approaches, providing service guarantees while maximizing the overall performance (Section V).

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., [17]–[19]), 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 (beyond the fraction of resources) will be assigned to each slice, and in turn, (ii) 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 [20]–[22]).

In line with standard network slicing frameworks [23], 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 [24]–[26] while other slices with latency requirements may opt for delay-sensitive schedulers [27].

Depending on its type of traffic, a slice may require different allocations. For instance, an Ultra-Reliable Low-Latency Communication (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 enhanced Mobile Broadband (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.

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 a division into guaranteed resources and extra ones is in line with current sharing models for cloud computing [28]–[30]. 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 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* ***excess share*** *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 a network resource, each slice v has the option to sub-divide its share amongst its 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 will have $l_b^v = \sum_{u \in \mathcal{U}_b^v} w_u$ as the slice v's *aggregate weight* to base station b, which is determined by its user weight 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 load* at base station b and $l_b^{-v} := \sum_{v' \neq v} l_b^{v'}$ the overall load excluding slice v. We define $\Delta_b^v := (l_b^v - s_b^v)_+$ as the *excess weight* at base station b of slice v. The 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

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

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 aggregate weight l_b^v at resource b. If $l_b > 1$, then (2) ensures that a slice whose aggregate weight at b is less than its guaranteed share, i.e., $l_b^v \leq s_b^v$, receives exactly its aggregate weight, and a slice with an aggregate weight 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 excess weight Δ_b^v. The extra resources here correspond to those not allocated based on guaranteed shares. A slice can always choose a weight allocation such that the aggregate weight at resource b, l_b^v, exceeds its guaranteed share, s_b^v, and thus this ensures that, if it so wishes, a slice can always attain its guaranteed 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 aggregate weight 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 [31], 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 base stations, to better meet the current requirements of the slice's users, by aligning the slice bids with the users' 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 aggregate weights to base stations that maximizes its utility. In this section, we study the behavior of this 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 slice's weight assignments across the network, $\mathbf{w} = (w_u : u \in \mathcal{U})$. Indeed, the weights determine the slice's resources at each base station, as well as the division of such resources across the slice's users within each 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 [32], [33], 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:

$$U_u(r_u(\mathbf{w})) = \begin{cases} \phi_u F_u(r_u(\mathbf{w}) - \gamma_u), & r_u(\mathbf{w}) > \gamma_u, \\ -\infty & \text{otherwise}, \end{cases} \tag{3}$$

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 [34]:

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

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], [35], we define the overall network utility as the sum of the individual slice utilities weighted by the respective overall shares,

$$U(\mathbf{w}) = \sum_{v\in\mathcal{V}} s^v U^v(\mathbf{w}), \tag{4}$$

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

$$\mathbf{w}^{\text{so}} = \underset{\mathbf{w}\geq 0}{\operatorname{argmax}} \ \{ \ U(\mathbf{w}) \ : \ \sum_{u\in\mathcal{U}^v} w_u \leq s^v, \ \ \forall v \in \mathcal{V} \} \tag{5}$$

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 aggregate weights 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 share and/or perform admission control so as to limit the number of users. We state this formally as follows:

Assumption 1. ***(Well dimensioned shares)*** *The slices' guaranteed shares are said to be well dimensioned if they are set so as meet or exceed the minimum rate requirements of their users at each base station. In particular, they are such that 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 shares of all slices are well dimensioned.*

A more restrictive assumption is that slices provision *exactly* the guaranteed share needed to meet its users' minimum rate requirements.

Assumption 2. ***(Perfectly dimensioned shares)*** *The slices' guaranteed shares are said to be perfectly dimensions if they are set so as to exactly cover the minimum rate requirements of their users at each base station, i.e., $\sum_{u\in\mathcal{U}_b^v} \underline{f}_u = s_b^v$ for all $v \in \mathcal{V}$ and $b \in \mathcal{B}$. When this assumption holds, we say that the shares of all slices are perfectly dimensioned.*

The following lemma ensures that under Assumption 1 a slice's best response is given by the solution to a convex problem and meets the minimum rate requirements of all its users. Thus, as long as a slice's guaranteed shares are well dimensioned, the proposed scheme will meet the slice's users requirements.

Lemma 1. *When Assumption 1 holds, a slice's Best Response under GREET-based resource allocation is the solution to a convex optimization problem, and the minimum rate requirements of all the slice's users will be satisfied.*

To characterize the system, it is desirable to determine the existence of a Nash Equilibrium (NE). The theorem below shows that, when the slices' shares are well dimensioned, with an additional constraint that weights be strictly positive, e.g., greater than δ (which can be arbitrarily small), the existence of a NE is guaranteed.

Theorem 1. *Suppose that Assumption 1 holds and that we constrain user weights to be strictly positive, i.e., for all $u \in \mathcal{U}$ $w_u \geq \delta$ for some $\delta > 0$. Then, a NE exists. However, without the strictly positive constraint on weights, a NE may not exist.*

When a NE exists, it is natural to ask whether the dynamics of slices' unilateral best responses to each others weight allocations would lead to such an equilibrium. Below, we consider 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.*

The following two theorems further characterize the NE allocations of the network slicing game relative to the socially optimal resource allocation and in terms of envy, respectively.

Theorem 3. *Consider a setting where all slices' users are elastic and have logarithmic utilities, i.e., $\alpha = 1$. Suppose also that a NE exists. Then, the overall utility associated with the socially optimal weight allocations $\mathbf{w}^{so}$ versus that resulting from the NE of the network slicing game under GREET-based resource allocation, $\mathbf{w}^{ne}$, satisfy*

$$U(\mathbf{w}^{so}) - U(\mathbf{w}^{ne}) \leq \log(e) \sum_{v \in \mathcal{V}} s^v$$

Furthermore, there exists a game instance for which this bound is tight.

Theorem 4. *Consider a setting where slices satisfy Assumption 2 and a NE exists. Further, suppose that two slices v and $\tilde{v}$ have the same guaranteed and excess shares and that slice v has users with logarithmic utilities, i.e., $\alpha^v = 1$. Let $\mathbf{r}^{ne,v}$ denote the rate allocations to slice v's users under the NE. Suppose slice v and $\tilde{v}$ exchange the overall allocations they get at the NE and let $\tilde{\mathbf{r}}^v$ denote the rate allocations to users of slice v maximizing slice v's utility after such an exchange. Let us define the envy that slice v has for $\tilde{v}$'s allocation at the NE as*

$$E^{ne}(v, \tilde{v}) \doteq U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ne,v})$$

Then, the following is satisfied: $E^{ne}(v, \tilde{v}) \leq 0.060$. Furthermore, there is a game instance where $E^{ne}(v, \tilde{v}) \geq 0.041$.

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 weight 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 weight allocation policy.

A. Algorithm definition and properties

The GREET resource allocation given in Section II depends on the aggregate weight that slices allocate at each base station. In the following, we propose the *GREET weight allocation policy* to determine how each slice allocates its weights across its users and base stations. We first determine the weights of all the users of the slice, and then compute the aggregate weights 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 weight 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. Note that this algorithm does not require revealing each slices' aggregate weights to the others but only the base stations' overall loads, which discloses very limited information about slices' individual weights and leads to low signaling overheads. The algorithm is formally defined as follows.

Algorithm 1 GREET weight 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 (6)
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 $w_u \leftarrow \underline{w}_u$
```

Definition 4. ***(GREET Weight 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 weight allocation is given by the weight computation determined by Algorithm 1.*

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 aggregate weight 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 weight 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 the 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 aggregate weight $\underline{l}_b^v$ 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):

$$\underline{w}_u = \frac{\underline{f}_u}{\underline{f}_b^v}\left(s_b^v + \frac{(\underline{f}_b^v - s_b^v)\sum_{v'\in\mathcal{V}_b\setminus\{v\}}\Delta_b^{v'}}{1-\underline{f}_b^v-\sum_{v'\in\mathcal{V}_b\setminus\{v\}}\min(s_b^{v'}, l_b^{v'})}\right). \quad (6)$$

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).

Theorem 5 lends support to the GREET weight 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, Theorem 6 shows that GREET is a good approximation to a slice's best response, suggesting that a slice cannot gain (substantially) by deviating from GREET. These results thus confirm 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).

Theorem 5. *Suppose that all users are elastic and user utilities are logarithmic, i.e., $\alpha = 1$. Suppose GREET weight allocations converge to an equilibrium, which we denote by GREET equilibrium (GE). Then, GREET provides all users with the same rate allocation as that resulting from the socially optimal weights, i.e., $r_u(\mathbf{w}^{ge}) = r_u(\mathbf{w}^{so}), \forall u$, where $\mathbf{w}^{so}$ is the (not necessarily unique) socially optimal weight allocation and $\mathbf{w}^{ge}$ is the weight allocation under GREET equilibrium.*

Theorem 6. *Suppose that all the users of a slice are elastic, user utilities are logarithmic (i.e., $\alpha = 1$) and $s^v/l_b^{-v} < \delta \ \forall b$. Then, the following holds for all users u on slice v:*

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

where $\mathbf{w}^{br,v}(\mathbf{w}^{-v}) = (w_u^{br}(\mathbf{w}^{-v}) : u \in \mathcal{U}^v)$ is the best response of slice v to the other slices' weights $\mathbf{w}^{-v}$ and similarly $\mathbf{w}^{g,v}(\mathbf{w}^{-v})$ is slice v's response under GREET.

Further, suppose that GREET converges to an equilibrium. Then the resulting allocation is an ε-equilibrium with $\varepsilon = \log(1+\delta)$.

The following results shows that, in contrast to the NE allocations analyzed in Section III, the GREET allocations are envy-free.

Theorem 7. *Consider a setting where slices satisfy Assumption 2 and slices' GREET weight allocations converge to a GREET Equilibrium (GE). Suppose two slices v and $\tilde{v}$ have the same guaranteed and excess shares, and that slice v has users with logarithmic utilities, i.e., $\alpha = 1$. Let $\mathbf{r}^{ge,v}$ denote the rate allocations to slice v's users under at the GE and $\tilde{\mathbf{r}}^v$ their rate allocations after slices v and $\tilde{v}$ exchange their overall allocations at the equilibrium. Then the envy that slice v has for $\tilde{v}$'s allocation at the GE satisifes*

$$E^{ge}(v,\tilde{v}) \doteq U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ge,v}) \leq 0.$$

One of the main goals of the GREET resource allocation model proposed in Section II, in combination with the GREET weight 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 2. *When Assumption 1 holds, the resource allocation resulting from combining the GREET resource allocation model with the GREET weight allocation policy meets all users' minimum rate requirements.*

B. Convergence of the algorithm

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

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 $n \rightarrow \infty$. 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 that there should be quite a bit of flexibility when managing guaranteed resources, leaving sufficient resources not committed to any slice. In practice, this may be appropriate in networks supporting slices with elastic traffic (which need non-committed resources), inelastic traffic (which may require some safety margins), or combinations thereof.

Theorem 8. *Suppose that Assumption 1 holds and the maximum aggregate resource requirement per slice, $f_{\max}$, satisfies*

$$f_{\max} := \max_{v\in\mathcal{V}}\max_{b\in\mathcal{B}} \underline{f}_b^v < \frac{1}{2|\mathcal{V}|-1}. \quad (7)$$

Then, if slices perform GREET-based updates of their weight 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 weight allocation $\mathbf{w}(0)$. Furthermore, the convergence is geometric, i.e.,

$$\max_{v\in\mathcal{V}}\sum_{b\in\mathcal{B}}|l_b^v(n) - l_b^{v,*}| \leq \xi^n \max_{v\in\mathcal{V}}\sum_{b\in\mathcal{B}}|l_b^v(0) - l_b^{v,*}| \quad (8)$$

where $\xi := \frac{2(|\mathcal{V}|-1)f_{\max}}{1-f_{\max}}$ and $\mathbf{l}^{v,*}$ corresponds to slice v's aggregate weights at the fixed point $\mathbf{w}^*$. Note that (7) imposes $\xi < 1$.

This convergence result can be further generalized under the asynchronous update model in continuous time [36]. 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 weight 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 weights 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' 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 3. ***(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 9. *Under Assumption 1, if slices perform GREET-based updates of their weight allocations asynchronously but satisfying Assumption 3, and if (7) 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 we observed that, beyond these sufficient conditions, the algorithm always converges quite quickly under normal circumstances (within a few rounds). To show this, we run GREET over two artificial network settings with different user distributions and minimal rate requirements leading to different ξ. The results are Illsutrated in Fig. 1, where the 'theoretical upperbound' is the distance computed as the R.H.S. of (8). We observe that the actual convergence of GREET is geometric, but with a rate significantly greater than the theoretical bound. Furthermore, even with a $\xi > 1$, GREET still converges in a geometric manner, even though the theoretical results do not guarantee convergence in this case. Based on this, we adopt an approach for the GREET weight allocation algorithm where we let the weights 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 [37]. 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 [38]: $\text{SINR}_{bu} = \frac{P_b G_{bu}}{\sum_{k\in\mathcal{B}\setminus\{b\}} P_k G_{ku} + \sigma^2}$, where, following [37], the noise σ^2 is set to -104dB, the transmit power P_b is equal to 41dB and the channel gain between sector b and user u, denoted by G_{bu}, accounts for path loss, shadowing, fast fading and antenna gain. The path loss is given by $36.7\log_{10}(d_{bu}) + 22.7 + 26\log_{10}(f_c)$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 [38]; 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 [39]. 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) [40], yielding roughly uniform load distributions, and SLAW model [41], typically yielding clustered users and thus non-uniform load distributions.

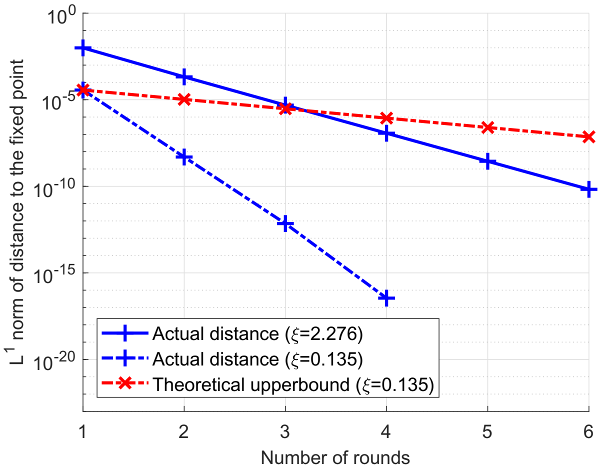

Fig. 1: Actual L^1 norm of the distance to the fixed point under GREET vs. the theoretical upper bound provided by Theorem 8.

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 (4) 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 $\hat{s}^v_b$.

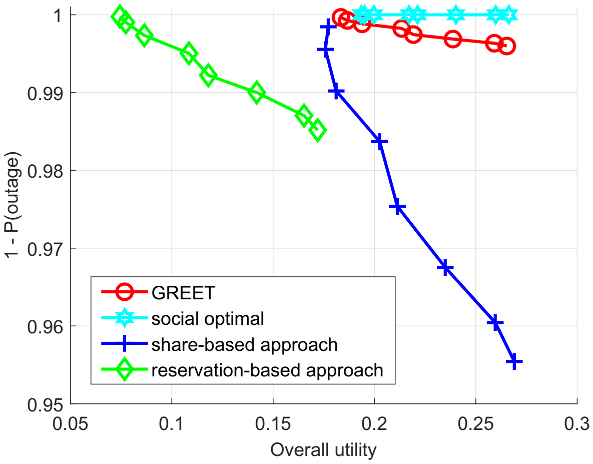

Fig. 2: Comparison of GREET against the benchmark approaches in terms of the overall Utility U and the outage probability $P(\text{outage})$.

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 [42] 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 the scheme proposed in [10], where 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 $\tilde{w}_u = \tilde{s}^v/|\mathcal{U}^v|$, 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 (5) 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. 2 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

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., slices have the same hotspots).
- *Heterogeneous Orthogonal*: all slices are distributed according to SLAW model but each slice follows a different distribution (i.e., slices have different hotspots).

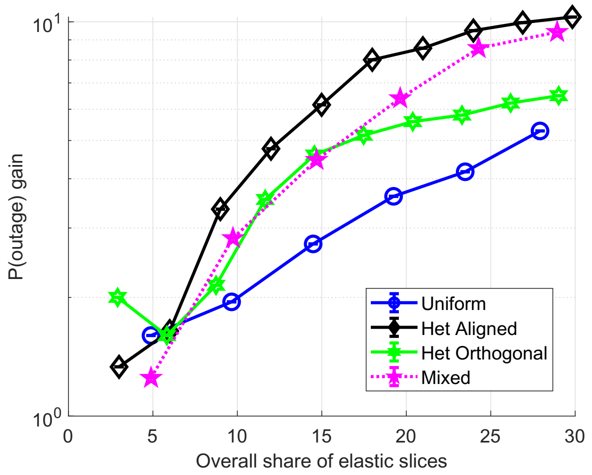

Fig. 3: Gain in $P(\text{outage})$ over the share-based approach, measured as the ratio of $P(\text{outage})$ under the share-based approach over that under GREET.

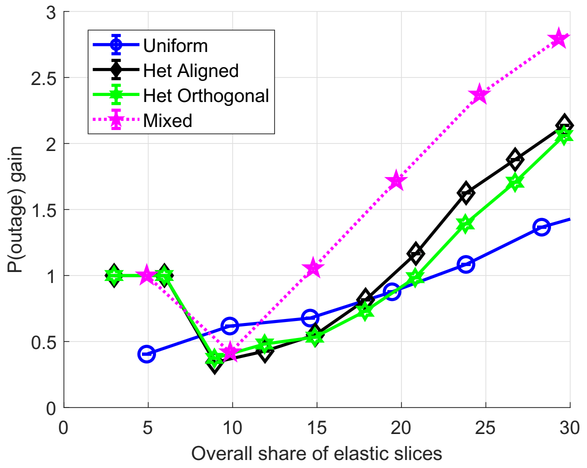

Fig. 4: Gain in $P(\text{outage})$ over the reservation-based approach, measured as the ratio of $P(\text{outage})$ under the reservation-based approach over that under GREET.

- *Mixed*: we have the same scenario as in Fig. 2, 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. 3 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.

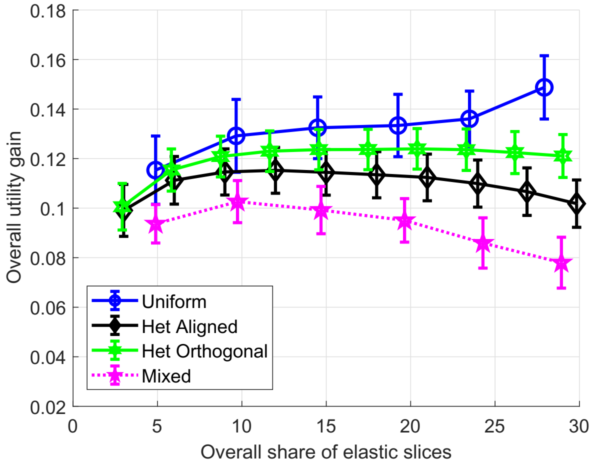

Fig. 5: Gain in utility over the reservation-based appraoch, measured as the utility under GREET minus that under the reservation-based approach.

Fig. 4 further compares the performance of GREET against the reservation-based approach in terms of $P(\text{outage})$, by showing the ratio of the $P(\text{outage})$ of the reservation-based approach over that of GREET. As expected, GREET offers a comparable performance to that of the reservation-based approach, since both approaches have been dimensioned to achieve a very small $P(\text{outage})$. In particular, when overall elastic slice share is between 5 to 15, the reservation-based approach beats GREET by a factor approximately 2, which translates to a margin in $P(\text{outage})$ of the order of 0.0001 in our simulation. Meanwhile, when the elastic slice share ramps up to over 20, GREET starts to offer lower $P(\text{outage})$ than the reservation-based approach in all 4 network configurations. This is because the mismatch between RWP/SLAW model and Poisson distribution assumed in dimensioning the share allocation of the reservation-based approach becomes more significant. Note that while the differences in relative terms are not necessarily negligible, since in all cases the $P(\text{outage})$ values are very low, the differences in absolute terms are indeed very small.

D. Utility gains

In order to gain additional insight on the utility gains over the reservation-based scheme, in Fig. 5 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.

Fig. 6 further compares the utility under GREET and that under the share-based approach. We observe that the difference in utilities are very small, which means that the share-based approach offers a very similar performance to GREET in terms of utility. A closer look reveals that, although the share-based approach can adapt to dynamic user distribution very well, GREET still con-

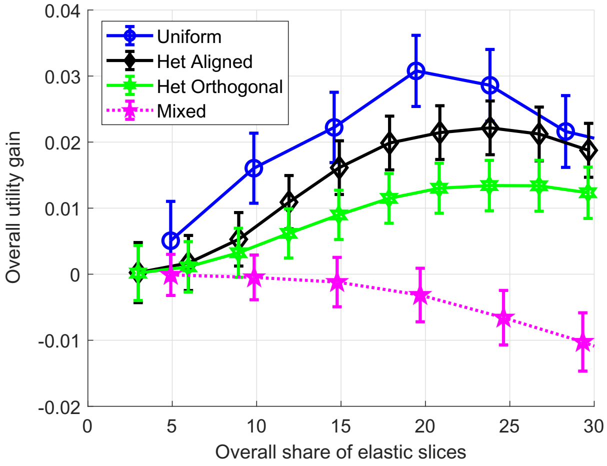

Fig. 6: Gain in utility over the share-based approach, measured as the utility under GREET minus that under the share-based approach.

sistently achieves better utility under all network configurations except for the *Mixed* scenario.

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.

Appendix: Proofs of the Theorems

Proof of Lemma 1

We first show that there exists a weight setting that meets the minimum rate requirements of all users. As long as a fraction of base station b equal to s_b^v is sufficient to meet the user minimum rate requirements, by applying an aggregate weight equal to s_b^v in the resource, the tenant is guaranteed to get this fraction of resources. As this can be applied to all resources, the minimum rate guarantees can be met for all the users of the tenant.

The optimization problem is given by the maximization of the sum of user utilities. This is a concave function on the weights as long as the individual user utilities are concave. As long as the minimum rate requirements are satisfied, individual user utilities are concave, as they are increasing concave function of a concave function (see [43]). The set of feasible weights need to satisfy $\sum_{u\in\mathcal{U}^v} w_u \leq s^v$ and $w_u \geq 0, \forall u \in \mathcal{U}$, and need to be such that the minimum rate requirements are satisfied. The latter imposes $w_u / \sum_{u\in\mathcal{U}_b} w_u \geq \gamma_u$ which yields $w_u - \gamma_u \sum_{u\in\mathcal{U}_b} w_u \geq 0,\ \forall u$. As a result, the set of flexible weights is convex.

Proof of Theorem 1

We first prove the existence of a NE when $w_u \geq \delta$. Let $\mathcal{W}$ be the convex and compact set of feasible weights $\mathbf{w}$ satisfying (i) $w_u \geq \delta, \forall u$, and (ii) $\sum_{u\in\mathcal{U}^v} w_u = s^v, \forall v \in \mathcal{V}$ and let us consider the mapping $\mathbf{w} \rightarrow \tilde{\mathbf{w}} = \Gamma(\mathbf{w})$, where $\tilde{\mathbf{w}}^v$ is the best response of slice v to $\mathbf{w}^{-v}$. 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 Lemma 1. According to that lemma, the best response of a slice to $\mathbf{w}^{-v}$ is the allocation $\tilde{\mathbf{w}}^v$ corresponding to the solution to a convex optimization problem. This implies that $\tilde{\mathbf{w}}^v$ exists and is a convex set, according to the properties of convex optimization problems. Hence, $\tilde{\mathbf{w}}$ exists and is a convex set as well.

Condition iii) is shown by proving that $\tilde{\mathbf{w}}^v$ is a continuous function of $\mathbf{w}^{-v}$ for all slices. Consider the set of base stations for which: (i) $l_b \leq 1$, (ii) $l_b > 1$ and $f_b^v \leq s_b^v$, and (iii) $l_b > 1$ and $f_b^v > s_b^v$. As long as these sets do not change, $\tilde{\mathbf{w}}^v$ can be expressed as a continuously differentiable function of $\{\tilde{\mathbf{w}}^v, \mathbf{w}^{-v}\}$, and it follows from the implicit function theorem that $\tilde{\mathbf{w}}^v$ is a continuous function of $\mathbf{w}^{-v}$. When some base station moves from one set to the other, such base station satisfies the equations corresponding to both cases, 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.

We now prove that $w_u \geq \delta$ does not hold, we may not have a NE. Consider a scenario with two slices, 1 and 2, and two base stations, a and b. Each slice has a user in each base station such that $\gamma_{1a} = \gamma_{2a} = 1/4$ and $\gamma_{1b} = \gamma_{2b} = 0$. Furthermore, we have $\phi_{1a} = \phi_{2a} = 0$, $\phi_{1b} = \phi_{2b} = 1$, $s_a^1 = s_a^2 = 1$, $s_b^1 = s_b^2 = 0$ and $e^1 = e^2 = 0$. In the best response, it holds $w_{1a} = w_{1b}/3$ and $w_{1b} = w_{1a}/3$, which implies that there exists no NE.

Proof of Theorem 2

Let us consider a scenario with three slices, denoted by Slices 1, 2 and 3, and three base stations, denoted by Base Station (BS) a, b, and c, respectively. Slice 1 has two users, one at BS a, another at BS b, denoted by 1a and 1b, respectively. Slice 2 has two users, one at BS b, another at BS c, denoted by 2b and 2c. Also, Slice 3 has two users at BS a and c, respectively, denoted by 3a and 3c. The share allocation is $s^1 = s^2 = s^3 = 3/4 + \epsilon$ for some $\delta < \epsilon < 1/4$, $s_a^1 = s_b^2 = s_c^3 = 3/4$, $\gamma_{1a} = \gamma_{2b} = \gamma_{3c} = 3/4$, $\gamma_{1b} = \gamma_{2c} = \gamma_{3a} = 0$, $\phi_{1a} = \phi_{2b} = \phi_{3c} = 0$ and $\phi_{1b} = \phi_{2c} = \phi_{3a} = 1$.

It can be seen that a NE in the above scenario is given by $w_{1a} = w_{2b} = w_{3c} = 9/16 + 3\epsilon/4$ and $w_{1b} = w_{2c} = w_{3a} = 3/16 + \epsilon/4$.

Let us start with $w_{3a} > 1/4$ and and apply the best response starting with slice 1 followed by 2 and 3. Slice 1 takes $w_{1a} = 3/4$ and $w_{1b} = \epsilon$. In turn, slice 2 selects $w_{2b} = 3\epsilon$ and $w_{2c} = 3/4 -$

$2\epsilon > 1/4$. This yields $w_{3c} = 3/4$ and $w_{3a} = \epsilon$. We thus enter an endless cycle where w_{1a}, w_{2b} and w_{3c} alternate the values of $3/4$ with 3ϵ.

Proof of Theorem 3

This theorem follows from Theorem 4 in [13]. In particular when the network has only elastic users with logarithmic utilities, the GREET slice-based resource allocation proposed in this paper coincides with the resource allocation mechanism proposed in [13]. Thus by Theorem 4 of [13] we have that

$$U(\mathbf{w}^{so}) - U(\mathbf{w}^{ne}) \leq \log(e) \sum_{v \in \mathcal{V}} s^v. \tag{9}$$

Note that in [13] the weights are normalized so $\sum_{v \in \mathcal{V}} s^v = 1$ but such a normalization is not required under GREET and hence we have the above result instead of $U(\mathbf{w}^{so}) - U(\mathbf{w}^{ne}) \leq \log(e)$ as in [13].

Proof of Theorem 4

This theorem is similar to Theorem 5 in [13], except that we need to address a setting with elastic, inelastic and rate adaptive users with guaranteed shares and a GREET-based resource allocation. Under Assumption 2 and considering the excess rate allocation to each user, i.e., those beyond the required guarantees, we will show that our problem reduces to that in Theorem 5.

Let $\mathbf{w}^{ne}$ denote weight allocation at the NE and l_b^{ne} and $l_b^{ne,v}$ the overall load and the aggregate weight of slice v at base station b. Consider a user u of slice v in base station b. The rate r_u^{ne} of user u at the NE under a GREET-based resource allocation satisfies one of the two following cases.

First, if the load at base station b satisfies $l_b^{ne} \leq 1$ the rate of a user u at this base station is given by

$$r_u^{ne} = \frac{w_u^{ne}}{l_b^{ne}} c_u.$$

Note from (3) that slice utility depends on $r_u^{ne} - \gamma_u$. We define the excess rate and the excess weight allocation to user u as $q_u^{ne} \doteq r_u^{ne} - \gamma_u$ and $m_u^{ne} \doteq w_u^{ne} - l_b^{ne} \gamma_u / c_u$, respectively, where $l_b^{ne} \gamma_u / c_u$ is the weight that user u needs in order to meet its minimal rate requirement γ_u when the load at b is l_b^{ne}. With this notation, we have that

$$q_u^{ne} = \frac{m_u^{ne}}{l_b^{ne}} c_u.$$

Second, if the load at base station b satisfies $l_b^{ne} > 1$, then the guaranteed share s_b^v is exactly what is needed to meet its users rate requirements at b (given that v is perfectly dimensioned). Thus, the excess rate of user u corresponds to the second term in (2), i.e.,

$$q_u^{ne} = \frac{m_u^{ne}}{\sum_{v' \in \mathcal{V}_b} \Delta_b^{ne,v'}} \left(1 - \sum_{v' \in \mathcal{V}_b} \min[s_b^{v'}, l_b^{ne,v'}] \right) c_u$$

where the excess weight of user u is now given by $m_u^{ne} = w_u - \gamma_u / c_u$. If we define $\hat{c}_u \doteq c_u(1 - \sum_{v' \in \mathcal{V}_b} \min[s_b^{v'}, l_b^{ne,v'}])$ and $\hat{l}_b^{ne} \doteq \sum_{v' \in \mathcal{V}_b} \Delta_b^{ne,v'}$, the above can be rewritten as

$$q_u^{ne} = \frac{m_u^{ne}}{\hat{l}_b^{ne}} \hat{c}_u$$

Putting together the above two cases, the excess weights for users u on slice v are given by $m_u^{ne} = w_u - \min[l_b^{ne} \gamma_u / c_u, \gamma_u / c_u]$ and satisfy:

$$\sum_{u \in \mathcal{U}^v} m_u^{ne} = s^v - \sum_{u \in \mathcal{U}^v} \min\left[\frac{l_b^{ne} \gamma_u}{c_u}, \frac{\gamma_u}{c_u} \right].$$

Now recall that in this theorem we consider two slices v and $\tilde{v}$ which have the same guaranteed and overall network shares and which exchange the resource allocations they achieved under the NE. We shall denote the rate and weight that a user u on slice v would receive under such an exchange by $\tilde{r}_u$ and $\tilde{w}_u$, respectively. We shall only consider the cases where $\tilde{w}_u \geq \min[l_b^{ne} \gamma_u / c_u, \gamma_u / c_u]$, as otherwise slice v would not meet its users' rate requirements after the exchange with $\tilde{v}$.

Note that the weight allocations after the exchange, $\tilde{\mathbf{w}}$, see base stations loads $\mathbf{l}^{ne}$, so we can express the rates that the users of slice v after the exchange with $\tilde{v}$ as follows. First, if $l_b^{ne} \leq 1$, we have

$$\tilde{q}_u = \frac{\tilde{m}_u}{l_b^{ne}} c_u$$

where $\tilde{q}_u$ and $\tilde{m}_u$ are the excess rate and weight of user u after v and $\tilde{v}$ exchange resource allocations. Second, when $l_b^{ne} > 1$, we have that

$$\tilde{q}_u = \frac{\tilde{m}_u}{\hat{l}_b^{ne}} \hat{c}_u,$$

where the $\tilde{m}_u$ for users on slice v must satisfy

$$\sum_{u \in \mathcal{U}^v} \tilde{m}_u = s^v - \sum_{u \in \mathcal{U}^v} \min\left[\frac{l_b^{ne} \gamma_u}{c_u}, \frac{\gamma_u}{c_u} \right]. \tag{10}$$

Note that expressions for the excess rates and constraints on the excess weights $\mathbf{m}^{ne}$ and $\tilde{\mathbf{m}}$ are the same as if all the users of slice v where elastic. Indeed, at a base station where when $l_b^{ne} \leq 1$, the resource allocation criterion akin to one where all users were elastic users. Similarly, when $l_b^{ne} > 1$, we obtain the same expressions by taking $\hat{c}_u$ and $\hat{l}_b^{ne}$ instead of c_u and l_b^{ne}. The constraints on $\sum_{u \in \mathcal{U}^v} m_u^{ne}$ and $\sum_{u \in \mathcal{U}^v} \tilde{m}_u$ are also equivalent to the case with elastic users, substituting the overall share s_v by the following expression: $s^v - \sum_{u \in \mathcal{U}^v} \min[l_b \gamma_u / c_u, \gamma_u / c_u]$.

Thus, by considering excess rates and weights, this makes the problem equivalent to the one where all users are elastic. Further, since slice v users are assumed to have logarithmic utilities, the envy associated with slice v and $\tilde{v}$ resource exchange at the NE can be established via the result in Theorem 5 of [13], which proves this theorem.

Proof of Theorem 5

The utility of the network depends on the users' rates r_u. Since there is a direct mapping between the fraction of resources assigned to each user and its rate, we can express utility as a function of the fractions f_u $\forall u$, i.e., $U(\mathbf{f})$. When there is only elastic traffic in the network, the total utility is given by

$$U(\mathbf{f}) = \sum_{u \in \mathcal{U}} s^{v(u)} \phi_u \log(f_u c_u)$$

subject to

$$\sum_{u \in \mathcal{U}_b} f_u = 1 \ \forall b,$$

where $v(u)$ is the slice user u belongs to.

The problem of maximizing the total utility subject to the above constraint is solved by Lemma 5.1 of [44], leading to

$$f_u = \frac{s^{v(u)} \phi_u}{\sum_{u' \in \mathcal{U}_{b(u)}} s^{v(u')} \phi_{u'}} \tag{11}$$

where $b(u)$ is the base station user u is associated with.

The above optimization did not impose the constraint on the weights of a slice, $\sum_{u \in \mathcal{U}^v} w_u \leq s^v$, and hence in principle represents an upper bound on the total utility of the socially optimal allocation. However, the weights resulting from the optimization satisfy this constraint, which means that the allocation of (11) is the socially optimal allocation.

Note that the allocation of (11) coincides with the allocation resulting from GREET. Indeed, when all users are elastic GREET simply sets the share fractions proportionally to the ϕ_u values, while forcing that all share fractions add up to the slice's share s^v.

Proof of Theorem 6

Let us consider the best response and the GREET response of slice v when other slices and associated users choose a weight allocation $\mathbf{w}^{-v}$ leading to per-base station overall load vector $\mathbf{l}^{-v}$. The best response to $\mathbf{l}^{-v}$ is the weight allocation that maximizes slice V's utility, and the GREET response is the result of applying the GREET weight allocation algorithm.

When there is only elastic traffic in the network, the weight allocation to a user u on slice v under the best response to $\mathbf{l}^{-v}$ is given by [13],

$$w_u^{br}(\mathbf{l}^{-v}) = s^v \frac{\phi_u \frac{l^{-v}_{b(u)}}{l^{br,v}_{b(u)}(\mathbf{l}^{-v}) + l^{-v}_{b(u)}}}{\sum_{u' \in \mathcal{U}^v} \phi_{u'} \frac{l^{-v}_{b(u')}}{l^{br,v}_{b(u')}(\mathbf{l}^{-v}) + l^{-v}_{b(u')}}}$$

where $b(u)$ denotes the base station serving user u.

Under elastic traffic, the GREET weight allocation algorithm simply sets the weights proportionally to the ϕ_u values, leading to the following GREET response:

$$w_u^{g}(\mathbf{l}^{-v}) = s^v \frac{\phi_u}{\sum_{u' \in \mathcal{U}^v} \phi_{u'}}$$

Noting that $l_b^{br,v}(\mathbf{l}^{-v})/l_b^{-v} \leq s^v/l_b^{-v} < \delta$, we have that

$$\begin{aligned} w_u^{br}(\mathbf{l}^{-v}) &= s^v \frac{\phi_u \frac{l^{-v}_{b(u)}}{l^{br,v}_{b(u)}(\mathbf{l}^{-v}) + l^{-v}_{b(u)}}}{\sum_{u' \in \mathcal{U}^v} \phi_{u'} \frac{l^{-v}_{b(u')}}{l^{br,v}_{b(u')}(\mathbf{l}^{-v}) + l^{-v}_{b(u')}}} \\ &> s^v \frac{\phi_u \frac{l^{-v}_{b(u)}}{l^{br,v}_{b(u)}(\mathbf{l}^{-v}) + l^{-v}_{b(u)}}}{\sum_{u' \in \mathcal{U}^v} \phi_{u'}} > s^v \frac{\phi_u}{\sum_{u' \in \mathcal{U}^v} \phi_{u'}} \left(\frac{1}{1+\delta} \right) \\ &= w_u^{g}(\mathbf{l}^{-v}) \left(\frac{1}{1+\delta} \right) \end{aligned}$$

and similarly we have that

$$\begin{aligned} w_u^{br}(\mathbf{l}^{-v}) &= s^v \frac{\phi_u \frac{l^{-v}_{b(u)}}{l^{br,v}_{b(u)}(\mathbf{l}^{-v}) + l^{-v}_{b(u)}}}{\sum_{u' \in \mathcal{U}^v} \phi_{u'} \frac{l^{-v}_{b(u')}}{l^{br,v}_{b(u')}(\mathbf{l}^{-v}) + l^{-v}_{b(u')}}} \\ &< s^v \frac{\phi_u}{\sum_{u' \in \mathcal{U}^v} \phi_{u'} \frac{l^{-v}_{b(u')}}{l^{br,v}_{b(u')}(\mathbf{l}^{-v}) + l^{-v}_{b(u')}}} \\ &\leq s^v \frac{\phi_u}{\sum_{u' \in \mathcal{U}^v} \phi_{u'}} (1+\delta) = w_u^{g}(\mathbf{l}^{-v}) (1+\delta) \end{aligned}$$

To show that the GREET equilibrium corresponds to an ε-equilibrium we proceed as follows. Let $f_b^{br,v}(\mathbf{l}^{-v})$ be the fraction of resources obtained by slice v at base station b in the best response to $\mathbf{l}^{-v}$, and let $f_b^{g,v}(\mathbf{l}^{-v})$ be the fraction of resources for the GREET response.

It follows that

$$f_b^{br,v}(\mathbf{l}^{-v}) = \frac{l_b^{br,v}(\mathbf{l}^{-v})}{l_b^{-v} + l_b^{br,v}(\mathbf{l}^{-v})}.$$

Given that the above is a monotonic increasing function in $l_b^{br,v}(\mathbf{l}^{-v})$ and we have that $l_b^{br,v}(\mathbf{l}^{-v}) \leq l_b^{g,v}(\mathbf{l}^{-v})(1+\delta)$, it follows that

$$\begin{aligned} f_b^{br,v}(\mathbf{l}^{-v}) &\leq \frac{l_b^{g,v}(\mathbf{l}^{-v})(1+\delta)}{l_b^{-v} + l_b^{g,v}(\mathbf{l}^{-v})(1+\delta)} \\ &< \frac{l_b^{g,v}(\mathbf{l}^{-v})(1+\delta)}{l_b^{-v} + l_b^{g,v}(\mathbf{l}^{-v})} = (1+\delta) f_b^{g,v}(\mathbf{l}^{-v}). \end{aligned}$$

Both in the best response and the GREET response, the f_b^v resources at base station b are shared among the users of slice v at that base station proportionally to their ϕ_u's. Note that the above holds for any setting of the other slices $\mathbf{l}^{-v}$.

In the argument below we will abuse notation to denote the utility of slice v as a function of the weights of v and those of the users of other slices as $U^v(\mathbf{w}^v, \mathbf{l}^{-v})$. Suppose that we have reached a GREET equilibrium (GE), with weights $\mathbf{w}^{ge}$, and that slice v deviates to take the best response to the base station loads uner GE, $\mathbf{l}^{ge,-v}$. Then, we have the following:

$$\begin{aligned} U^v(\mathbf{w}^{br,v}, \mathbf{l}^{ge,-v}) &= \sum_{u \in \mathcal{U}^v} \phi_u \log(f_u^{br,v}(\mathbf{l}^{ge,-v}) c_u) \\ &< \sum_{u \in \mathcal{U}^v} \phi_u \log((1+\delta) f_u^{ge,v}(\mathbf{l}^{ge,-v}) c_u) \\ &= U^v(\mathbf{w}^{ge,v}, \mathbf{l}^{ge,-v}) + \varepsilon \end{aligned}$$

where $\varepsilon \doteq \sum_{u \in \mathcal{U}^v} \phi_u \log(1+\delta) = \log(1+\delta)$.

Proof of Theorem 7

Let $\mathbf{r}^{ge,v}$ be the allocation to users of slice v under the GREET equilibrium and $\tilde{\mathbf{r}}^v$ be the utility maximizing rate allocation when slice v and $\tilde{v}$ exchange the allocations at the GREET equilibrium. To show envy-freeness we need to show that $U^v(\mathbf{r}^{ge,v}) \geq U^v(\tilde{\mathbf{r}}^v)$,

Following the development of Theorem 4, $U^v(\mathbf{r}^{ge})$ can be expressed as follows:

$$U^v(\mathbf{r}^{ge}) = \sum_{u \in \mathcal{U}^{v,1}} \phi_u \log\left(\frac{m_u^{ge}}{l_b^{ge}} c_u\right) + \sum_{u \in \mathcal{U}^{v,2}} \phi_u \log\left(\frac{m_u^{ge}}{\hat{l}_b^{ge}} \hat{c}_u\right)$$

where $m_u^{ge} = w_u^{ge} - \min[l_b^{ge}\gamma_u/c_u, \gamma_u/c_u]$ is the excess weight allocation to user u under GREET equilibrium, $\mathcal{U}^{v,1}$ is the set of users of slice v at base stations where $l_b^{ge} \leq 1$ and $\mathcal{U}^{v,2}$ is the set of users of slice v at base stations for where $l_b^{ge} > 1$. The quantities $\hat{l}_b^{ge}$, and $\hat{c}_u$ are defined as in the proof of Theorem 4.

In order to characterize the user rate allocations that slice v would obtain with slice $\tilde{v}$'s resources, we shall find the split of the aggregate weight of $\tilde{v}$ at each base station b to users $u \in \mathcal{U}_b^v$ of slice v that maximizes the utility of slice v. We let $\tilde{\mathbf{w}}^v$ denote this weight allocation, $\tilde{\mathbf{m}}^v$ the excess weights and $\tilde{\mathbf{r}}^v$ the resulting rates.

Following the development of Theorem 4 and using the assumption that slice v has logarithmic utilities, $U^v(\tilde{\mathbf{r}})$ can be expressed as follows:

$$U^v(\tilde{\mathbf{r}}) = \sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\left(\frac{\tilde{m}_u}{l_b^{ge}} c_u\right) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\left(\frac{\tilde{m}_u}{\hat{l}_b^{ge}} \hat{c}_u\right)$$

Let us consider the $\tilde{m}_u$ values for $u \in \mathcal{U}^v$ that maximize $U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ge,v})$ subject to the constraint given by (10). One can simplify $U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ge,v})$ as follows

$$\begin{aligned}
&U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ge,v}) = \\
&\quad= \sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\big(\frac{\tilde{m}_u}{l_b^{ge}} c_u\big) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\big(\frac{\tilde{m}_u}{\hat{l}_b^{ge}} \hat{c}_u\big) \\
&\quad- \sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\big(\frac{m_u^{ge}}{l_b^{ge}} c_u\big) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\big(\frac{m_u^{ge}}{\hat{l}_b^{ge}} \hat{c}_u\big) \\
&\quad= \sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\big(\tilde{m}_u c_u\big) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\big(\tilde{m}_u \hat{c}_u\big) \\
&\quad- \sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\big(m_u^{ge} c_u\big) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\big(m_u^{ge} \hat{c}_u\big)
\end{aligned}$$

Since the m_u^{ge} values are fixed, the above optimization is equivalent to finding the $\tilde{m}_u$ values that maximize

$$\sum_{u\in\mathcal{U}^{v,1}} \phi_u \log\big(\tilde{m}_u c_u\big) + \sum_{u\in\mathcal{U}^{v,2}} \phi_u \log\big(\tilde{m}_u \hat{c}_u\big)$$

subject to

$$\sum_{u\in\mathcal{U}^v} \tilde{m}_u = s^v - \sum_{u\in\mathcal{U}^v} \min\left[\frac{l_b^{ge}\gamma_u}{c_u}, \frac{\gamma_u}{c_u}\right].$$

By solving the associated convex optimization problem, one can show that the optimum $\tilde{m}_u$ values satisfy

$$\tilde{m}_u = \frac{\phi_u}{\sum_{u'\in\mathcal{U}^v} \phi_{u'}} \left(s^v - \sum_{u\in\mathcal{U}^v} \min\left(\frac{l_b^{ge}\gamma_u}{c_u}, \frac{\gamma_u}{c_u}\right)\right).$$

This in turn coincides to with the GREET allocation at the equilibrium, i.e., m_u^{ge}. Indeed, GREET first provides the share fraction needed by all users to satisfy the rate requirements and then distributes the remaining share proportional to ϕ_u, which is exactly what the above expression does. This implies that for the above $\tilde{m}_u$ values it holds $U^v(\tilde{\mathbf{r}}^v) - U^v(\mathbf{r}^{ge,v}) = 0$ and hence slice v has no envy for the resources allocated to slice $\tilde{v}$ at a GREET equilibrium.

Proof of Lemma 2

By construction, the GREET weight allocation algorithms allocates to each user the necessary weight to meet its minimum rate requirement.

Proof of Theorem 8

We show convergence by showing that Algorithm 1 is a contraction mapping. Specifically, consider two sequences slice-based weight allocations denoted $(\mathbf{l}(n) : n \in \mathbb{N})$ and $(\tilde{\mathbf{l}}(n) : n \in \mathbb{N})$, where $\mathbf{l}(n) := (l_b^v(n) : v \in \mathcal{V}, b \in \mathcal{B})$ and $\tilde{\mathbf{l}}(n) := (\tilde{l}_b^v(n) : v \in \mathcal{V}, b \in \mathcal{B})$, corresponding to two initial weight allocations denoted denoted $\mathbf{l}(0), \tilde{\mathbf{l}}(0)$ where at each step each slice performs its GREET weight allocation in response to that of the others in the previous step. We will establish that regardless the initial conditions, the following holds:

$$\max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(n) - \tilde{l}_b^v(n)| \;\leq\; \xi \max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(n-1) - \tilde{l}_b^v(n-1)|$$

which suffices to establish convergence as long as $\xi < 1$.

We let $\underline{\mathbf{l}}(n) := (\underline{l}_b^v(n) : v \in \mathcal{V}, b \in \mathcal{B})$ denote the minimal slice weight allocations required by slice v at base station b based on the weight allocations in the previous round, i.e., $\mathbf{l}(n-1)$. Under Assumption 1, only Lines 4 and 5 in Algorithm 1 will be in effect, so

$$\underline{l}_b^v(n) = \begin{cases} \frac{\underline{f}_b^v}{1-\underline{f}_b^v} l_b^{-v}(n-1), & l_b^{-v}(n-1) + \underline{f}_b^v \leq 1, \\ \underline{f}_b^v, & l_b^{-v}(n-1) + \underline{f}_b^v > 1. \end{cases} \tag{12}$$

Again under Assumption 1, the weight allocations for each slice and base station in response to the others $\mathbf{l}(n)$ is given by Line 21 in Algorithm 1, i.e., $l_b^v(n) = \underline{l}_b^v(n) + \phi_b^v\left(s^v - \sum_{b'\in\mathcal{B}} \underline{l}_{b'}^v(n)\right)$ where $\phi_b^v = \sum_{u\in\mathcal{U}_b^v} \phi_u$. Note that two particular cases are as follows: (i) if a slice v has solely inelastic users, we have $\phi_b^v = 0$ and thus $l_b^v(n) = \underline{l}_b^v(n)$; and (ii) if a slice has solely elastic users, then $\underline{l}_{b'}^v(n) = 0$ for all $b' \in \mathcal{B}$ and $l_{\underline{b}}^v(n) = \phi_b^v s^v$. We define $\underline{\tilde{\mathbf{l}}}(n)$ in the same way as $\underline{\mathbf{l}}(n)$, based on $\tilde{\mathbf{l}}(n)$.

Next consider the difference between the two weight allocation sequences. Using the Triangle inequality, we obtain

$$|l_b^v(n) - \tilde{l}_b^v(n)| \leq |\underline{l}_b^v(n) - \underline{\tilde{l}}_b^v(n)| + \phi_v^b \sum_{b'\in\mathcal{B}} |\underline{l}_{b'}^v(n) - \underline{\tilde{l}}_{b'}^v(n)|.$$

Noting that (12) is a concave function with slope no greater than $\frac{\underline{f}_b^v}{1-\underline{f}_b^v}$ and again using the Triangle inequality, we have that

$$\begin{aligned}
|\underline{l}_b^v(n) - \underline{\tilde{l}}_b^v(n)| \quad &\leq \quad \frac{\underline{f}_b^v}{1-\underline{f}_b^v} |l_b^{-v}(n-1) - \tilde{l}_b^{-v}(n-1)| \\
&\leq \quad \frac{\underline{f}_b^v}{1-\underline{f}_b^v} \sum_{v'\neq v} |l_b^{v'}(n-1) - \tilde{l}_b^{v'}(n-1)|.
\end{aligned}$$

Thus, after one round of share updates, we have the following bound:

$$|l_b^v(n) - \tilde{l}_b^v(n)| \leq \frac{\underline{f}_b^v}{1-\underline{f}_b^v} \sum_{v'\neq v} \left|l_b^{v'}(n-1) - \tilde{l}_b^{v'}(n-1)\right|$$

$$+\phi_b^v \sum_{b'\in\mathcal{B}} \frac{\underline{f}_{b'}^v}{1-\underline{f}_{b'}^v} \sum_{v'\neq v} \left| l_{b'}^{v'}(n-1) - \tilde{l}_{b'}^{v'}(n-1) \right|. \quad (13)$$

This in turn leads to the following bound on $\mathbf{l}(n) - \tilde{\mathbf{l}}(n)$:

$$\begin{aligned}
&\max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(n) - \tilde{l}_b^v(n)| \\
&\leq \quad \max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} \left\{ \frac{\underline{f}_b^v}{1-\underline{f}_b^v} \sum_{v'\neq v} |l_b^{v'}(n-1) - \tilde{l}_b^{v'}(n-1)| \right. \\
&\qquad \left. + \, \phi_b^v \sum_{b'\in\mathcal{B}} \frac{\underline{f}_{b'}^v}{1-\underline{f}_{b'}^v} \sum_{v'\neq v} |l_{b'}^{v'}(n-1) - \tilde{l}_{b'}^{v'}(n-1)| \right\} \\
&\leq \quad \frac{2(|\mathcal{V}|-1)f_{\max}}{1-f_{\max}} \max_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} |l_b^v(n-1) - \tilde{l}_b^v(n-1)|,
\end{aligned}$$

where we have used the bound $f_{\max}$ and that $\sum_{b\in\mathcal{B}} \phi_b^v = 1$ unless slice v is inelastic in which case it equals 0. If (7) holds, we have that the weight allocation updates get closer. It follows by Proposition 1.1 in Chapter 3 of [36] that under simultaneous updates one has geometric convergence to the fixed point. Similarly, under round-robin updates, geometric convergence follows as a result of Proposition 1.4 in Chapter 3 of [36].

Proof of Theorem 9

This follows directly from the proof of Theorem 8 and Proposition 2.1 in Chapter 6 of [36].

References

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

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

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

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

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

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

[19] V. Sciancalepore, V. Mancuso, A. Banchs, S. Zaks, and A. Capone, "Interference coordination strategies for content update dissemination in LTE-A," in *Proc. of IEEE INFOCOM*, May 2014.

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

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

[22] 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.

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

[24] 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.

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

[26] 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.

[27] 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.

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

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

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

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

[32] 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.

[33] 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.

[34] 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.

[35] P. Caballero *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.

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

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

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

[39] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," TS 36.213, v12.5.0, Rel. 12, Mar. 2015.

[40] 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.

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

[42] 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.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 20042.

[44] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. of IEEE INFOCOM*, Apr. 2008.