Optimizing Network Slicing via Virtual Resource Pool Partitioning

Pablo Caballero*, Gustavo de Veciana*, Albert Banchs†, Xavier Perez-Costa‡
*The University of Texas at Austin, Austin, TX. Email: pablo.caballero@utexas.edu, gustavo@ece.utexas.edu
†University Carlos III of Madrid and IMDEA Networks Institute, Madrid. Spain. Email: banchs@it.uc3m.es
‡NEC Laboratories Europe, Heidelberg, Germany. Email: xavier.costa@neclab.eu

Abstract*—This paper focuses on optimizing resource allocation amongst a set of tenants, *network slices*, supporting dynamic customer loads over a set of distributed resources, e.g., base stations. The aim is to reap the benefits of statistical multiplexing resulting from flexible sharing of 'pooled' resources, while enabling tenants to differentiate and protect their performance from one another's load fluctuations. To that end we consider a setting where resources are grouped into *Virtual Resource Pools* (VRPs) wherein resource allocation is jointly and dynamically managed. Specifically for each VRP we adopt a Share-Constrained Proportionally Fair (SCPF) allocation scheme where each tenant is allocated a fixed share (budget). This budget is to be distributed equally amongst its active customers which in turn are granted fractions of their associated VRP resources in proportion to customer shares. For a VRP with a single resource, this translates to the well known Generalized Processor Sharing (GPS) policy. For VRPs with multiple resources SCPF provides a flexible means to achieve load elastic allocations across tenants sharing the pool. Given tenants' per resource shares and expected loads, this paper formulates the problem of determining optimal VRP partitions which maximize the overall expected shared weighted utility while ensuring protection guarantees. For a high load/capacity setting we exhibit this network utility function explicitly, quantifying the benefits and penalties of any VRP partition, in terms of network slices' ability to achieve performance differentiation, load balancing, and statistical multiplexing. Although the problem is shown to be NP-Hard, a simple greedy heuristic is shown to be effective. Analysis and simulations confirm that the selection of optimal VRP partitions provide a practical avenue towards improving network utility in network slicing scenarios with dynamic loads.

I. Introduction

It is widely agreed, see e.g., multiple standardization and industrial efforts [2], [30], that enabling *network slicing* to support multi-tenancy on shared infrastructure will be a key component to enable the success of next generation wireless networks. Network slicing allows to partition physical network resources among multiple fully-functional and configurable logical networks, or slices, each assigned to a tenant or service, providing them the opportunity to customize the network functions to their requirements. Network slicing is expected to reduce deployment and operational expenditures by enabling infrastructure sharing between multiple tenants, e.g., mobile virtual network operators and over the top service providers, as well as joint investments by infrastructure providers. It is expected to be a critical ingredient towards tailoring distributed compute/communication resources to meet the stringent requirements of next generation applications.

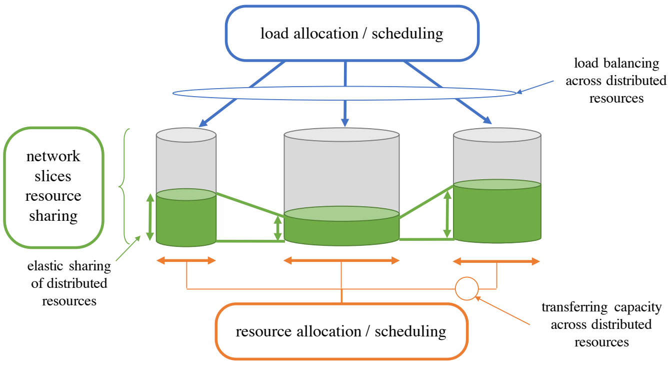
load allocation / scheduling
load balancing across distributed resources
network slices resource sharing
elastic sharing of distributed resources
transferring capacity across distributed resources
resource allocation / scheduling

Fig. 1. Alternative mechanisms to reap the benefits of resource pooling.

A key component underlying network slicing is a scalable mechanism for resource allocation across tenants supporting dynamic mobile customer loads, when the infrastructure resources are spatially distributed, e.g., wireless base stations, edge computing resources. It should be efficient in that it promotes statistical multiplexing gains while protecting tenants from one another's traffic load variations. Ideally one would like the network to behave as if it were a *centralized pool* of resources which can be flexibly allocated to tenants spatio-temporal loads [35]. However, in typical wireless networks a slice's active customers must associate with local/proximal resources. In turn if network resources are statically allocated, variations in slices' customer loads may lead to poor per customer performance and variability. The challenge is thus to devise dynamic resource allocation schemes which adapt to customer loads but still provide a degree of protection amongst slices, i.e., improve the effective pooling of resources.

Figure 1 exhibits different approaches towards achieving improved resource pooling – for a taxonomy of these techniques in the context of wireless networks see [19]. Broadly speaking traditional approaches are of two types: "*load allocation*" and "*resource allocation*" mechanisms. In the wireless context, load allocation might correspond to shifting customer loads by exploiting possible diversity in customer to base station association, routing and other load balancing mechanisms. Clearly the flexibility of such mechanisms in wireless networks is limited. Alternatively one can also consider resource allocation mechanisms, wherein one shifts capacity to regions where there are currently more demands. In wireless settings this could in principle be done by borrowing spectrum, cell breathing, borrowing cloud based compute resources from neighboring base stations, etc. When distributed resources are shared among multiple network slices, it is possible to capital-

The work of P. Caballero and G. de Veciana was supported in part by a gift from Cisco. This research was partially supported by NSF grant CNS-1731658.

ize on complementary spatio-temporal demands across tenants by dynamically shifting resource allocations within and across slices to achieve improved per slice/customer performance. Doing so requires joint management of allocations to tenants' customers over sets of resources – we shall refer to these as *Virtual Resource Pools* (VRPs).

The main problem addressed in this paper centers on devising optimal VRP partitions of infrastructure resources to best meet tenant's needs. Small VRP pools allow tenants fine grain control on their resource allocations, providing improved performance differentiation and protection. Larger VRP pools enable the overall network to achieve improved statistical multiplexing through elastic resource allocations, i.e., allocations that dynamically track tenant's customer loads. An optimal VRP partition achieves the best overall tradeoffs by maximizing the weighted network utility subject to protection guarantees. In fact we shall see VRP partitions can be asymptotically characterized precisely in terms of how they 'deviate' from homogeneous centralized resource pooling solutions.

A. Related Work

As mentioned above improved resource pooling may be realized through both load or resource allocation techniques. Load allocation techniques include dynamic routing policies, such as join the shortest queue, see e.g., [17], [29] or queue with the smallest expected delay, e.g., [24]. Such mechanisms have proven to be very effective resource pooling enablers. Flexibility in routing wireless customers to resources (e.g. base stations) is somewhat limited, whence our approach to achieve resource pooling is focused on multi-tenant sharing and resource allocation.

Resource allocation techniques, include per-resource mechanisms that have been designed to achieve fairness among customers, such as Proportional Fairness and Processor Sharing, and their multi-class equivalents Weighted Proportional Fairness [3] and Generalized Processor Sharing [31] have seen wide applicability in wireless networks. However, a focus on resource management on a per resource basis, e.g., base station, fails to exploit potential benefits of coupled management across resources, see e.g. [8], [37]. Consequently, researchers have explored joint resource management [8], [27], [5] and have shown its effectiveness at enabling improved resource pooling. The above-mentioned works focus on single-tenant networks; in contrast, this paper focuses on slicing and sharing resources among multiple tenants. Recently, some extensions of resource allocation mechanisms for multi-tenant wireless networks were studied in [4], [28], [16], [26]. The reader is referred to [32] for a survey on resource slicing techniques for virtual wireless networks.

The above multi-tenant resource allocation mechanisms have for the most part focused in 'elastic' users whose sojourn times would depend on the allocated rate. This coupling makes the study of dynamic customer loads challenging. In this paper, instead, we focus on customers whose network activity is independent of their resource allocation (e.g., video, voice and other rate-adaptive user sessions) but which also favor higher rates/utility. In [9], [37] the Share-Constrained Proportionally Fair mechanism adopted in this paper was proposed, where tenants are allocated a share of the overall network resources which is redistributed dynamically based on the tenants customer loads. In [37] the authors showed that this mechanism achieves improved statistical multiplexing, resulting in capacity savings versus per-resource mechanisms such as GPS and characterized these gains, suggesting that load spatial distribution impacts the perceived gains as well as the degree of tenants' isolation and performance variability.

Although optimal network partitioning has been object of studies for decades in several contexts with different applications [18], [14], [33], [34], this work is, to the authors best knowledge, the first attempt at formally studying joint resource management of network slices on VRPs.

II. System model

We start by defining our multi-tenant mobile network model. The network is comprised of a set $\mathcal{B} = \{1, 2, \ldots, |\mathcal{B}|\}$ of $|\mathcal{B}|$ *resources* spatially distributed and shared by a set $\mathcal{O} = \{1, 2, \ldots, |\mathcal{O}|\}$ of $|\mathcal{O}|$ tenants (also denoted as *network slices*).

The tenants' traffic load is assumed to be stochastic and at a given point in time, the network supports a set of users $\mathcal{U}$ (the customers or devices). For the rest of the paper, we will assume that each user can belong to only one slice and be served by one resource at a time. The set of users $\mathcal{U}$ can be subdivided into $\mathcal{U}^o$ and $\mathcal{U}_b$ which represent respectively the set of customers of a slice o and the set of customers served by a resource b. $\mathcal{U}_b^o$, which correspond to the intersection of the previous sets, will be used to denote the set of users of slice o at station b.

The distribution of the vector of random variables $\mathbf{N} = (N_b^o : b \in \mathcal{B}, o \in \mathcal{O})$ characterizes the marginal distribution of the number of active users on the network and $\boldsymbol{\rho} = (\rho_b^o : b \in \mathcal{B}, o \in \mathcal{O})$ denotes their mean loads. The traffic load state at a certain instant is represented by $\mathbf{n} = (n_b^o : b \in \mathcal{B}, o \in \mathcal{O})$, where $n_b^o = |\mathcal{U}_b^o|$ represents the active number of users from slice o at resource b.

Each slice o requests a share $s_b^o \in [0, 1]$ of each network resource $b \in \mathcal{B}$. We denote the resource share request by a vector $\mathbf{s}$ given by

$$\mathbf{s} = (\mathbf{s}^o : o \in \mathcal{O}) \quad \text{where} \quad \mathbf{s}^o = (s_{b_1}^o, s_{b_2}^o, \ldots, s_{b_{|\mathcal{B}|}}^o). \tag{1}$$

The aggregate share request for a given resource is assumed to not exceed 1, i.e., for all $b \in \mathcal{B}$ we have that

$$s_b \triangleq \sum_{o \in \mathcal{O}} s_b^o \leq 1. \tag{2}$$

This assumption is made without loss of generality, since the shares correspond to relative quantities across entities contending for resources, and can always be normalized. However, the above normalization is the most natural since provides slices with the notion that a share corresponds to a fraction of the resource.

A. Virtual Resource Pools allocation

In this work, we aim to determine a partition $\mathcal{P}$ of the resource set $\mathcal{B}$ into a collection of VRPs

$$\mathcal{P} = \{P_i \mid i = 1, \ldots, |\mathcal{P}|\}. \tag{3}$$

Each of the subsets $P_i \subset \mathcal{B}$ of the partition will act as a VRP. The idea underlying multi-tenant sharing of a VRP is as follows. From a resource allocation perspective, the goal is to provide slices with more flexible allocations by allowing to dynamically use the requested share in a particular resource in a different resource of the virtual pool, to better adapt to the instantaneous traffic conditions. To this end, we will assume that tenants have a fixed share (which may be understood as a budget) of the virtual resource pool, which it is assumed to be equal to the sum of their aggregated shares of the pool's constituent resources, i.e., slice o has a share $s^o(P_i)$ at virtual pool P_i given by $s^o(P_i) \triangleq \sum_{b\in P_i} s_b^o$.

Note that the sum of shares over a pool are not restricted to be less than 1. As mentioned earlier, only the relative shares of each slices will be relevant in the sequel. Furthermore, we will let $n^o(P_i) \triangleq \sum_{b\in P_i} n_b^o$ denote the number of active users of slice o at pool P_i.

Next, we formally define our proposed multi-tenant resource allocation for a VRP.

Definition 1. *(**VRP resource allocation**) Each virtual pool P_i is composed by a collection of resources shared by several slices, each of them having a share equal to $s^o(P_i)$. At any instant, all $n^o(P_i)$ users of slice o are assigned an equal portion of $s^o(P_i)$ as a weight $w^o(\mathbf{n}, P_i)$, i.e.,*

$$w^o(\mathbf{n}, P_i) = \frac{s^o(P_i)}{n^o(P_i)}, \quad \forall o \in \mathcal{O},\ P_i \in \mathcal{P}. \tag{4}$$

Resource allocation among slices at each resource b in the virtual pool P_i is performed in proportion to the weights, i.e., the fraction of resource b allocated to any user of slice o is equal and given by

$$f_b^o(\mathbf{n}, P_i) = \frac{w^o(\mathbf{n}, P_i)}{\sum\limits_{v\in\mathcal{O}} w^v(\mathbf{n}, P_i) \cdot n_b^v \cdot \mathbf{1}(n_b^v > 0)}. \tag{5}$$

For any user u from slice o served by resource b, we can represent its allocated service rate as

$$r_u^o(\mathbf{n}, P_i) = t_u \cdot f_b^o(\mathbf{n}, P_i). \tag{6}$$

where t_u is the achievable transmission rate of user u if the user had the entire resource b to itself.

The transmission rate $t_{u'}$ for any user u' at resource b, denotes a realization of a random variable $T_{u'}$, which is a copy of the random variable T_b that characterize the marginal distribution of the achievable rate of any user at station b, which is assumed to be independent across slices and users, i.e., only time and space dependent.

We note that the notion of a VRP represents an abstraction. Indeed, this notion can be applied to any scenario where underlying physical resources might be at different spatial locations they may not be interchangeable in terms of serving a particular tenants' users sharing the pool. We say that virtual pool physical resource capacities may **not be transferable** to adapt to spatial variations in the traffic conditions. Additionally, we shall for simplicity assume that resources a user can only be served by one resource at a time, as it is usually the case in wireless networks.

B. Benchmark allocations

In the sequel, we will contrast the performance of a network under VRP partition $\mathcal{P}$ with two benchmark partitions:

1) *Generalized Processor Sharing (GPS)* [31]: partition of the resources into $|\mathcal{B}|$ VRPs, each with a single resource, i.e., $\mathcal{P}^{GPS} = \{\{b\} : b \in \mathcal{B}\}$.
2) *Complete Pooling (CP)*: partition with a single VRP containing all of the resources, i.e., $\mathcal{P}^{CP} = \{\mathcal{B}\}$.

C. Share, load and capacity distributions

Next, we introduce some definitions and notation that will be used in the sequel.

Definition 2. *We define the normalized shares and the normalized active number of users distributions of slice o on VRP P_i, respectively, as follows*

$$\tilde{\mathbf{s}}^o(P_i) = (\tilde{s}_b^o(P_i) : b \in P_i),\ \textit{where}\ \tilde{s}_b^o(P_i) \triangleq \frac{s_b^o}{s^o(P_i)},$$
$$\tilde{\mathbf{n}}^o(P_i) = (\tilde{n}_b^o(P_i) : b \in P_i),\ \textit{where}\ \tilde{n}_b^o(P_i) \triangleq \frac{n_b^o}{n^o(P_i)}.$$

Definition 3. *We define the overall normalized share distribution over a partition $\mathcal{P}$ as*

$$\hat{\mathbf{s}}(\mathcal{P}) = (\hat{s}_{P_i}^o : o \in \mathcal{O}, P_i \in \mathcal{P}),\ \textit{where}\ \hat{s}_{P_i}^o \triangleq \frac{s^o(P_i)}{s},$$

where $s = \sum\limits_{o\in\mathcal{O},b\in\mathcal{B}} s_b^o$ is the total share. We further define the normalized load distribution over a partition $\mathcal{P}$ by

$$\hat{\boldsymbol{\rho}}(\mathcal{P}) = (\hat{\rho}^o(P_i) : o \in \mathcal{O}, P_i \in \mathcal{P}),\ \textit{where}\ \hat{\rho}^o(P_i) \triangleq \frac{\rho^o(P_i)}{\rho},$$

where $\rho = \sum\limits_{o\in\mathcal{O},b\in\mathcal{B}} \rho_b^o$ denotes the total system mean load.

Definition 4. *We define the share weighted normalized relative number of active users distribution as*

$$\hat{\mathbf{g}}(\mathbf{n}, \mathcal{P}) = (\hat{g}_b(\mathbf{n}, P_i)\ :\ b \in \mathcal{B})$$
$$\textit{where}\qquad \hat{g}_b(\mathbf{n}, P_i) \triangleq \sum_{o\in\mathcal{O}} \hat{s}_{P_i}^o\, \tilde{n}_b^o(P_i)\, \mathbf{1}(n_b^o > 0).$$

We adopt the convention that $0/0 = 1$ if $n_b^o = n^o(P_i) = 0$. Note that $\hat{\mathbf{g}}(\mathbf{n}, \mathcal{P})$ can also be interpreted as a mixture distribution of the $\tilde{n}^o(P_i)$ distributions with weights $\hat{s}_{P_i}^o$. [1]

We define the equivalent share weighted normalized relative mean load distribution as

$$\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P}) = (\hat{g}_b(\boldsymbol{\rho}, P_i)\ :\ b \in \mathcal{B}),\ \textit{where}\ \hat{g}_b(\boldsymbol{\rho}, P_i) \triangleq \sum_{o\in\mathcal{O}} \hat{s}_{P_i}^o\, \tilde{\rho}_b^o(P_i).$$

III. VRP PARTITIONING

This study focuses on finding the optimal partition of VRPs $\mathcal{P}$ to be chosen by the infrastructure provider. Creating such VRPs of many resources enables the ability to absorb bursty traffic variations by exploiting statistical multiplexing. I.e., by jointly managing several base stations in the same VRP, we allow resource allocation to absorb the traffic variations

[1] In the definition of $\hat{g}_b(\mathbf{n}, P_i)$ we have abused notation when denoting the b^{th} component of the vector, since for clarity of reading, we identify that $\hat{g}_b$ depends only of P_i and not the complete partition $\mathcal{P}$.

of slices across resources transferring share from one resource to another, consequently improving the users expected performance. However, pooling may reduce the ability of guaranteeing each slice a desired degree of protection, e.g., by strictly enforcing the per slices shares s_b^o at each resource. Moreover, geographical and architectural network constraints may need to be considered limiting the resources that can be pooled together. Taking into account these aspects, in this section we describe the optimal VRP partitioning problem.

A. Stochastic network utility

The optimal VRP partition will be set to maximize a certain network statistic, that reflects the overall network performance and which we will define by the means of a utility function. To obtain this function, first we will define a relevant statistic of utility per slice and pool to continue with a discussion on how to combine the various utilities along and across slices and pools to generate a global network statistic.

Recall that the number of active users on each slice and resource are modeled by a random vector $\mathbf{N}$. We shall define the expected network utility as follows. We consider, as in [20], the utility of a user as the logarithm of its rate and let $U^o(P_i)$ denote the expected utility of a **typical user** of slice o on VRP P_i, i.e., the expected log rate of a randomly selected user of slice o on VRP P_i. This quantity is given by

$$\begin{aligned} U^o(P_i) &= \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o} \mathbb{E}\left[\frac{1}{\mathbb{E}[N^o(P_i)]}\log(T_u\cdot f_b^o(\mathbf{N},P_i))\right] \\ &= \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\frac{\mathbb{E}\left[\log(T_u)\right]}{\rho^o(P_i)} + \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\frac{\mathbb{E}\left[\log(f_b^o(\mathbf{N},P_i)\right]}{\rho^o(P_i)}. \end{aligned}$$

To simplify notation, from the rest of the paper we will rely on the following definition of effective capacities.

Definition 5. ***(Effective capacities)*** *We will define the effective capacities of the set of resources b as*

$$\mathbf{c} = (c_b > 0:\ b\in\mathcal{B}), \quad \textit{where} \quad c_b = e^{\mathbb{E}[\log(T_u)]}. \qquad (7)$$

At this point, it is worth noting that this effective capacity is a mechanism to generalize the traditional concept of resource capacity, where its various end users may have different achievable service rates. If T_u (and therefore T_b) is a constant, the notion of effective capacity corresponds to the traditional notion of resource capacity.

Then, it follows that

$$\begin{aligned} U^o(P_i) &= \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\frac{\log(c_b)}{\rho^o(P_i)} + \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\frac{\mathbb{E}\left[\log(f_b^o(\mathbf{N},P_i)\right]}{\rho^o(P_i)} \\ &= \mathbb{E}\left[\sum_{b\in P_i}\frac{N_b^o}{\rho^o(P_i)}\log(c_b\cdot f_b^o(\mathbf{N},P_i))\right]. \end{aligned}$$

Recall that a "typical" user here should be viewed as a randomly selected user of slice o on VRP P_i, whence the utility of the user is weighted by $\frac{N_b^o}{\rho^o(P_i)}$ to reflect uneven loads on the VRP's resources. To deal with the case where the number of active users is zero, i.e., $N_b^o = 0$ we have used the convention (see e.g., [11]) $0\cdot\log(0)\triangleq 0$.

Then, the overall expected network utility is given by a weighted combination of the slices' utilities per VRP. We define the overall expected utility to account for slices shares of the network resources. The typical user utility of a slice with a higher share per user load, i.e., $\frac{s^o(P_i)}{\rho^o(P_i)}$, should be given a higher weight. Furthermore, if the slice has a higher load $\rho^o(P_i)$ should be prioritized thus the overall weight is $\rho^o(P_i)\frac{s^o(P_i)}{\rho^o(P_i)} = s^o(P_i)$. This gives an overall utility

$$U(\mathcal{P}) = \sum_{P_i\in\mathcal{P}}\sum_{o\in\mathcal{O}}\hat{s}_{P_i}^o\, U^o(P_i) = \sum_{o\in\mathcal{O}} U^o(\mathcal{P}). \qquad (8)$$

where we have defined the utility of a tenant $U^o(\mathcal{P})$ as the share weighted combination of their expected utility of a **typical user** per VRP P_i,

$$U^o(\mathcal{P}) = \sum_{P_i\in\mathcal{P}}\hat{s}_{P_i}^o\, U^o(P_i) \qquad (9)$$

and we have included the division by the normalization constant s (independent of $\mathcal{P}$) for clarity of future results.

In summary, the overall expected network utility accounts for the slices loads and share per load on various resources by weighting the relative importance of each slices users' utility.

B. Slices protection guarantees

Classical allocation schemes, such as GPS provide protection, i.e., for slice o on resource b it ensures an allocation of at least s_b^o (since $\sum_{o\in\mathcal{O}} s_b^o \leq 1$). However, in a multi-resource network, the inability of the resource allocation to adapt to the traffic variations across resources indicates that fair allocation schemes may benefit from a network-wide view [8] where an example of this is the resource allocation proposed for a VRP.

Naturally, adopting a pool-wide allocation scheme may compromise such guarantees among slices of GPS. Thus, it is desirable to provide slices with a pool-wide notion of relaxed performance isolation. Hereby, we define the following VRP notion of protection, that ensures that if a slice share request is appropriately chosen in proportion to its traffic demands, it will always benefit from sharing.

Definition 6. ***(Slice protection)*** *We say a slice is* ***protected*** *at a VRP if, as long as the slices' number of active users is proportional to its share, i.e., if for all $b\in P_i$, $n_b^o \approx \gamma s_b^o$, it is possible to ensure that*

$$\sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\log\left(t_u f_b^o(\mathbf{n},P_i)\right) \geq \sum_{b\in P_i}\sum_{u\in\mathcal{U}_b^o}\log\left(t_u\frac{s_b^o}{n_b^o}\right) \qquad (10)$$

i.e., that a slice can obtain, at least, a utility at each pool greater than if the slice would receive at each resource a fraction of resource equal to its share s_b^o.

We say that a slice is protected at the network if the condition in Eq. (10) *is fulfilled for every VRP P_i in $\mathcal{P}$.*

Note that under this notion of protection, a slice whose loads align with its share requests is guaranteed better utility through VRP pools, irrespective of the number of active users on other slices. A sufficient protection condition (both per pool and per VRP) is presented in the following lemma.

Lemma 1. *A sufficient condition to ensure protection for a slice o in a VRP P_i is*

$$\mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) = -\sum_{b\in P_i} \frac{s_b^o}{s^o(P_i)} \log\left(\frac{s_b^o}{s^o(P_i)}\right) \geq \log(s(P_i)), \tag{11}$$

where $\mathrm{H}\left(\tilde{\mathbf{s}}^{o}(P_i)\right)$ *is the entropy of the* normalized share distribution *of slice o on pool P_i and* $s(P_i) = \sum_{o\in\mathcal{O},b\in P_i} s_b^o$ *is the total VRP share.*

Therefore, the set of partitions that provide protection at the network to slice o are given by:

$$\mathcal{C}_p^o = \{\mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) \geq \log(s(P_i)), \forall P_i \in \mathcal{P}\} \tag{12}$$

The proof of this result has been relegated to the extended version of this paper in [10].

We note that protection only depends on the share distribution of slice o, and aggregated shares of the slices on each pool. Remarks on the protection condition are presented next.

Remark 1. Note that the entropy of a discrete distribution is bounded by log of the cardinality of the support [13] $0 \leq \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) \leq \log(|P_i|)$. We can thus conclude that

1) If the share distribution of slice o is uniform, the entropy is maximized $\mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}\right) = \log(|P_i|)$ and the protection condition will always be fulfilled irrespective of the aggregate share $s(P_i)$, since $s(P_i) \leq |P_i|$
2) If the slice share requests in a pool are maximal, i.e., $s(P_i) = |P_i|$ the protection condition is only fulfilled if the demand distribution of slice o is uniform, i.e., $\mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) = \log(|P_i|)$ only if $\tilde{\mathbf{s}}^{\mathbf{o}}(P_i) = \frac{1}{|P_i|}$.
3) If $s(P_i) \leq 1$, and thus $\log(s(P_i)) \leq 0$ the protection condition will always be fulfilled irrespective of the demand distribution of slice o, since the entropy is positive. In such scenario, there is enough slack in the VRP shares to ensure protection at all times.
4) The finest grain partition $\mathcal{P}^{GPS}$ always achieves protection, as a direct consequence of the previous point.

We can define the set of protection constraints as follows

Definition 7. ***(Protection constraint set)*** *Considering $\hat{\mathcal{O}} \in \mathcal{O}$ as the set of slices that demand protection constraints at the network, the protection constraint set can be defined as*

$$\mathcal{C}_p = \bigcap_{o\in\hat{\mathcal{O}}} \mathcal{C}_p^o. \tag{13}$$

C. Design constraints

Even with the protection constraints satisfied, some partitions may be impractical/inefficient for the network infrastructure provider. Realizing VRPs requires an exchange of information within the resources, which may impose some architectural or design constraints. For instance, a virtual controller may have capacity to coordinate a maximum number of resources $\bar{K}$. To capture design constraints associated with the limitations of the architecture in terms of pooling management capacity, we will define the following constraint

$$\mathcal{C}_c = \{\ \mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid |P_i| \leq \bar{K},\ \forall P_i \in \mathcal{P}\ \}. \tag{14}$$

Also, in some settings, it may be desirable that the creation of VRPs is based on resources that are nearby, which decreases the impact of users handoffs, or physically interconnected, which increases the information sharing capacity. To that end consider a graph $\mathcal{G}(N, E)$ whose nodes are $N = \mathcal{B}$ and the edges $e_{i,j} \subset N \times N$ denote resources that are neighbors or interconnected. A partition $\mathcal{P} = (P_1, P_2, \ldots P_{\mathcal{P}})$ can be viewed as the partition of $\mathcal{G}(N, E)$ into a collection of subgraphs $G_i(N, E)$ whose $E_i = E \cap (P_i \times P_i)$. A logical architectural requirement on the partition could be that $G_i(N, E)$ are connected subgraphs. We will abstract this constraint as follows

$$\mathcal{C}_l = \{\mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid p(G_i(P_i, E_i)) = 1,\ \forall P_i \in \mathcal{P}\}. \tag{15}$$

where $p(G_i(P_i, E_i))$ is equal to 1 if the subgraph is connected.

D. Optimal VRP Partitioning

Joining the constraints from previous subsections, we can define the partition constraints as $\mathcal{C} = \mathcal{C}_p \cap \mathcal{C}_c \cap \mathcal{C}_l$ where $\mathcal{C}_p,\ \mathcal{C}_c,\ \mathcal{C}_l$ are defined in Eqs. (13)-(15) respectively. We can write the optimal spatial pooling problem as the following optimization.

Definition 8. ***(Optimal VRP Partitioning Problem (OVP))*** *The Optimal VRP Partition is given by*

$$\max_{\mathcal{P}} \quad \{\ U(\mathcal{P}) \mid \mathcal{P} \in \mathcal{C}\ \}. \tag{16}$$

Unfortunately, finding the solution of OVP is a complex problem for several reasons: (i) evaluating the utility function implies finding the expected value of a non-linear function of random variables, which is per se a hard problem and (ii) the possible number of feasible of partitions that need to be considered in order to find the optimal pooling increases exponentially with the number of resources (in accordance to the Bell numbers [6]). In fact, this is already a problem even in the case where the loads are not stochastic. The combinatorial aspect of this problem can be translated into a more formal notion of algorithmical complexity.

Theorem 1. *Optimal VRP Partitioning is NP-Hard.*

The technical proof of this result has been relegated to [10].

In order to overcome the high complexity of finding the exact solution for the OVP, we propose an algorithm based on the idea of cost-benefit greedy algorithm [22].

E. Greedy algorithm for OVP

The algorithm is initialized by a GPS partition $\mathcal{P}^{(GPS)}$, which is always feasible Then it iteratively considers merging VRPs so as to ensure the fulfillment of the constraints while maximizing benefit to cost ratio. We define the benefit as the utility improvement and the cost $\mathcal{H}(\hat{\mathcal{P}}^{i,j})$ as the inverse of the share entropy, i.e., $\mathcal{H}(\mathcal{P}) = \left(\sum_{P_i\in\mathcal{P}} \sum_{o\in\mathcal{O}} H(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i))\right)^{-1}$. Therefore, the gain over cost ratio of joining P_i and P_j is

$$\delta U(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}}) = \frac{U(\hat{\mathcal{P}}^{i,j}) - U(\hat{\mathcal{P}})}{\mathcal{H}(\hat{\mathcal{P}}^{i,j})}$$

where $\hat{\mathcal{P}}^{i,j} = \{\hat{\mathcal{P}} \backslash \{P_i, P_j\}\} \cup \{P_i \cup P_j\}$. This is motivated by the fact that, despite our aim is to maximize network utility, a low share entropy may impact the ability to meet the protection constraints in future possible merges. In order to evaluate the utility improvement of a possible merge, one must evaluate the expected network utility $U(\hat{\mathcal{P}}^{i,j})$, which can be performed by Monte Carlo sampling methods or appropriate approximations.

This is repeated until the algorithm does not find any beneficial merge or all resources has been aggregated into a single pool, i.e., the partition is equal to $\mathcal{P}^{CP}$.

IV. Utility analysis

To provide insights the different factors that impact utility when choosing a partition, we focus on analyzing the asymptotic case when capacities and loads grow linearly, reducing the load variations around the mean. To this end, we consider a sequence of networks, indexed by β, as follows.

Assumption 1. ***(Linear scaling)*** *Consider a share vector* $\mathbf{s} > 0$, *a load vector* $\boldsymbol{\rho} > 0$ *and resource capacity vector* $\mathbf{c} > 0$ *and a sequence of networks indexed by* β. *For the* β^{th} *network, the stochastic numbers of active users* $\mathbf{N}^{(\beta)} = (N_b^{o,(\beta)} : o \in \mathcal{O}, b \in \mathcal{B})$ *are mutually independent and Poisson distributed with strictly positive means* $\beta \cdot \boldsymbol{\rho}$, *i.e.,* $N_b^{o,(\beta)} \sim$ *Poisson*$(\beta \cdot \rho_b^o)$ *and the resource capacities* $\mathbf{c}^{(\beta)} = \beta \mathbf{c}$ *such that* $c_b^{(\beta)} = \beta c_b$. *We let* $U^{(\beta)}(\mathcal{P})$ *denote the expected network utility, given Eq.* (8), *of the* β^{th} *network.*

Theorem 2. *Under Assumption 1, the expected network utility under partition* $\mathcal{P}$ *is given by*

$$U^{(\beta)}(\mathcal{P}) = \log\left(\frac{c}{\rho}\right) + \mathrm{D}(\mathcal{P}) - \mathrm{M}(\mathcal{P}) - \frac{S(\mathcal{P})}{\beta} + o\left(\frac{1}{\beta}\right), \quad (17)$$

where $\mathrm{D}(\mathcal{P}) = \mathrm{D}_{KL}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right), \mathrm{M}(\mathcal{P}) = \mathrm{D}_{KL}\left(\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P})||\hat{\mathbf{c}}\right);$ *where* D_{KL} *stands for the KL divergence [23]. Also,*

$$\mathrm{S}(\mathcal{P}) = \langle \hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P}) \rangle + \langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P}) \rangle \quad (18)$$

$$\begin{aligned} \textit{where} \qquad \mathbf{q}(\mathcal{P}) &= \left((\rho^o(P_i))^{-1} : o \in \mathcal{O}, P_i \in \mathcal{P}\right), \\ \mathbf{h}(\mathcal{P}) &= (h_b^o(\mathcal{P}) : o \in \mathcal{O}, b \in \mathcal{B}), \\ h_b^o(\mathcal{P}) &= \sum_{b' \in \mathcal{B}} \frac{\partial^2 \left(\bar{g}_{b'}(\mathbf{x}, P_i)\, \log\left(\hat{g}_{b'}(\mathbf{x}, P_i)\right)\right)}{\partial (x_b^o)^2}\Bigg|_{\boldsymbol{\rho}_b^o} \\ \textit{and} \quad \bar{g}_{b'}(\mathbf{x}, P_i) &= \sum_{v \in \mathcal{O}} \frac{\hat{s}^v(P_i)}{\rho^v(P_i)} x_{b'}^v. \end{aligned}$$

The proof of this result and the subsequent Facts 2 and 3 in this section are technical and has been relegated to [10].

To better understand the above result, we will analyze the impact of the different components in the utility function.

Remark 2. The utility $U^{(\beta)}(\mathcal{P})$ serves to rank a partition $\mathcal{P}$ based on the **load, shares and capacity distributions** as well as by how **statistical multiplexing** is realized in its associated VRPs. Let us consider Eq. (17) in more detail.

i) The **homogeneous perfect pooling utility** $\log(c/\rho)$ corresponds to the utility of a system where the total effective network capacity c (i.e., the sum of the effective capacities of all resources in $\mathcal{B}$) is pooled and equally divided among its mean total number of users ρ.

ii) The **slice differentiation gain** is such that

$$\mathrm{D}(\mathcal{P}) = \mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right) \geq 0$$

and only equals zero if the per slice and partition normalized shares $\hat{\mathbf{s}}(\mathcal{P})$ and loads $\hat{\boldsymbol{\rho}}(\mathcal{P})$ distributions coincide. When the distributions diverge, the term increases resulting in slice differentiation gains relative to $\log(c/\rho)$.

iii) The **load misalignment loss** is such that

$$\mathrm{M}(\mathcal{P}) = \mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P})||\hat{\mathbf{c}}\right) \geq 0$$

and equals zero if the weighted normalized load distribution $\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P})$ and normalized capacity distributions

$$\hat{\mathbf{c}} = \left(\frac{c_b}{c} : b \in \mathcal{B}\right),$$

are equal, otherwise the losses increase as they diverge.

iv) The **stochastic pooling losses** $\mathrm{S}(\mathcal{P})/\beta$ capture a utility loss arising from the variation in the number of active users relative to their mean loads. Each partition exploit statistical multiplexing differently, resulting into different stochastic pooling losses. The losses decrease with β, vanishing as $\beta \to \infty$, since under Poisson distribution as $\beta \to \infty$ the number of active users concentrates.

For a general network, the expected utility of a VRP partition will reflect the ability to differentiate performance (typical user utilities) across slices and resources, i.e., inter and intra slice differentiation as well as the balancing of load and statistical multiplexing losses.

Note that, in general, $\mathrm{D}(\mathcal{P}) - \mathrm{M}(\mathcal{P})$ can be either negative or positive, as we can observe in the following scenarios.

1) Consider a network where the loads are proportional to the shares, i.e., $\rho_b^o = \gamma s_b^o$ and $\mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right) = 0$ but the capacities are not aligned with the share weighted loads. Given the proportionality of loads and shares, the misalignment term M is independent of $\mathcal{P}$.

   Fact 1. *If the loads are equally in proportion to the shares* $\rho_b^o = \gamma s_b^o$, *the share weighted pool load is independent of the partition, i.e.,* $\hat{\mathbf{g}}(\rho, \mathcal{P}) = \hat{\mathbf{g}}(\rho) = \frac{1}{\rho}\sum_{o \in \mathcal{O}} \rho_b^o$.

   Then the utility is given by

   $$U^{(\beta)}(\mathcal{P}) = \log\left(\frac{c}{\rho}\right) - D_{\mathrm{KL}}(\hat{\mathbf{g}}(\rho)||\hat{\mathbf{c}}) - \frac{1}{\beta}\,\mathrm{S}\,(\mathcal{P}) + o\left(\frac{1}{\beta}\right).$$

   In this case, we can see that the network deviates from acting as a single pool since the resource capacities across resources are misaligned with the share weighted load distribution. Note that this occurs since once deployed the resources of a network, with their respective capacities, these capacities are non transferable among resources. [2]

2) If resource capacities were transferable among resources or were engineered to coincide with the share weighted

[2] Note that, despite we are not considering that capacities are transferable among resources; in cellular networks, certain resource capacities transferability can be achieved in several ways as for example by having a C-RAN [12] that use its computational capabilities to perform Baseband Unit Pool Planning to align the capacities [36] and/or by appropriate admission control. Although exploiting jointly network slicing and capacity transferring capabilities is an interesting problem, it is out of the scope of this study.

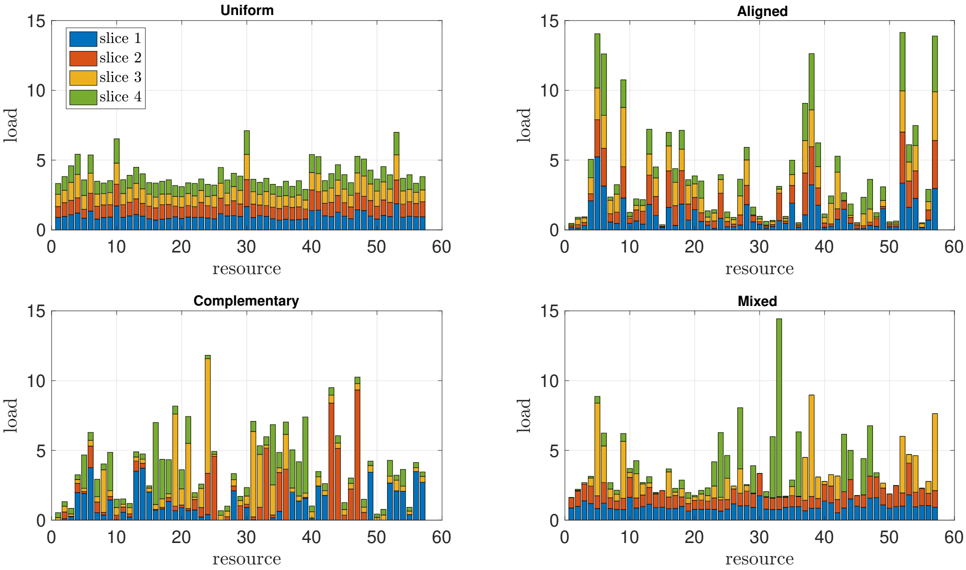

Fig. 2. Load distribution of the illustrative scenarios.

mean traffic loads, then $\mathrm{M}(\mathcal{P}) = 0$ and the expected network utility is given by

$$U^{(\beta)}(\mathcal{P}) = \log\left(\frac{c}{\rho}\right) + \mathrm{D}(\mathcal{P}) - \frac{1}{\beta}\,\mathrm{S}\,(\mathcal{P}) + o\left(\frac{1}{\beta}\right).$$

Fact 2. *The term* $\mathrm{D}(\mathcal{P}) = \mathrm{D}_{KL}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right)$ *is maximized when* $\mathcal{P} = \mathcal{P}^{GPS}$*. Therefore, for large* β*, an upper bound on the utility is given by* $\bar{U} = \log\left(c/\rho\right) + \mathrm{D}_{KL}\left(\hat{\mathbf{s}}(\mathcal{P}^{GPS})||\hat{\boldsymbol{\rho}}(\mathcal{P}^{GPS})\right)$.

The general expression for the stochastic pooling losses is complex and it is hard to obtain insight and further closed-form expressions. Some properties of $\mathrm{S}(\mathcal{P})$ are presented next.

Fact 3. *The term* $\mathrm{S}(\mathcal{P}) = \langle \hat{\mathbf{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P})\rangle + \langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P})\rangle$*, where the inner product* $\langle \hat{\mathbf{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P})\rangle$ *is maximized when* $\mathcal{P} = \mathcal{P}^{GPS}$ *and* $\langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P})\rangle = 0$ *when* $\mathcal{P} = \mathcal{P}^{GPS}$.

Given the properties detailed in Fact 3, it is intuitive that the stochastic pooling losses are reduced as the cardinality of the partition grows, i.e., as virtual pools aggregate resources, resulting in statistical multiplexing gains.

To conclude, we summarize our main observations next.

Remark 3. The optimal partition is dependent on the capacity, loads and shares distribution as well as on the variability in the number of active users and it is the result of a tradeoff between differentiation and statistical multiplexing. On the one hand, creating large VRPs achieves better statistical multiplexing but on the other hand creating small VRPs preserves the ability to differentiate slice performance. Therefore, a partition that includes virtual pools with similar load and share profiles is most beneficial, since it allows slices to reap the benefits of statistical multiplexing through sharing without compromising their ability to differentiate.

V. Performance evaluation

We have conducted a set of simulations to emulate a cellular network following the IMT Advanced evaluation guidelines for dense 'small cell' deployments [1]. The network is composed by 57 resources with identical capacities, disposed in a hexagonal cell grid layout with an intersite distance of 200 meters and shared among four slices. Unless otherwise specified, shares are configured to be uniform and equal to $s_b^o = 1/4$. A fixed set of users move around the network region, by combining users following two mobility models: (i) Random Waypoint model (RWP) which generates almost uniform distributions of mobile users over the network [7] and (ii) SLAW model [25], a human walk based mobility model which generates space uneven load distributions. A combination of both models generates uneven load distributions across resources. We explored 4 different scenarios of 4 slices described next and for which the average load distributions per resource are displayed in Figure 2 for the case of $L = 4$:

1) **Uniform**: homogeneous slices with uniform spatial loads.
2) **Aligned**: homogeneous slices with non-uniform loads.
3) **Complementary**: heterogenous slices with orthogonal non-uniform loads.
4) **Mixed**: 2 heterogenous slices with complementary non-uniform spatial loads and 2 with uniform spatial loads.

In Figure 3, we display the capacity savings of the optimal VRP partition versus GPS and CP for different scenarios. As can be seen, the gains over GPS are maximized in the scenarios where the mean load distributions across slices do not coincide, i.e., in the complementary and mixed where the gains can go up to 50%. With respect to CP, the capacity savings (except for the uniform) are very high since creating a big partition with all the resources eliminates the ability of slices to differentiate, resulting in resource allocations which exploit statistical multiplexing but are not able differentiate to slices' users performance. The reader is referred to [10] for an extended set of simulations and numerical evaluations.

VI. Conclusions

We have addressed the problem of finding a good compromise between (i) allowing tenants to shift resources across base stations, thus providing more flexible allocations, and (ii) protecting individual tenants from others' customer loads. The adopted solution finds a tradeoff by the shifting of allocations

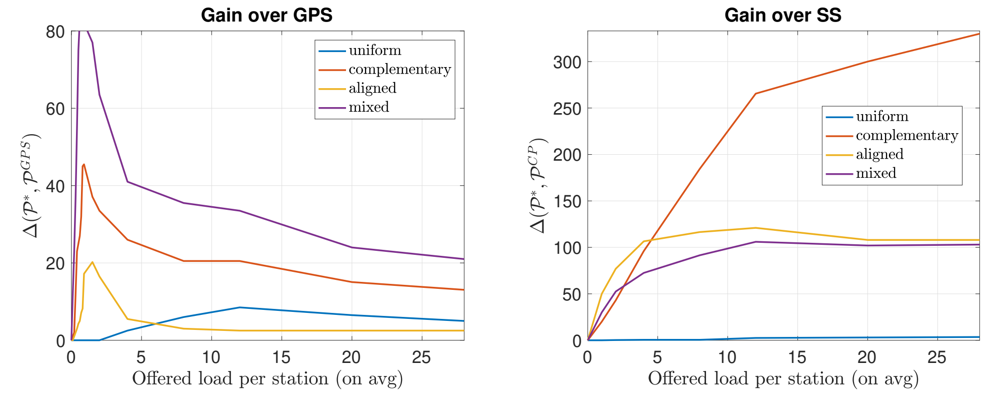

Fig. 3. Capacity savings for the different scenarios vs GPS and CP for uniform shares as a function of the mean offered load.

to sets of resources (VRPs) that are chosen to ensure slices are protected. Our optimal VRP partitioning problem creates jointly managed resource pools so as to optimize overall expected network utility while enabling network slice performance differentiation and isolation. Our results indicate that pooling resources on which slices have similar shares and load distributions is beneficial since it does not harm slices ability to differentiate their performance while allowing the pool to reap benefits from statistical multiplexing. Moreover, the analytical and numerical evaluations demonstrate that adequate partitioning provides substantial capacity savings as compared to GPS per resource sharing.

References

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

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

[3] R. Agrawal, A. Bedekar, R.J. La, and V. Subramanian. Class and channel condition based weighted proportional fair scheduler. In *Teletraffic Science and Engineering*, volume 4, pages 553–567. Elsevier, 2001.

[4] S. A. AlQahtani. Adaptive rate scheduling for 3g networks with shared resources using the generalized processor sharing performance model. *Computer Communications*, 31(1):103–111, 2008.

[5] A. Banchs. User fair queuing: fair allocation of bandwidth for users. In *Proc. of IEEE INFOCOM*, Mar. 2002.

[6] E. T. Bell. The iterated exponential integers. *Annals of Mathematics*, 39(3):539–557, 1938.

[7] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–567, 2004.

[8] T. Bu, Li Li, and R. Ramjee. Generalized Proportional Fair Scheduling in Third Generation Wireless Data Networks. In *Proc. of IEEE INFOCOM*, April 2006.

[9] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Prez. Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads. *IEEE/ACM Transactions on Networking*, 25(5), Oct 2017.

[10] P. Caballero, G. de Veciana, A. Banchs, and X. Costa-Perez. Optimizing Network Slicing via Virtual Resource Pool Partitioning (Extended). https://www.dropbox.com/s/hso0yqupo5wfwwl/Extended_VRPP.pdf.

[11] S.H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical models and Methods in Applied Sciences*, 1(4):300–307, 2007.

[12] China Mobile White paper. C-RAN The Road Towards Green RAN. 2011.

[13] T.M. Cover and J.A. Thomas. *Elements of information theory 2nd edition*. Wiley Series in Telecommunications and Signal Processing. Wiley-interscience, 2 edition, July 2006.

[14] D. Nicoara et al. Hermes: Dynamic partitioning for distributed social network graph databases. In *EDBT*, pages 25–36, 2015.

[15] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[16] A. Gudipati, L. Li, and S. Katti. RadioVisor: A Slicing Plane for Radio Access Networks. In *Proc. of HotSDN*, Aug. 2014.

[17] V. Gupta, M. H. Balter, K. Sigman, and W. Whitt. Analysis of join-the-shortest-queue routing for web server farms. *Performance Evaluation*, 64(9-12):1062–1081, 2007.

[18] P. A. Jensen. Optimum network partitioning. *Operations Research*, 19(4):916–932, 1971.

[19] J.Qadir, A. Sathiaseelan, L. Wang, and J. Crowcroft. Resource Pooling for Wireless Networks: Solutions for the Developing World. *ACM SIGCOMM Computer Communication Review*, 46(4):30–35, 2016.

[20] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research*, 49:237–252, Mar 1998.

[21] F. P. Kelly and R. J. Williams. Fluid model for a network operating under a fair bandwidth-sharing policy. *Ann. Appl. Probab.*, 14(3):1055–1083, 2004.

[22] A. Krause and D. Golovin. Submodular function maximization.

[23] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

[24] CN. Laws. Resource pooling in queueing networks with dynamic routing. *Advances in Applied Probability*, 24(3):699–726, 1992.

[25] K. Lee et al. SLAW: self-similar least-action human walk. *IEEE/ACM Transactions on Networking*, 20(2):515–529, April 2012.

[26] Y. L. Lee, J. Loo, T. C. Chuah, and L. Wang. Dynamic network slicing for multitenant heterogeneous cloud radio access networks. *IEEE Transactions on Wireless Communications*, 17(4):2146–2161, April 2018.

[27] L. Li, M. Pal, and R. Yang. Proportional fairness in multi-rate wireless LANs. In *Proc. of IEEE INFOCOM*, April 2008.

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

[29] M. Neely, E. Modiano, and C. Rohrs. Packet routing over parallel time-varying queues with application to satellite and wireless networks. In *Proc. of ALLERTON Conf.*, volume 39, pages 1110–1111. The University; 1998, 2001.

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

[31] A. K. Parekh. *A Generalized Processor Sharing Approach to Flow Control In Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, 1992.

[32] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho. Resource slicing in virtual wireless networks: A survey. *IEEE Transactions on Network and Service Management*, 13(3):462–476, 2016.

[33] K. Samdanis and A. H. Aghvami. Load balancing through dynamic partitioning for hierarchical cellular networks. In *2008 International Conference on Telecommunications*, Jun 2008.

[34] L. Wang and A. Chen. Optimal radio resource partition for joint contention-and connection-oriented multichannel access in ofdma systems. *IEEE Transactions on Mobile Computing*, 8(2):162–172, 2009.

[35] D. Wischik, M. Handley, and M.B. Braun. The resource pooling principle. *SIGCOMM Comput. Commun. Rev.*, 38(5):47–52, 2008.

[36] S. Xu and S. Wang. Baseband unit pool planning for cloud radio access networks: An approximation algorithm. *IEEE Communications Letters*, 21(2):358–361, Feb 2017.

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