The Dissertation Committee for Pablo Caballero Garces
certifies that this is the approved version of the following dissertation:

Design and Performance of Resource Allocation Mechanisms for Network Slicing

Committee:

Gustavo de Veciana, Supervisor

Albert Banchs Roca, Supervisor

Jeffrey G. Andrews

Francois Baccelli

Sanjay Shakkottai

John J. Hasenbein

Design and Performance of Resource Allocation Mechanisms for Network Slicing

by

Pablo Caballero Garces.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2018

Dedicated to my parents, my brother and Ghadi.

Acknowledgments

I would like to express my gratitude to my PhD supervisors Prof. Gustavo de Veciana and Prof. Albert Banchs. Their masterful and inspiring tutoring have deeply contributed to my growth, not only professionally but also personally. They helped and supported me during this fantastic adventure and have always stood by my side in the difficult moments. Your patience, encouragement and extraordinary guidance set an example that I would carry on during the rest of my life.

I would also like to thank Dr. Xavier Perex-Costa, Prof. Jeffrey Andrews, Prof. Francois Baccelli, Prof. Sanjay Shakkottai, Prof. John Hasenbein and Prof. Evdokia Nikolova for their time and valuable comments on this dissertation that undoubtedly improved my approach and the final result.

I want also to thank my lab-mates that helped and accompanied me during these intense years: Arjun, Jiaxiao, Yicong, Saadallah, Jean, Yuhuan, Ambika, Rebal, Philippe, Christian, Patricia and many others. Thanks to my friend Dr. Evgenia Christoforou, you helped me greatly at all times when it was not easy. Also to my friends Enol, Sofia, Rober, Maria, MJ for showing me that real friendship endures through distance.

Finally, I will always be indebted to my family: mom, dad, Enrique and Ghadi, you are the most important pillar in my life and nothing that I achieved I could have done without you. This milestone is as yours as mine. Also, I am

thankful for the rest of my wonderful family: grandpas, Manoli, Charo, Ernesto, Pedro, Gregorio Pablo, Carlos, Carlota, Pedro; thanks for always being there and for your love. I would like to dedicate this work to my uncle Gregorio, I hope this dissertation validate for that field report in Valle and that you are proud of all of us from up there.

Design and Performance of Resource Allocation Mechanisms for Network Slicing

Publication No. _____

Pablo Caballero Garces, Ph.D.
The University of Texas at Austin, 2018

Supervisors: Gustavo de Veciana
Albert Banchs Roca

Next generation wireless networks are expected to handle an exponential increase in demand for capacity generated by a collection of tenants and/or services with heterogeneous requirements. Multi-tenant network sharing, enabled through virtualization and network slicing, offers the opportunity to reduce operational and deployment costs, and the challenge of managing resource allocations among multiple tenants serving possibly mobile diverse customers. When designing shared radio resource allocation mechanisms, it is as important to provide tenants with customization and isolation guarantees, as it is to achieve high resource utilization and to do so via low complexity and easy to implement algorithms. This thesis is devoted to the design and analysis of resource allocation mechanisms that meet these objectives.

We propose a sharing model in which tenants are assigned a share/budget of a pool of network resources. This share is then redistributed in the form of weights

amongst users, which in turn drive dynamic resource allocations which are partially able to adapt to the traffic demands on, and requirements of, different slices customer populations. We propose and analyze two approaches for redistributing slices' share among customers which we classify into their associated (i) cooperative, and (ii) competitive resource allocations.

In the cooperative resource allocation setting, a pre-established policy is proposed, in which resources are eventually assigned in proportion to the slice's share and the relative number of active users in currently has at a resource. This is shown to be socially optimal in a particular setting and simple to implement, with statistical multiplexing gains that increase with the number of tenants and the size of the resource pool. These gains stem from the ability of the scheme to adapt to dynamic loads leading to an up to 50% network capacity savings with respect to static allocations. We further improve these gains by presenting a framework that combines resource allocation and wireless user association which uses limited computational, information, and handoff overheads. However, using our cooperative scheme over a large pool of resources restricts the degree to which a slice can differentiate its customers' performance at a per resource level. Thus, we study how this trade-off affects the network utility and propose a mechanism to determine an optimal partition the resources into a collection of self-managed pools under cooperative resource allocations.

Our competitive resource allocation approach enables tenants to reap the performance benefits of sharing while retaining the ability to customize their own users' allocations. This setting results in a network slicing game in which each ten-

ant reacts to the user allocations of the others so as to maximize its own customers' utility. We show that, under appropriate conditions, the game associated with such strategic behavior converges to a Nash equilibrium. At the Nash equilibrium, a tenant always achieves the same, or better, utility than it could achieve under a static partitioning of resources, hence providing the same level of inter-slice protection as static resource partitioning. The network utility of the equilibrium allocations is shown to be, under mild conditions, close to the socially optimal ones. The competitive resource allocation framework is complemented with a study on admission control policies that enable tenants to ensure minimum rate guarantees to their users. Our analysis and extensive simulation results confirm that our framework provides a comprehensive practical solution towards multi-tenant network slicing. We also discuss how our theoretical results fill a gap in the general resource allocation literature for strategic players.

Table of Contents

List of Tables

List of Figures

Chapter 1

Resource Allocation for Network Slicing

1.1 The origin of network sharing

Wireless networks play a key role in today's infrastructure through enabling communication and information sharing. Next generation wireless networks are expected to handle an exponential increase in demand for capacity from a collection of services with heterogeneous requirements, such as high data rates, very low latency and massive connectivity. To support the high volume of demand and heterogeneity of services, mobile networks are expected to use denser station deployments, network virtualization and unexploited portions of the spectrum, such as millimeter Wave (mmWave) high-frequency bands [14]. This level of sophistication and densification result in high infrastructure operation and deployment costs.

Using the traditional single ownership infrastructure model, Mobile Network Operators (MNOs) are finding difficulties to justify the required investment in new network infrastructure technologies deployment. Over-The-Top (OTT) service providers obtain the lion's share of revenues derived from wireless connectivity without incurring any wireless infrastructure cost and the net neutrality principle - whereby MNOs are forced to treat packets equally - impedes MNOs from generating new revenue streams from OTTs. From an OTT perspective, current

infrastructure lacks on service differentiation control - which is especially critical since some of these new services are becoming increasingly complex to manage and critically rely on the infrastructure to meet their requirements, as for example, the Ultra-Reliable Low Latency Communications (URLLC) required for vehicular communications.

This situation fueled the need for improved approaches to network sharing. Mobile network infrastructure sharing provides a mechanism: (i) to share the operational (OPEX) and capital (CAPEX) expenditures, (ii) to increase resource utilization via statistical multiplexing of heterogeneous and bursty traffic loads, and (iii), to provide service providers with operational capabilities to differentiate their services. In the rest of this thesis, we will use the term multi-tenant network to denote a network shared by multiple entities, e.g., MNOs, MVNOs, and/or by multiple OTT services belonging to these or other entities.

1.2 Who shares network resources?

Network sharing is thus as a key business and technological requirement, where traditional and virtual operators, along with service providers (e.g. OTTs) share wireless networks. The diversity of tenants opens the possibility for multiple and flexible use case scenarios, as a function, for example, of who owns the costly wireless spectrum and the type of tenants. Several possible use cases for multi-tenant networks are described below:

(a) Multiple traditional MNOs pool their licensed spectrum and share a common

pool of resources through Radio Access Network Sharing to obtain mobile operational capabilities, using a shared or private core network.

(b) One or multiple MNOs rent a share of their owned licensed spectrum and/or infrastructure to service providers or Mobile Virtual Network Operators (MVNO), thus, offering them mobile network operational capabilities.

(c) One or multiple tenants deploy mobile network infrastructure to share unlicensed spectrum or the "shared spectrum" band. Federal Communications Commission (FCC) has been promoting the use by commercial entities of (i) unlicensed spectrum in the 2.4GHz and 5GHz bands devices [28], by the so-called LTE-U and/or (ii) the 3.5 GHz "shared spectrum" band (or the Citizen's Broadband Radio Service) [29].

(d) A wholesale company or government (e.g., [88]) deploys mobile network infrastructure using (i) their own licensed spectrum, (ii) unlicensed spectrum or "shared spectrum" or (iii) MNOs purchased proprietary licensed spectrum. The owner itself absorbs the infrastructure CAPEX and OPEX costs and resells mobile network operational capabilities to MNOs and/or service providers.

1.3 What resources can be shared?

To bring network sharing to fruition, network infrastructure sharing can be realized in multiple ways, depending on the network elements to be shared. As classified in [36], network sharing is traditionally divided into *roaming*, *passive* sharing, and *active* sharing.

- *Roaming* is defined as an agreement that enables customers of a provider, which does not have coverage in a certain area, to connect to the network of another provider.

- *Passive sharing* refers to the sharing of physical components such as physical sites, tower masts, cabling, cabinets, power supply, air-conditioning, etc.

- *Active sharing* refers to the sharing of (i) the *Transport Network*, i.e., backhaul; (ii) the *Core Network* and/or (iii) the *Radio Access Network* (RAN).

As forecasted in [27], the operational cost savings from network sharing can be as high as $60 - 80\%$. A detailed forecast on CAPEX/OPEX savings per sharing type is displayed in Table 1.1.

	Passive Sharing	**Active Sharing**		
		Transport Network	Core Network	RAN
CAPEX	10%	10-20%	15-30%	25-40%
OPEX	15%	10-15%	20-25%	20-30%

Table 1.1: CAPEX/OPEX savings forecasts [27].

The biggest portion of these savings are expected to come from active **RAN sharing** which will be the focus of this thesis.

1.4 How should network resources be shared?

Many of the tenants that will want to share resources will serve a spatially distributed, possibly mobile, population of customers. Thus, a tenant will require a

collection of spatially distributed RAN resources to meet their needs. In this thesis, we propose several **sharing criteria** to effectively share a collection of resources. This section provides a high-level perspective on the architectures, vision and objectives underlying our approach to RAN Sharing.

1.4.1 Architectural enablers and network slicing

The 3GPP Standardization body proposed in [6] two main functional architectures for active RAN sharing:

- Multi-Operator Core Network (**MOCN**): Under this architecture, each network provider owns its dedicated Core Network, while the RAN is common.
- Gateway Core Network (**GWCN**): In this model, in addition to the RAN, some of the Core Network and Transport Networks elements and functionalities can be shared, such as the Mobility Management Entity (MME).

With the emergence of technologies such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) it is expected that networks will realize sharing modalities beyond MOCN and GWCN. One of the main concepts arising from network virtualization, and one of the main enablers for network sharing, is **network slicing**, a recognized key element in 5G mobile networks [8, 86].

The idea behind network slicing is to allow the physical mobile network infrastructure to be "sliced" into logical networks, where each logical network is a collection of resources and functions that are orchestrated to support a specific service. Each slice may contain software modules running at different locations as

well as computational resources and communication resources in the backhaul and radio network. The intention is to only provision a slice with what is needed for the service while avoiding unnecessary overheads and complexity. From a RAN perspective, the virtualization increases flexibility in realizing resource allocation mechanisms.

Assuming that network slicing is enabled and building on top of its capabilities, this thesis will focus on the task of achieving efficient resource allocation among participant slices; which is referred to as RAN Slicing. In the sequel, we will use the terms "slice", "tenant" and "operator" to refer to the sharing entities interchangeably.

1.4.2 Vision and objectives for RAN slicing

Our vision for RAN slicing is to engineer a resource allocation mechanism which enjoy similar features as those in today's cloud computing infrastructure. Cloud computing infrastructure uses resource pooling, which aims to make the computational resources to behave as if they conform a single pooled resource that can be elastically provisioned to adapt to the demand [75, 105]. The resource pooling can be achieved by a large centralized pool of equipment installed in a data centers, or by load balancing the demand across distributed server farms.

However, there is a fundamental difference engineering radio resource pooling for mobile networks. The set of resources reachable by a user is restricted due to physical radio propagation and changes dynamically given that users are mobile, preventing infrastructure planning to overcome this problem.

For parallel resource networks, as described in [56][1], different techniques can be used to leverage "*virtual*" resource pooling besides using centralized resources per se, such as for example:

(i) *load balancing* to control congestion, e.g., through multipath routing, user association and user admission control [63],

(ii) *dynamic resource allocation*, e.g., through Dynamic Spectrum Access [101] or network wide allocations [19, 111], and

(iii) *network sharing*, e.g, through multi-tenancy [21, 33],

(iv) *community sharing/ crowdsourcing*, e.g., through Device to Device Communications and WiFi offloading.

In this thesis, we aim to design a RAN slicing approach that enable virtual resource pooling through a combination of dynamic resource allocation, network sharing and load balancing techniques, although we put special emphasis on resource allocation and network sharing. In designing such approach among slices, we face multiple challenges for which the main design objectives (as envisioned in [30] and inspired by the features of cloud computing [75]) are described next:

- **Flexibility**: The approach should allow for flexible and *dynamic* sharing and resource allocation to meet operators' heterogeneous requirements and changes in demands across time and space.

[1]The reader is referred to [56] for a detailed taxonomy.

- **Customization**: Per slice allocations should be tailored to satisfy the particular demands of each slice, so the resource allocation mechanism should allow customization of the service, according to each tenant preferences, allowing them to differentiate their service from that of other slices.

- **Isolation**: Although it is desirable to allow a certain degree of customization, the resource allocation mechanism should be *fair* and the slices should be *protected*. Each tenant shall receive an amount of resources proportional or equivalent to their contribution and independent from the customization choices of other slices.

- **Implementability**: Designing mechanisms to implement this solution, while realizing timely adaptation to network changes, is also very challenging. Given the amount of information involved and its dynamic nature, the solutions should be as *distributed* as possible while being low in overhead and complexity and easy to deploy across heterogeneous resources.

1.4.3 Virtual pooling resource allocation mechanisms: cooperative vs competitive

We use the term virtual pool to denote a collection of geographically dispersed base stations resources to be shared by multiple tenants. Each slice is assigned a **share** (portion) of a virtual pool, which can be redistributed among its various customers in the form of weights. Then, at each station of the virtual pool, resources are distributed in proportion to the weights of its active users. This approach, in principle, allows resources to be allocated in a manner that tracks the

possible spatial variations in customer demands of tenants. Such a dynamic resource allocation among heterogeneous slices aims to enable the virtual pool to act as a centralized resource pool.

As mentioned, our approach is predicated on each slices' share being subdivided in weights assigned to its users, which in turn determine the resource allocation per station. In this thesis, we will consider two weight allocation mechanisms.

The first, referred as **cooperative resource allocation** assumes a predefined policy where resources are assigned in proportion to the share and the relative number of active users of a slice at each base station, i.e., that the weight of each user is equal to the share of its slice divided by the total number of active users of its slice. The choice to re-distribute a slice's share (budget) equally amongst its users, can be viewed as a network mandated policy, but, as shown in the first part of this thesis, also emerges naturally as the socially optimal weight allocation when slices exhibit (price taking) strategic behavior in optimizing their own utility.

The second weight allocation mechanism to be considered, hereafter called **competitive resource allocation** allows each slice to unilaterally customize its share allocation amongst users to maximize its own benefit according to its requirements. Such a competitive weight allocation increases the ability of a tenant to customize but may impact other objectives such as isolation and fairness. Such unilateral competitive weight allocation lead to a so-called network slicing game, which is studied in detail in the second part of this thesis.

1.5 Outline

The content of the remaining chapters of this thesis is organized in two parts, as displayed in the conceptual chart of Figure 1.1, each of them described next.

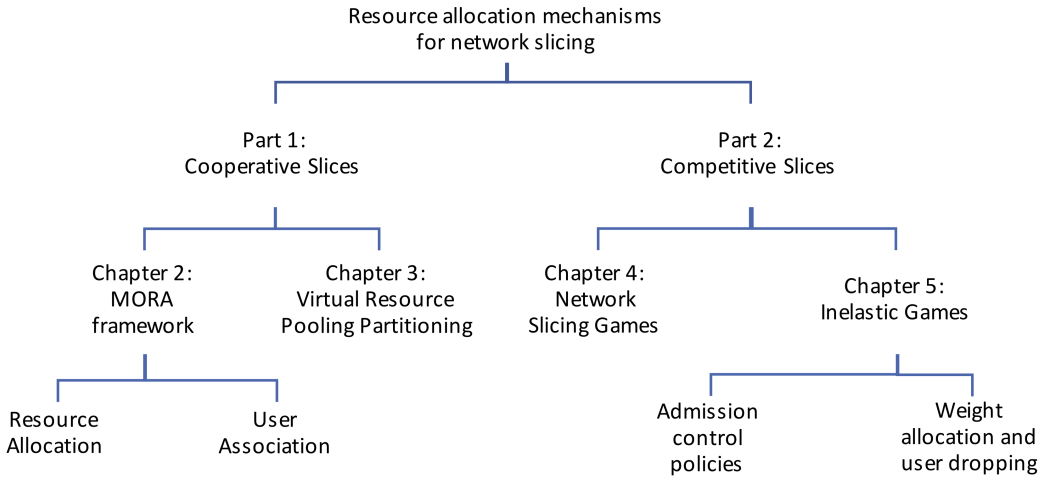

Figure 1.1: Thesis Outline Tree Chart

Part 1: The first part is devoted to study the case of centralized or cooperative resource allocation for RAN network slicing. **Chapter 2** presents a Multi-Operator Resource Allocation (MORA) based on a weighted proportionally fair global objective, which provably achieves desirable fairness/protection across the network slices of the different tenants and their associated users. Also, a user association algorithm is proposed and the performance of the system is evaluated through performance analysis and simulations. **Chapter 3** discusses how to partition the network stations to create sets of virtual pools that maximize statistical multiplexing and slice differentiation gains while maintaining service isolation among slices, and proposes an algorithm to determine such partitions.

Part 2: By contrast, the second part of this thesis analyzes the case where slices are competitive and take unilateral weight allocation decisions to maximize their own benefit. Since the decisions are now decentralized, slices' interactions can be viewed as a game. This network slicing game is described and analyzed in **Chapter 4** where we show the existence of, and convergence to, a Nash Equilibrium. Our analytical results show that the price of anarchy associated with measuring the loss in performance due to competition is bounded and the allocations are envy-free. Simulation results comparing the performance of cooperative and competitive scenarios are presented and confirm our analytical results. In **Chapter 5**, we expand this framework to consider slices that support inelastic users, i.e., users with *minimum rate requirements*. This generates the necessity for admission control and user dropping mechanisms that ensure the minimum rate requirements are met. Several admission control mechanisms are presented and their impact on the blocking probability and rate distribution across slices is evaluated through simulation.

Finally, in **Chapter 6**, we conclude by highlighting the main results and insights of the research conducted along with suggestions for future work to expand this thesis.

1.6 Publications

Below is a list of conference and journal publications I have co-authored in the context of this research topic.

1. "Network Slicing Games for Guaranteed Rate Services", **P. Caballero**, A. Banchs, G. deVeciana, X. C. Perez, A. Azcorra, *IEEE Transactions on Wireless Communications*, [Accepted, to appear].

2. "Statistical Multiplexing and Traffic Shaping Games for Network Slicing", J. Zheng, **P. Caballero**, G. deVeciana, A. Banchs, *ACM/IEEE Transactions on Networking* [Under review].

3. "Statistical Multiplexing and Traffic Shaping Games for Network Slicing", J. Zheng, **P. Caballero**, G. deVeciana, A. Banchs, *IEEE WiOPT*, 2017.

4. "Network Slicing Games: Enabling Customization in Multi-Tenant Networks"; **P. Caballero**, A. Banchs, G. deVeciana, X. C. Perez, *ACM/IEEE Transactions on Networking* [Under review].

5. "Network Slicing Games: Enabling Customization in Multi-Tenant Networks"; **P. Caballero**, A. Banchs, G. deVeciana, X. C. Perez, *IEEE INFOCOM*, 2017.

6. "Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads"; **P. Caballero**, A. Banchs, G. deVeciana, X. C. Perez, *ACM/IEEE Transactions on Networking*, 2017.

7. "RMSC: A Cell Slicing Controller for Virtualized Multi-Tenant Mobile Networks"; **P. Caballero**, X. C. Perez, K. Samdanis and A. Banchs, *IEEE Vehicular Technology Conference VTC*, 2015.

Part I

Cooperative Resource Allocation

Chapter 2

Multi-Tenant Radio Access Network Slicing

This chapter[1] proposes a complete solution for multi-tenant network slicing along with an algorithm to allocate resources accordingly. In this solution, we assume slices to be cooperative and so resource allocation decisions are taken by the network infrastructure management entities, with the aim to optimize the global network utilization through a global network utility function. The main goal is to analyze whether our model provides to the participant tenants gains with respect to scenarios where each tenant privately owns network infrastructure (defined as Static Slicing Benchmark).

We formalize the "Share Constrained Dynamic Network Slicing" formulation for this particular scenario and we propose a criterion, based on share constrained proportional fairness, to solve the problem. While similar criteria has been proposed before, we provide a characterization supporting its use in a multi-tenant network setting, by (i) presenting a set of desirable properties and (ii) analyzing the resulting performance benefits. These properties provide insights on the optimality and fairness of the resulting allocations, and the benefits are studied by characteriz-

[1]Publications based on this chapter: P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez. Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads. IEEE/ACM Transactions on Networking, 25(5), Oct 2017. All co-authors contributed equally.

ing the capacity savings by means of a closed-formula. We show that the criterion not only improves overall network utility but also that of each individual slice, thus guaranteeing that slices are not harmed by the sharing of resources amongst slices.

As shown, the proposed criterion corresponds to an NP-hard problem, motivating the need to devise an efficient approximation algorithm which is engineered in this chapter. The proposed algorithm is semi-online, distributed, incurs low computational complexity, and has been specifically designed to control overheads associated with handoffs and/or mobile user reassociations. We rely on several intermediate analytical results to drive the key design choices underlying our algorithm. One of these intermediate results is a variation of the algorithm which achieves similar performance bounds to state-of-the art algorithms while being distributed – which is in itself a valuable theoretical contribution on state-of-the-art approaches.

Lastly, a comprehensive performance evaluation based on detailed simulations is provided, showing that (i) slices can save up to 80% capacity while providing the same quality to their users, and (ii) for a fixed capacity, we improve user performance in terms of file download times by up to 30%, among other results.

2.1 Related work

We next review and contrast our work to the state-of-the-art in (i) resource allocation based on proportional fairness, and (ii) resource sharing among operators.

Considerable research effort has been devoted to address the problem of

fair resource allocation in networks. In wireline networks, fair resource allocation based on utility function maximization has been extensively studied following the seminal work of [58]. Building on this work, further algorithms for congestion control in multi-path environments have been proposed [41, 61]. Not unlike our work, these algorithms are distributed. However, they allow users to decide among multiple routes while we focus on a wireless setting where each user can only use one resource (her base station).

In the specific context of wireless networks, several approaches have been proposed [15, 19, 74] to the problem of resource allocation and user association based on weighted and unweighted proportional fairness, respectively. The unweighted case has been largely studied in the literature in different contexts (e.g., power control [70], interference avoidance [100]). The authors of [19] and [100] analyzed the complexity of the problem and proved the existence of polynomial time algorithms which provide an exact solution, and [15] designed a distributed algorithm with convergence guarantees. In contrast to the above, the resource allocation criterion proposed in this chapter relies on *weighted* proportional fairness, with operator-specific weights; this is a more difficult problem as it is NP-hard [19] and the convergence of distributed greedy algorithms cannot be guaranteed [78].

Weighted proportional fair resource allocation in the context of wireless networks has also been studied from different angles. In [74], an algorithm with tight worst-case performance bounds is proposed, while [112] proposes an heuristic algorithm. In contrast to the distributed approach proposed in this chapter, both algorithms are centralized and require the availability of the full network state infor-

mation, which may be very challenging to gather in a timely manner. The authors of [47, 48] propose a Gibbs-sampling mechanism based on simulated annealing that converges to an optimal solution. However, the convergence of such mechanisms is known to be very slow and for this reason the authors resort to a more practical greedy solution. For the proposed greedy solution, the authors neither provide performance bounds nor analyze convergence; additionally, the overhead is not controlled, which limits their practical deployment. All the approaches mentioned above address the problem of a single-operator network, in contrast to our work which focuses on the slicing and sharing of resources among multiple operators.

Multi-operator network sharing has been studied from many different angles, including planning, economics, coverage, performance, etc. (see e.g. [35, 73, 89]). This chapter focuses specifically on the design of algorithms for resource sharing among operators, which has been previously addressed by [20, 22, 43, 76, 77]; however, all these works differ substantially from ours in terms of scope, criterion or approach. In [22, 43], the optimization of the network utility follows a different criterion from the one in this chapter, weighted proportional fairness, which (as we show) provides many desirable properties. The works of [76, 77] present a proportional fair formulation similar to ours; however, they do not provide a rationale for their choice, in contrast to the solid analytical arguments provided here. Furthermore, [77] does not address the algorithm design, while [76] uses a general non-linear solver that incurs a very high computational complexity (as confirmed by our results of Section 2.7.4). Finally, [20] follows a game theoretic approach

where operators bid for resources, which results in a fundamentally different problem from the one addressed here.

In summary: (i) while there has been substantial research on proportional fair resource allocation, its application to multi-operator settings and the associated problems have not been studied, and (ii) in spite of the substantial work devoted to proportional fairness in general settings, there is a gap in the systematic study of distributed mechanisms for joint resource allocation and user association that build on analytical results.

2.2 Chapter organization

The organization of the rest of this chapter is as follows. In Section 2.3, we introduce a criterion for dynamic resource sharing among operators; while the criterion has been proposed before, we provide a characterization supporting its use in a multi-tenant network setting. These properties are developed in Section 2.4.1, providing insights on the optimality and fairness of the resulting allocations, and the benefits are studied in Section 2.5, by characterizing the capacity savings by means of a closed-formula. We show that the criterion not only improves overall network utility but also that of each individual operator, thus guaranteeing that operators are not harmed by the sharing of resources amongst slices. In Section 2.6.1, we show the criterion corresponds to an NP-hard problem, motivating the need to devise an efficient approximation algorithm which is introduced in Section 2.6.2. The proposed algorithm is semi-online, distributed, incurs low computational complexity, and has been specifically designed to control overheads associated with handoffs

and/or mobile user reassociations; we rely on several intermediate analytical results to drive the key design choices underlying our algorithm. Section 2.7 provides a comprehensive performance evaluation based on detailed simulations, showing that (i) operators can save up to 80% capacity while providing the same quality to their users, and (ii) for a fixed capacity, we improve user performance in terms of file download times by up to 30%, among other results. The proof of the theoretical results for this chapter are provided in Section 2.9.

2.3 System model

We start by presenting our system model which was developed with LTE/ LTE-A systems in mind, but is generally applicable to cellular systems. We consider a network consisting of a set $\mathcal{B}$ of base stations (or sectors in case of sector antennas) that are shared by a set of operators $\mathcal{O}$. At any given time, we let $\mathcal{U}$ denote the set of users sharing the network and $\mathcal{U}_o,\ o \in \mathcal{O}$ the subsets of users belonging to each operator. An allocation of resources involves two sets of variables: (i) the association of users to base stations, denoted by $\mathbf{x} = (x_{ub} : u \in \mathcal{U}, b \in \mathcal{B})$, where each user u is associated with a single base station, i.e., $x_{ub} = 1$ for one of the base stations and 0 otherwise, and (ii) the allocation of the resources of each base station among its associated users, denoted by $\mathbf{f} = (f_{ub} : u \in \mathcal{U}, b \in \mathcal{B})$, where f_{ub} is the fraction of the base station b's resources which are allocated to user u.[2] Note that in our model we ignoring the discrete nature of such resources, and assume that $f_{u,b}$

[2]For instance, in LTE/LTE-A f_{ub} denotes the fraction of physical Resource Blocks, in FDM the fraction of bandwidth and in TDM the fraction of time

can take any value in the continuous range [0,1].

We let $\tilde{c}_{ub}$ denote the average rate per resource unit seen by user u at base station b under current radio conditions,[3] and let $\mathcal{C}_b$ be the base station's total amount resources. Given that the user is allocated a fraction f_{ub} of the base station's resources, her rate is given by $f_{ub}\mathcal{C}_b\tilde{c}_{ub}$. For notational convenience, we define the achievable rate of the user as $c_{ub} := \tilde{c}_{ub}\mathcal{C}_b$, which yields the following rate allocation:

$$r_u(\mathbf{x}, \mathbf{f}) := \sum_{b \in \mathcal{B}} x_{ub} f_{ub} c_{ub}.$$

Note that the definition of c_{ub} actually represents an abstraction of the underlying physical resources, accounting for the various physical layer techniques (such as, e.g., power control or MU-MIMO) as well as the interference from different sources (including that of neighboring base stations). In line with similar analyses in the literature [43, 76, 77, 97, 108, 109], we shall assume that c_{ub} is fixed for each user and base station pair.

2.4 MORA criterion

In this section, we formulate the optimization problem that will drive (i) the association of users to base stations, and (ii) the allocation of base stations' resources to users. Hereafter, we refer to this optimization as the *multi-operator resource allocation* (MORA) criterion. We show analytically that the criterion sat-

[3] Note that such average rates depend on the choice of modulation and coding scheme(s) selected for the user, after averaging out short-term fluctuations.

isfies desirable properties in terms of optimality and fairness, and develop a simple model to evaluate the potential sharing gains of our network slicing approach.

In line with previous approaches [43, 76, 77], the underlying assumption behind our criterion is that operators share the cost of deploying and/or maintaining the infrastructure, and the resources received by each operator should be based on the level of its (financial) contribution to the shared network: if an operator contributes twice as much as another, it should roughly get twice the resources. To this end, each operator is assigned a *network share* $s_o \in [0, 1]$, to represent its level of contribution to the network. Without loss of generality, these shares are normalized so that $\sum_{o \in \mathcal{O}} s_o = 1$.

The proposed criterion allocates resources across operators dynamically, tracking changes in the numbers and locations of operators' mobile users and the associated transmission rates c_{ub}. When doing this, we need to make sure that (i) network resources are fairly shared among the various operators according to their share, and (ii) at the same time, the resources allocated to a given operator are fairly shared among the users of that operator. To achieve this, we follow an approach akin to that in [16][4]: we maximize the overall network utility resulting from aggregating operator utilities, where the utility of an operator is in turn the aggregation of its users' utilities. To this end, we define the overall network utility as the

[4]Reference [16] addresses a similar problem to ours in the context of users and flows, as it aims at allocating resources fairly to users while preserving fairness among the flows of each user.

sum of operators' utilities weighted by the shares,

$$W(\mathbf{x}, \mathbf{f}) = \sum_{o \in \mathcal{O}} s_o \, U_o(\mathbf{x}, \mathbf{f}),$$

and the *operator utility* as the sum utility of the operator's users normalized by the number of users (where a user's utility is logarithmic in its rate),

$$U_o(\mathbf{x}, \mathbf{f}) = \frac{1}{|\mathcal{U}_o|} \sum_{u \in \mathcal{U}_o} \log(r_u(\mathbf{x}, \mathbf{f})),$$

By weighting the operator utilities with the shares, we give higher priority to operators with larger shares, and by normalizing with the number of users, we avoid that operators with more users are better off. For instance, with this choice, under uniformly loaded base stations an operator with twice the share of another one will get twice as many resources, independent of the number of users of each. Combining the above equations, one can rewrite the network utility as follows:

$$W(\mathbf{x}, \mathbf{f}) = \sum_{o \in \mathcal{O}} \sum_{u \in \mathcal{U}_o} w_u \log(r_u(\mathbf{x}, \mathbf{f})), \tag{2.1}$$

where the user weights w_u are defined as the operator network share divided by the current number of users of the operator, i.e., $w_u = s_o/|\mathcal{U}_o|$ (in simple terms, the network share of an operator is divided equally amongst its current users).[5]

With the above, we can now formulate the MORA optimization problem as follows. Such optimization corresponds to the weighted *proportional fair* criterion

[5]While our definition of network utility coincides with that for *weighted proportional fairness*, the criterion proposed here is fundamentally different: we consider resource allocation across time and vary the weights with the number of users, while *weighted proportional fairness* typically focuses on a static scenario and relies on fixed weights.

(see e.g. [58]) extended to a multi-operator setting that considers utilities of the operators, rather than the ones of the individual users:[6]

$$\max_{\mathbf{x},\mathbf{f}} \; W(\mathbf{x},\mathbf{f}), \tag{2.2a}$$

subject to:

$$r_u(\mathbf{x},\mathbf{f}) = \sum_{b\in\mathcal{B}} x_{ub} f_{ub} c_{ub}, \; \forall \, u \tag{2.2b}$$

$$\sum_{b\in\mathcal{B}} x_{ub} = 1 \text{ and } x_{ub} \in \{0,1\}, \; \forall \, b,u \tag{2.2c}$$

$$\sum_{u\in\mathcal{U}} f_{ub} x_{ub} \leq 1 \text{ and } f_{ub} \geq 0, \; \forall \, b,u. \tag{2.2d}$$

In the sequel we shall let $\mathbf{x}^{MORA}, \mathbf{f}^{MORA}$ denote a (possibly not unique) optimal solution to this optimization problem. This formulation provides the optimal resource allocation at a given time under the current c_{ub} values (given by the selected modulation-coding schemes); in a dynamic setting, such allocations would be re-evaluated when any of the c_{ub} values change, due to changes in the (average) channel quality.

Note that, once MORA returns the user association $\mathbf{x}$ and resource allocation $\mathbf{f}$, physical layer techniques (such as MU-MIMO or power control) are employed to optimize performance, under the constraint that users are provided with rates proportional to the r_u values given by MORA.

[6] Note that (2.2c) ensures that a user is associated with one (and only one) base station.

2.4.1 Properties of MORA resource allocation

Next, we show that the MORA criterion satisfies some desirable properties both in the way base stations' resources are allocated to associated users, and the way users are associated with base stations.

2.4.1.1 Per-base station resource allocation

Let us first consider a general setting, where user associations to base stations are *fixed*, to see how MORA allocates base station resources. Let $\mathbf{x}^*$ be the fixed (not necessarily optimal) user to base station association. If we optimize the resource allocation $\mathbf{f}$ for this user association, i.e., $\max_{\mathbf{f}} W(\mathbf{x}^*, \mathbf{f})$ subject to (2.2b) and (2.2c), it can be seen from Lemma 5.1 of [74] that the resulting resource allocation is unique and given by $\mathbf{f}^M(\mathbf{x}^*) = (f^M_{ub}(\mathbf{x}^*) : u \in \mathcal{U}, b \in \mathcal{B})$, where

$$f^M_{ub}(\mathbf{x}^*) = \frac{w_u x^*_{ub}}{\sum_{v \in \mathcal{U}} w_v x^*_{vb}}. \tag{2.3}$$

Further if $\mathbf{x}^* = \mathbf{x}^{MORA}$, then $\mathbf{f}^M(\mathbf{x}^*) = \mathbf{f}^{MORA}$, i.e., we have MORA optimal allocation of network resources.

The above result is fairly intuitive. Users associated with a given base station are allocated resources proportionally to their weights w_u. This can be viewed as follows. The share of an operator represents the total budget of the operator. When assigning a weight $w_u = s_o/|\mathcal{U}_o|$ to users, this share is distributed among the operator's users, and hence the user's weight represents the budget of a user. As the resources allocated to a user are inversely proportional to the sum of weights at her base station, the sum of weights can be viewed as the cost of a unit of resource

at the base station. Thus, operators with users associated with heavily loaded base stations will have to pay a higher cost (e.g, increase their network share or limit their overall number of users) or receive fewer resources.

The above shows that the number of active users that operators have on the network and their *spatial distribution* will impact the resources allocated under MORA. Indeed, allocations across base stations are coupled together through $|\mathcal{U}_o|$, i.e., an operator with a large number of active users will have lower weights and likely lower per-user allocations. At the same time, the resources obtained by an operator heavily depend on the *load* at base stations to which its users will be associated with.

2.4.1.2 User association

Next we study the MORA user associations. Building on the optimality of our formulation, we can show that the resource allocation resulting from MORA is Pareto-optimal, which means that for any alternative allocation $(\mathbf{x}', \mathbf{f}')$ for which $r_u(\mathbf{x}', \mathbf{f}') > r_u(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ for some u, we necessarily have $r_v(\mathbf{x}', \mathbf{f}') < r_v(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ for some $v \neq u$. Indeed, if this was not the case then $W(\mathbf{x}', \mathbf{f}')$ would be larger than $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$, which contradicts the fact that the optimal MORA allocation $(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ maximizes $W(\mathbf{x}, \mathbf{f})$.

Thus, Pareto optimality in this context means that if under some other user association choice, a user sees a higher throughput than that under MORA then there must be another user which sees a lower throughput allocation. Note that this need not always be the case. Consider, for instance, a network with $|\mathcal{U}|$ users, such

that the largest c_{ub} of each user corresponds to a different base station. While the optimal allocation would associate each user to the base station with largest c_{ub}, a criterion based on local decisions that looks at users one by one may lead to a different association. The above result guarantees that this will not happen under MORA.

2.5 Gains and Savings of MORA

In the following we evaluate the benefits of MORA. To that end, we introduce a simple baseline – *static slicing* (SS), a proxy for not sharing resources at all.[7]

2.5.1 Static Slicing (SS) baseline

Suppose each operator contracts for a *fixed* slice/fraction s_o of the network resources at each base station for its exclusive use. The operator can of course still optimize its users associations, $\mathbf{x}^o = (x_{ub} : u \in \mathcal{U}_o, b \in \mathcal{B})$, and allocation of resources $\mathbf{f}^o = (f_{ub} : u \in \mathcal{U}_o, b \in \mathcal{B})$, so as to maximize its utility. Specifically each operator $o \in \mathcal{O}$ can determine its user association and resource allocations

[7] By *slicing* we refer to the way resources are shared (or sliced) among operators (while *resource allocation* refers to the allocation of resources to specific users). In contrast to the dynamic nature of MORA-based slicing, static slicing divides the infrastructure in fixed fractions.

based on:

$$\begin{aligned} \max_{\mathbf{x^o},\mathbf{f^o}} \quad & U_o(\mathbf{x^o},\mathbf{f^o}) \\ \text{subject to} \quad & r_u(\mathbf{x^o},\mathbf{f^o}) = \sum_{b\in\mathcal{B}} x_{ub} f_{ub} c_{ub},\ \forall u \in \mathcal{U}_o, \\ & \sum_{b\in\mathcal{B}} x_{ub} = 1,\ \ \forall u \in \mathcal{U}_o, \\ & \sum_{u\in\mathcal{U}_o} f_{ub} x_{ub} \le s_o,\ \ \forall b \in \mathcal{B}, \\ & x_{ub} \in \{0,1\},\ f_{ub} \ge 0,\ \ \forall b \in \mathcal{B}, \forall u \in \mathcal{U}_o. \end{aligned} \tag{2.4}$$

This is similar to MORA except limited to the operator o's current users $\mathcal{U}_o$ and the resource constraint corresponds only to the fixed slice s_o allocated to the operator at each base station. Although the user associations and resource allocations under static slicing are independently optimized by each operator, we shall let $\mathbf{x}^{SS}, \mathbf{f}^{SS}$ be a (possibly not unique) optimal choice across all operators under static slicing. Also paralleling our discussion of MORA, it is easy to show that if one fixes a feasible user association $\mathbf{x}^*$, (2.4) is convex and yields resource allocations given by

$$\mathbf{f}^S(\mathbf{x}^*) := (f^*_{ub}(\mathbf{x}^*)\ :\ \forall u \in \mathcal{U}, \forall b \in \mathcal{B}),$$

where

$$f^*_{ub}(\mathbf{x}^*) = \frac{x^*_{ub} s_o}{\sum_{v\in\mathcal{U}_o} x^*_{vb}} \mathbf{1}\{u \in \mathcal{U}_o\}, \tag{2.5}$$

i.e., this is again a weighted proportionally fair allocation of the operators' slice of the base station resources.

2.5.2 Operator utility gains and protection

The *overall* network utility under MORA is clearly larger than that under the more constrained allocations possible under SS. This however does not guarantee that a given *operator's* utility under MORA is greater than that under SS. Below we show that for the *same* user association an operator utility under MORA exceeds that under SS, indicating that beyond the overall network utility, we have that each operator is indeed better off. This shows that MORA effectively protects operators when sharing their resources with other operators, which is very important to ensure that operators accept this criterion. Note that the result is completely general and holds for any possible scenario.[8]

Theorem 1. *For a given user association* $\mathbf{x}$*, MORA's resource allocation* $\mathbf{f}^M(\mathbf{x})$ *(see Eq. 2.3) achieves a higher utility than that of SS given by* $\mathbf{f}^S(\mathbf{x})$ *(see Eq. 2.5), i.e., for all* $o \in \mathcal{O}$

$$U_o(\mathbf{x}, \mathbf{f}^M(\mathbf{x})) \geq U_o(\mathbf{x}, \mathbf{f}^S(\mathbf{x})).$$

2.5.3 Capacity Savings

Next we consider the capacity savings resulting from operators sharing infrastructure. Specifically, we compare the spectrum capacities, i.e., total amount of resource, required to achieve the same *average utility* per operator under MORA and SS. The aim is to give some intuition on the typical savings one might expect and its dependence on the network load, number of operators and their shares. For

[8]The proofs of the theorems are provided in the Appendix.

tractability we will examine a scenario where traffic is spatially homogenous and operators' network shares are proportional to their load.

We consider a network model in which there is a *fixed* total number of users $|\mathcal{U}|$ of which each operator contributes a fixed number of users proportional its network share s_o, i.e., $n_o = s_o|\mathcal{U}|$ which are assumed to be integer valued. Each operator's users are randomly (uniformly) distributed amongst the $|\mathcal{B}|$ base stations, so the number of users of operator o associated with base station b, is given by a random variable $N_{o,b}$, such that $N_{o,b} \sim$ Binomial$(n_o, \frac{1}{|\mathcal{B}|})$. The total number of users at base station b is denoted by a random variable $N_b = \sum_{o\in\mathcal{O}} N_{o,b} \sim$ Binomial$(|\mathcal{U}|, \frac{1}{|\mathcal{B}|})$. We also assume for simplicity that users have the same capacity $c_{ub} = c$ to the base stations with which they associate.

Note that under the above traffic model every user u of every slice o have the *same* weight $w_u = \frac{s_o}{n_o} = \frac{1}{|\mathcal{U}|}$. Thus expected overall network utility under MORA is given by:

$$\begin{aligned}\bar{W} &= \mathbb{E}\left[\sum_{o\in\mathcal{O}}\sum_{b\in\mathcal{B}} N_{ob} w_u \log\left(\frac{c}{N_b}\right)\right] = \mathbb{E}\left[\sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}} \frac{N_{ob}}{|\mathcal{U}|}\log\left(\frac{c}{N_b}\right)\right] \\ &= \mathbb{E}\left[\sum_{b\in\mathcal{B}} \frac{N_b}{|\mathcal{U}|}\log\left(\frac{c}{N_b}\right)\right] = \frac{|\mathcal{B}|}{|\mathcal{U}|}\mathbb{E}\left[N_b \log\left(\frac{c}{N_b}\right)\right],\end{aligned}$$

where the last equality follows by using the uniformity of traffic across base stations. Moreover, under our model the network utility $\bar{W}$ is the average utility across all users, which by symmetry is equal to the expected utility of a given operator o under MORA, i.e., $\bar{U}_o^{MORA} = \bar{W}$.

Now applying Taylor's approximation to the function $x\log(c/x)$ at $\mathbb{E}[N_b]$

we obtain

$$N_b \log\left(\frac{c}{N_b}\right) \approx \mathbb{E}[N_b]\log\left(\frac{c}{\mathbb{E}[N_b]}\right) + \left[\log\left(\frac{c}{\mathbb{E}[N_b]}\right) - 1\right] \cdot (N_b - \mathbb{E}[N_b]) - \frac{1}{2\mathbb{E}[N_b]}(N_b - \mathbb{E}[N_b])^2,$$

which in turn gives

$$\mathbb{E}\left[N_b \log\left(\frac{c}{N_b}\right)\right] \approx \mathbb{E}[N_b]\log\left(\frac{c}{\mathbb{E}[N_b]}\right) - \frac{1}{2\mathbb{E}[N_b]}\text{Var}(N_b).$$

Since $N_b \sim \text{Binomial}(|\mathcal{U}|, \frac{1}{|\mathcal{B}|})$ we have that $\text{Var}(N_b) = \frac{|\mathcal{U}|}{|\mathcal{B}|}(1 - \frac{1}{|\mathcal{B}|}) \approx \frac{|\mathcal{U}|}{|\mathcal{B}|}$, and so

$$\bar{U}_o^{MORA} \approx \log\left(\frac{c}{\mathbb{E}[N_b]}\right) - \frac{|\mathcal{B}|}{2|\mathcal{U}|}. \tag{2.6}$$

Let Δ_o denote the extra capacity that operator o would require under SS to achieve the above utility. The expected utility experienced by operator o under SS is given by

$$\begin{aligned}\bar{U}_o^{SS} &= \mathbb{E}\left[\sum_{b\in\mathcal{B}} \frac{N_{o,b}}{n_o}\log\left(\frac{s_o c(1+\Delta_o)}{N_{o,b}}\right)\right] \\ &= \frac{|\mathcal{B}|}{n_o}\mathbb{E}\left[N_{o,b}\log\left(\frac{s_o c}{N_{o,b}}\right)\right] + \log(1+\Delta_o).\end{aligned}$$

Again, using a Taylor expansion this can be approximated as

$$\bar{U}_o^{SS} \approx \log\left(\frac{s_o c}{\mathbb{E}[N_{o,b}]}\right) - \frac{|\mathcal{B}|}{n_o}\frac{\text{Var}(N_{o,b})}{2\mathbb{E}[N_{o,b}]} + \log(1+\Delta_o).$$

Noting that $\text{Var}(N_{o,b}) \approx s_o\frac{|\mathcal{U}|}{|\mathcal{B}|} = \frac{n_o}{|\mathcal{B}|}$ we have that

$$\bar{U}_o^{SS} \approx \log\left(\frac{c}{\mathbb{E}[N_b]}\right) - \frac{|\mathcal{B}|}{2n_o} + \log(1+\Delta_o). \tag{2.7}$$

Finally equating the expected utilities, i.e., (2.6) and (2.7), we obtain the following estimate of the necessary extra capacity Δ_o required when static slicing rather than MORA is used:

$$\log(1+\Delta_o) \approx \frac{|\mathcal{B}|}{2n_o} \times (1-s_o). \tag{2.8}$$

where under our traffic load model $n_o = s_o|\mathcal{U}|$.

This result gives a clear intuition on the possible savings resulting from sharing the infrastructure with MORA dynamic slicing. In particular, the savings increase exponentially in the product of two terms. The first is inversely proportional to the average number of users operator o has per base station, i.e., $n_o/|\mathcal{B}|$; indeed, if the operator has a large number of users, its multiplexing gain is already high without sharing the infrastructure, and hence there is little gain from sharing. The second term is large when the operator has a small network share: if its share is high, the operator is using most of the network resources and there is little sharing.

In summary, capacity savings will be highest when infrastructure is shared by a large number of operators each with a small number of users per base station. With current trends towards small cells, the number of users per base station is expected to be small, suggesting that infrastructure sharing may be particularly beneficial.

2.6 Approximation algorithm for MORA

The analysis in previous section and simulations to be presented in the sequel suggest that MORA resource allocation across operators not only has desirable

characteristics but will make efficient use of resources while protecting operators from one another. Unfortunately, as we show below, the complexity and information overheads associated with doing so for are already high for a static system, and excessive when operators' mobile users and associated channels are subject to constant change. In this section, we further discuss the state-of-the-art algorithms to tackle MORA, and then propose an approximation algorithm based on a sequence of theoretical results and insights that support the design.

2.6.1 Complexity and state-of-the-art algorithms

The optimization problem underlying MORA is a *non-linear integer programming problem*, which can be shown to be NP-hard and hence there is no polynomial time algorithm unless $P = NP$.

Theorem 2. *The MORA problem is NP-hard.*

There have been a number of works in the literature devoted to solving problems similar to MORA. In particular, [74] proposes an approximation algorithm for the single operator case with guaranteed performance bounds. However, their approach is still computationally demanding; indeed, the results in Section 2.7.4, show that for a network with only 100 users, the algorithm takes 20 seconds on a dual-core 2.8GHz processor. Given that this would need to be executed every time c_{ub} values change or new users enter/leave the network, this seems computationally impractical. Moreover, the proposed approach is centralized, so there would be a substantial information overhead to gather the c_{ub} of each user to each potential base stations, given the amount of data and dynamic nature of mobile users.

In the multi-operator setting, [76] proposes an approach based on using a standard non-linear solver to address a problem similar to MORA. Unfortunately, the approach is also very complex and centralized. Indeed, our evaluation of this proposal in Section 2.7.4, shows that the time required to execute this algorithm increases sharply with the number of users, making it impractical at about 50 users. Moreover, [76] does not provide any analytical performance bounds.

In summary, to make dynamic multi-operator resource sharing possible, a new radically simplified approach is required. It should have low computational complexity and be based on distributed operation requiring only local information, to allow near real-time operation.

2.6.2 Algorithm design

In the following, we devise an algorithm for MORA that can be used in practical deployments. In contrast to previous approaches, our algorithm involves a low computational complexity and relies on data that can be gathered from neighboring base stations, allowing for a distributed implementation.[9]

Given the user dynamics, i.e., joining, moving and leaving the network, an offline algorithm that computes an optimal resource allocation for a fixed set of users is impractical. Instead, we will pursue an approach that tracks users dynamics, and occasionally adjusts resource allocations by modifying current or new users'

[9]Note that, while the algorithm implementation is distributed, the logic is centralized: i.e., we assume that the algorithm is run centrally by a single entity, without the intervention of the different operators.

associations. Since reassociations of current users correspond to handoffs, their number should be kept to a minimum. To design such an algorithm, we need to answer

- Do we really need to reassociate users?
- Where should users be (re)associated to?
- In which order should users be reassociated?
- How many reassociations do we need?

For each of these questions, in the following we provide some theoretical analysis that eventually leads to our proposed algorithm. In all cases, once a user association $\mathbf{x}$ is set, resources at each base station are allocated according MORA's resource allocation $\mathbf{f}^M(\mathbf{x})$.

2.6.2.1 Need for reassociations

Following the standard terminology of online algorithms, we say that an algorithm is *online* if, upon a user joining the network, it only decides how to associate the new user, without triggering any reassociations of existing users. We say the algorithm is *semi-online* if it can further trigger reassociations of a limited number of users. Thus, our first question is whether an online algorithm would suffice. The following theorem suggests that the performance of an online algorithm can be arbitrarily bad, motivating us to consider semi-online approaches.

Theorem 3. *Consider an online algorithm that triggers no reassociations of existing users. Let* $(\mathbf{x}', \mathbf{f}')$ *denote the solution resulting from this algorithm and* $(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ *a MORA optimal solution. Then,* $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - W(\mathbf{x}', \mathbf{f}')$ *cannot be bounded.*

2.6.2.2 Criterion for (re)associations

Next we address the question regarding how to associate, or reassociate, users to base stations. In particular, consider a *Distributed Greedy algorithm* wherein we iteratively examine (in arbitrary order) if there is a user which could change her association to increase her rate, and if this is the case, she chooses to re-associate with the base station providing the largest rate. The following result characterizes the performance of this algorithm *if* an equilibrium is reached.

Theorem 4. *Let* $(\mathbf{x}', \mathbf{f}')$ *be an equilibrium allocation for the Distributed Greedy algorithm, and* $(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ *a MORA optimal solution, then*[10]

$$W(\mathbf{x}', \mathbf{f}') \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - \log(e).$$

There exists an instance of the problem for which it holds that

$$W(\mathbf{x}', \mathbf{f}') = W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - \log(2).$$

Note that the above bound of $\log(e)$ is fairly close to the $\log(2)$ bound provided by [74]. This is quite remarkable, considering that the algorithm proposed

[10]To gain some intuition on this bound, we note that a $\log(e)$ gap is equivalent to reducing the throughput of each user by a factor of e.

in [74] is centralized and much more complex. Furthermore, the theorem shows that the bound is rather tight, as there exists a problem instance that provides a gap of $\log(2)$, which is quite close to the $\log(e)$ bound.

While the above theorem bounds network utility in equilibrium, we have not established the convergence of this algorithm to an equilibrium. The convergence of this type of algorithms has received substantial attention in the literature [37,40,78]. Indeed, since the throughput of user u is an increasing function of $c_{ub}/\sum_{v\in\mathcal{U}} w_v x_{vb}$, the Distributed Greedy algorithm can be viewed as a congestion game in which the load at a base station is given by the sum of weights of the users at the base station, $l_b = \sum_{v\in\mathcal{U}} w_v x_{vb}$, and a user seeks to minimize $a_{ub} l_b$, where $a_{ub} = 1/c_{ub}$. This game falls in the category of a singleton weighted congestion game with player-specific multiplicative constants and linear variable cost. Based on the lack of a counter-example and the existence of polynomial-time algorithms for special cases, [40] conjectures that this type of games have an equilibrium (see Conjecture 3.7 of [40]). Based on the simulations we have run for numerous instances of the game, we further conjecture that the Distributed Greedy algorithm (which implements a best response dynamics) converges to this equilibrium.

In particular, Distributed Greedy satisfies $W(\mathbf{x}', \mathbf{f}') \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - \log(e)$, while [74] proposes an algorithm that provides a throughput larger than $r_u(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})/(2+\epsilon)$ to all users, which translates into

$$W(\mathbf{x}', \mathbf{f}') \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - \log(2+\epsilon);$$

hence, the algorithm of [74] provides only a slightly tighter bound than Distributed

Greedy.

2.6.2.3 Order of reassociations

While our analysis of the Distributed Greedy algorithm suggests a user should (re)associate to maximize her rate, it does not indicate in which order user reassociations should be considered to speed up convergence. To address this, we consider the *Greedy Largest Gain algorithm*, which operates as the Distributed Greedy algorithm but at each iteration updates the association of the user achieving the highest gain, i.e., the one achieving the largest r_u^{new}/r_u^{old}, where r_u^{old} is the user's current throughput and r_u^{new} is the throughput she would receive under the improved association.

The following theorem shows that the Greedy Largest Gain algorithm exhibits a desirable convergence property. In particular, one can guarantee that at each iteration the network utility increases until it reaches $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - 2\log(e)$, and from then on it never decreases below $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - (2 + \max_u w_u)\log(e)$. Note that Distributed Greedy does not exhibit this kind of behavior: if we select users in an arbitrary order, the network utility may decrease at any iteration (as the increase in utility of the reassociated user may be smaller than the decrease experienced by the other users).

Theorem 5. *Let $(\mathbf{x}^i, \mathbf{f}^i)$ be the solution at the i^{th} iteration of the Greedy Largest Gain algorithm and $(\mathbf{x}^{MORA}, \mathbf{f}^{MORA})$ a MORA optimal solution. Then $W(\mathbf{x}^i, \mathbf{f}^i)$ increases at each iteration until $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - 2\log(e)$, and thereafter it never decreases below $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - (2 + \max_u w_u)\log(e)$.*

2.6.2.4 Proposed algorithm

Greedy Local Largest Gain. Based on the above considerations we now propose our algorithm for MORA, the *Greedy Local Largest Gain* algorithm. We shall first describe how it operates at a high level, and then provide a more detailed algorithmic description. When a user joins the network, she greedily joins the base station providing the largest throughput. However, as we have seen, we may need to consider triggering user reassociations. To limit their number and associated handoffs overheads we constrain these to at most m. For the first $m-1$ reassociations, users choose the base station that provides the largest throughput, but in the m^{th} the user chooses the base station so as to maximize the network utility $W(\mathbf{x}, \mathbf{f})$. In each of these steps, we select which user to reassociate (if any) based on Greedy Largest Gain criterion, but instead of considering all users in the network, involving possibly a high overhead, we restrict the selection *locally* to users associated with only two base stations (see below).

In a dynamic and time-varying setting, the algorithm needs to consider the following cases: (i) a user joins the network, (ii) leaves, or (iii) changes her location. The algorithm for a joining user is detailed in the pseudocode of next page. The rationale is as follows. In the optimal allocation, users are somehow balanced among base stations, users' weights playing a role in this balance. When a new user joins the network, the balance is broken and the base station with which the user associates may have too many users. Hence, in the first step we reassociate one of the users of this base station. In the next step, the base station that received the reassociated user may have too many users; however, depending on the weights of

the joining and reassociated users, the original base station may still have too many users as well. Hence, we consider the users from the two base stations as candidates for reassociation. We repeat this, considering users from two base stations, in the subsequent steps. Finally, in the last step, to avoid that the reassociation of a user harms the overall performance, we select the base station association that maximizes the overall network utility rather than the throughput of the reassociated user.

When a user leaves the network, the algorithm is quite similar. When she moves, her c_{ub} values to the neighboring base stations may change; if, as a result of these changes, at some point the user would receive a larger throughput in a new base station, we reassociate her to this base station. Then, the old base station executes the same algorithm as when a user leaves the network while the new base station executes the algorithm corresponding to a joining user.

2.6.2.5 Controlling the number of reassociations

The remaining question is how to set the limit on the number of reassociations m, which determines the trade-off between the performance of the algorithm and reassociation overhead. Such trade-offs have been analyzed for a similar setting in [104], which aims to distribute tasks among servers (where each task can only be associated to a restricted set of servers) in such a way that the maximum load across all servers is minimized. This problem is similar to ours, with tasks and servers corresponding to users and base stations respectively, in the particular case where all users have the same w_u and c_{ub}. Not unlike their setting, the performance

Algorithm 1: GLLG user joining.

Definitions:
$r_{v,b}$: throughput of user v if she associates to b;
r_v : current throughput of user v;
$\mathcal{U}_b$: set of users associated to b, $(u \in \mathcal{U}$ s.t. $x_{u,b} = 1)$;
$\mathcal{U}_{\{c \cup p\}}$: set of users associated to c or p;
$W_{u,q}$: network utility if user u associates to q;
Input: $\mathbf{x}$
User v joins the network:
$b' = \arg\max_{b \in \mathcal{B}} r_{v,b}$;
$x_{v,b'} = 1 \leftarrow$ Associate user v with base station b';
$[u^*, p^*] = \arg\max_{(u,p) \in \mathcal{U}_{b'} \times \mathcal{B}} \frac{r_{u,p}}{r_u}$;
if $r_{u^*,p^*}/r_u > 1$ **then**
  Associate user u^* with base station $p^*, x_{u^*p^*} = 1$;
else
  stop
$c = p^*$ (current base station); $p = b'$ (previous base station);
for $m - 1$ *times* **do**
  $[u^*, q^*] = \arg\max_{(u,q) \in \mathcal{U}_{\{c \cup p\}} \times \mathcal{B}} \frac{r_{u,q}}{r_u}$;
  if $r_{u^*,q^*}/r_u > 1$ **then**
    Associate user u^* with base station $q^*, x_{u^*q^*} = 1$;
    $c \leftarrow q^*$; $p \leftarrow$ previous base station of user u^*;
  else
    stop
$W \leftarrow$ current network utility;
$[u^*, q^*] = \arg\max_{(u,q) \in \mathcal{U}_{\{c \cup p\}} \times \mathcal{B}} \frac{W_{u,q}}{W}$;
if $W_{u^*,q^*}/W > 1$ **then**
  Associate user u^* with base station $q^*, x_{u^*q^*} = 1$;

in this case is optimized when base station loads are as balanced as possible (i.e., the highest load is minimized). According to the analysis of [104], the performance in terms of the highest load with our algorithm (which has a limit of m reassociations)

over the highest load with the optimal algorithm (with no constraint m) is given by $O(e^{1-\frac{m}{ln|\mathcal{B}|}})$. This shows that algorithm's performance improves rapidly (exponentially) in m, and suggests a small m suffices to achieve near-optimal network utility.

To further explore the impact of m on network utility, we present the following simulation results (see Section 2.7 for a description of the simulation setup). Here, $W(m)$ is the network utility achieved for a given m value, $W(\infty)$ is the utility with unconstrained overhead, $W(0)$ is the utility with no reassociations, and $G_W(m) \doteq 1 - \frac{W(m)-W(\infty)}{W(0)-W(\infty)}$ represents the normalized utility gain with m reassociations, showing how close we get to the unconstrained overhead utility. Fig. 2.1 depicts this gain as a function of m for different scenarios. As can be seen, utility gains increase very sharply. Furthermore, for $m = 3$ the gains are already very close to their maximum value; based on this, we set m equal to 3 (this is indeed the value used in the experiments of Section 2.7). With this setting, the proposed algorithm only introduces a small overhead, since our approach may trigger up to three handovers for every handover performed by a "traditional" solution [3].

2.7 Performance evaluation

Next, we evaluate the performance of our proposed approach. The mobile network scenario considered is based on the IMT Advanced evaluation guidelines for dense 'small cell' deployments [1]. It consists of base stations with an intersite distance of 200 meters in a hexagonal cell layout with 3 sector antennas (thus in this setting users will associate with sectors rather than the base stations we used in our

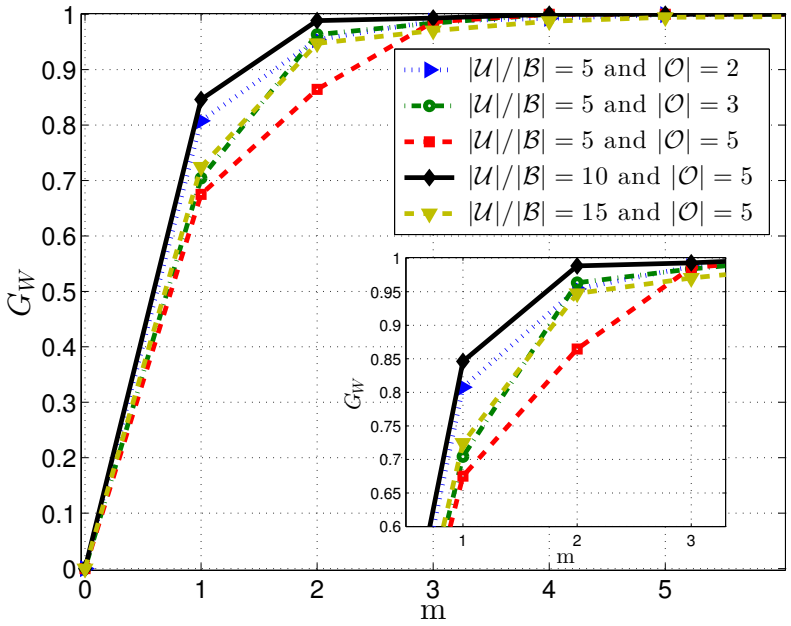

Figure 2.1: Normalized utility gain as a function of m.

algorithm description). The Signal Interference to Noise Ratio (SINR) is computed as in [108], $\text{SINR}_{ub} = P_b g_{ub}/(\sum_{k\in\mathcal{B},k\neq b} P_k g_{uk} + \sigma^2)$, where P_b is the transmit power and g_{ub} denotes the channel gain between user u and base station b, which includes path loss, shadowing, fast fading and antenna gain. Following [1], we set $P_b = 41$ dBm, $\sigma^2 = -104$dB, path loss equal to $36.7\log_{10}(\text{dist}) + 22.7 + 26\log_{10}(f_c)$ for carrier frequency $f_c = 2.5$GHz, and antenna gain of 17 dBi. The shadowing factor is given by a log-normal function with a standard deviation of 8dB (as in [108]) updated every second, and fast fading follows a Rayleigh distribution dependent of the user speed and the angle of incidence (as in [32]). Achievable rates are then computed with the Shannon formula, $\text{BW}\log_2(1+\overline{\text{SINR}}_{ub})$, for the average $\overline{\text{SINR}}_{ub}$ given by fading and shadowing [97] and a channel bandwidth of $\text{BW} = 10$MHz

[97]. Finally, the modulation-coding scheme is selected according to the $\overline{\text{SINR}}_{ub}$ thresholds reported in [7]. Unless otherwise stated (i) users move according to the Random Waypoint Model (RWP) with speeds uniformly distributed between 0.2 and 4 m/s and pause intervals between 0 and 10 seconds, (ii) network size $|\mathcal{B}|$ is 57 sectors, (iii) all operators have the same share, and (iv) the number of users of each operator is proportional to s_o, i.e., $|\mathcal{U}_o| = |\mathcal{U}| \cdot s_o$. Confidence intervals are below 1%.

2.7.1 Utility gains

We start by evaluating the gains in terms of the overall network utility. We consider a scenario with a user density of 10 users/sector and 3 operators, and plot $W(\mathbf{x}, \mathbf{f})$ as a function of the network size $|\mathcal{B}|$. In this setting, we compare the performance of our algorithm for dynamic sharing, *Greedy Local Largest Gain* ('GLLG'), against the following approaches:

i) *SINR-based Static Slicing* ('SINR SS'): the resources of each sector are statically divided among operators and users associate with the based station with highest SINR;

ii) *Distributed Greedy Static Slicing* ('DG SS'): resources are also sliced statically and user associations follow the Distributed Greedy algorithm discussed in Section 2.6.2.2;

iii) *Distributed Greedy* ('DG'): this is the algorithm for dynamic sharing presented in Section 2.6.2.2;

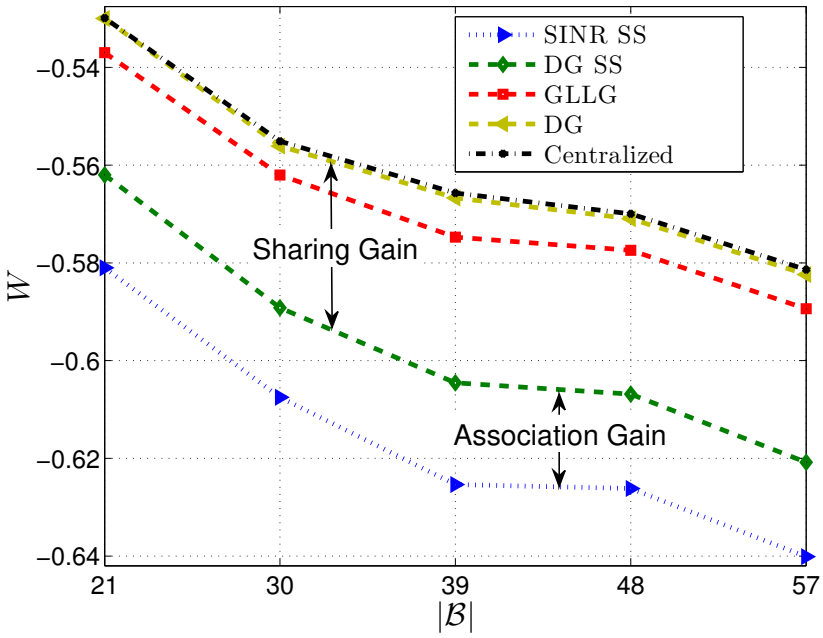

Figure 2.2: Utility gains for different approaches as a function of the network size.

iv) *Centralized* ('Centralized'): this is the centralized algorithm proposed in [74].

The results are exhibited in Fig. 2.2. We draw the following conclusions: (i) significant gains result from both improving user association (DG SS vs. SINR SS) and sharing resources dynamically (DG vs. DG SS); (ii) the Distributed Greedy approach of Section 2.6.2.2 performs almost at the same level of the baseline approach of [74] (DG vs. Centralized); and (iii) the proposed approach performs closely to these two approaches, although it pays a small price for reducing the handoff overheads (GLLG vs. DG).

In addition to the overall network gain, it is also interesting to look at the gains of the individual operators. Theorem 1 showed that the difference in oper-

ator's utility under MORA and SS is positive as long as we have the same user association in both approaches; however, we would expect this to hold in general, i.e., even when we have different user associations. To this end, we have evaluated the difference between the operator's utility under MORA and SS over a large number of different scenarios and settings. We have observed that in all cases, MORA always provides better performance than SS to all individual operators, which confirms that MORA effectively protects all operators, ensuring gains to all of them.

2.7.2 Capacity savings

We next evaluate the benefits of our approach to operators based on the capacity savings they would achieve. Specifically, consider a network operated under our algorithm for dynamic sharing, where the capacity (i.e., total amount of resource) of each base station is given by $\mathcal{C}_{GLLG}$, and let $C_{baseline}$ be the base stations' capacity required to achieve the same network utility under two baselines: (a) static slicing with SINR-based user association, and (b) static slicing with enhanced user association (i.e., using our algorithm for user association). These two baselines allow us to study the potential gains earned due to a smarter user association and the gains achieved by dynamic resource sharing. Fig. 2.3 illustrates the corresponding capacity savings, computed as $\Delta = (\mathcal{C}_{baseline} - \mathcal{C}_{GLLG})/\mathcal{C}_{GLLG}$, for different numbers of operators, $|\mathcal{O}| \in \{2, \ldots, 6\}$, and three different user densities, $|\mathcal{U}|/|\mathcal{B}| = 5$ (low density), $|\mathcal{U}|/|\mathcal{B}| = 10$ (medium) and $|\mathcal{U}|/|\mathcal{B}| = 15$ (high). The results show that substantial gains can be realized, and that gains increase with the number of operators and decrease with per-sector user load. The latter is indeed

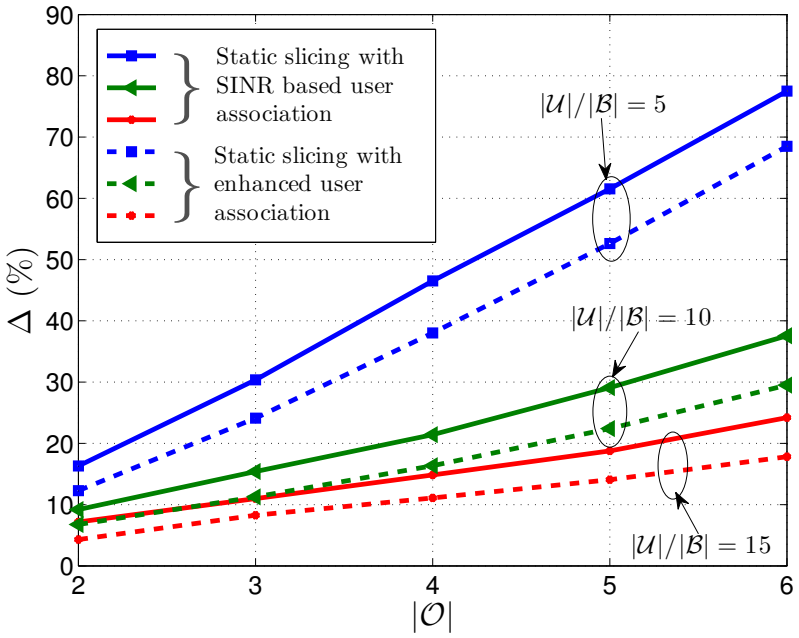

Figure 2.3: Capacity savings for different scenarios as a function of the number of operators.

rather intuitive, since under light user loads static slicing performs poorly while MORA obtains substantial benefits from statistical multiplexing.

In order to gain additional insight into the impact of the various factors, Fig. 2.4 displays the influence of the share of the operator (s_o) and the average load per base station sector $|\mathcal{U}|/|\mathcal{B}|$ in the percent of extra capacity required to achieve the same utility (Δ) with the static slicing with enhanced user association baseline. Results are also compared with the analytical result of Section 2.5.3, confirming that the theoretical analysis result holds in real conditions.

Note that in the above experiments all operators always have the same share s_o. To illustrate the behavior of MORA under heterogeneous shares, we evaluated

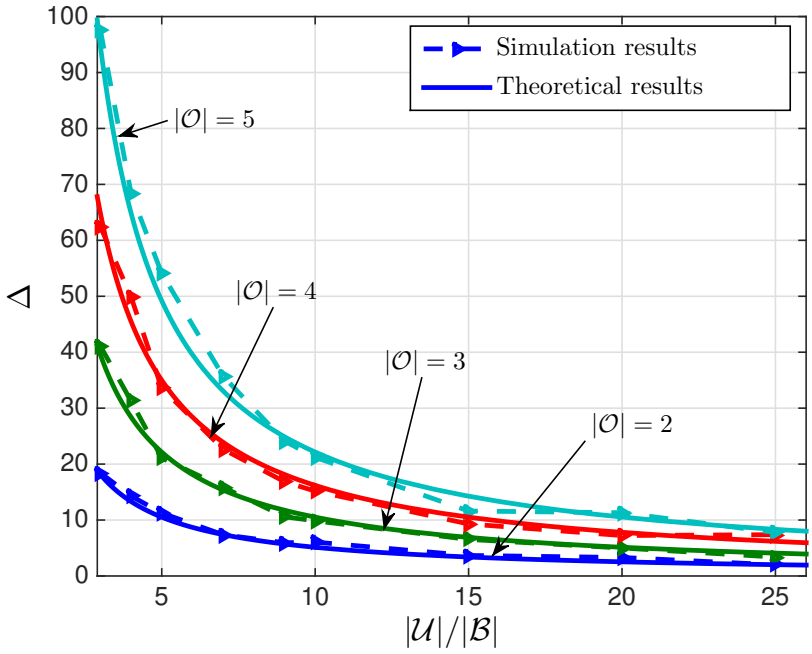

Figure 2.4: Validation of the theoretical results on capacity savings.

the performance of a scenario with $|\mathcal{U}|/|\mathcal{B}| = 5$ and 2 operators under the following share settings: (i) $s_1 = s_2 = 1/2$ and (ii) $s_1 = 2/3$ and $s_2 = 1/3$. The gains obtained for operators 1 and 2 in the former case are $G_1 = G_2 = 11.1\%$, while in the latter case they are $G_1 = 5.3\%$ and $G_2 = 21.6\%$, respectively. Thus, this result shows that overall performance remains similar under heterogeneous shares, but gains are unevenly distributed.

2.7.3 User performance

To illustrate the gains from a user perspective, we compare the per-user throughput achieved by our approach against the two baselines: static slicing with SINR-based user association ('Baseline 1'), and static slicing with enhanced user

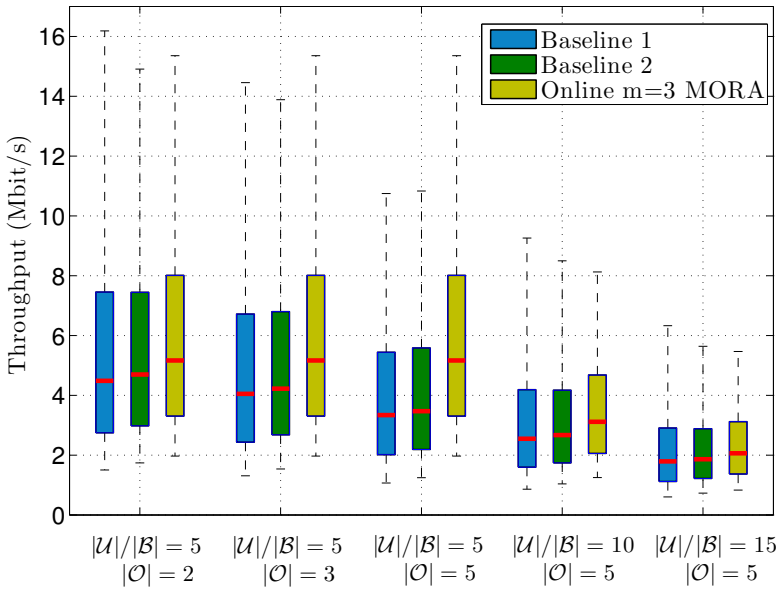

Figure 2.5: Improvement on the user throughput.

association ('Baseline 2'). The resulting box-and-whisker plots are shown in Fig. 2.5 for different user densities and numbers of operators. We observe that our approach provides substantial gains both in terms of the median values as well as the various percentiles. Furthermore, as expected, gains increase with the number of operators but decrease with per-sector user load.

To complement the previous results, we compare the file download times achieved by our approach against a baseline scenario (static slicing with enhanced user association), when base stations have the same capacity in both cases and users are constantly downloading files. Let us define the file download time gain as $G_D = (D_{SS} - D_{GLLG})/D_{SS}$, where D_{SS} is the average file download time with the static slicing approach and D_{GLLG} with ours. The gains achieved are shown in Fig. 2.6 as a function of the file download size, for different user densities and numbers of

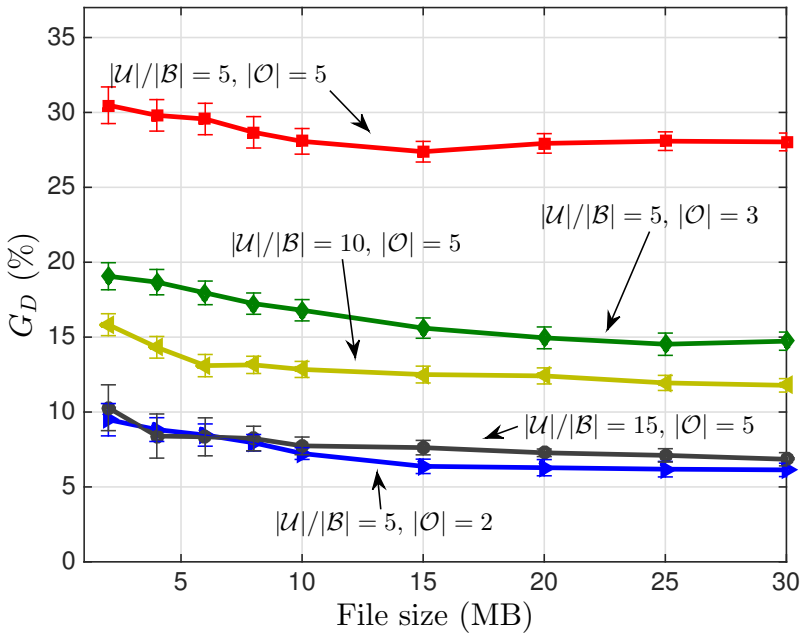

Figure 2.6: Improvement on the file download time for different file sizes.

operators. We observe the gains are substantial, and fairly independent of the file size.

2.7.4 Computational complexity

As mentioned in Section 2.6.1, one of the key advantages of the proposed approach over the state-of-the-art is its reduced computational complexity. To quantify this, we have measured the time required to execute the following algorithms in a dual-core 2.8GHz processor: (i) our algorithm for dynamic sharing ('GLLG'); (ii) the Distributed Greedy approach of Section 2.6.2.2, which has unconstrained overhead ('DG'); (iii) the centralized algorithm of [74] ('Centralized'); and (iv) the non-linear solver used by [76] ('Non-linear Solver'). Fig. 2.7 shows the resulting

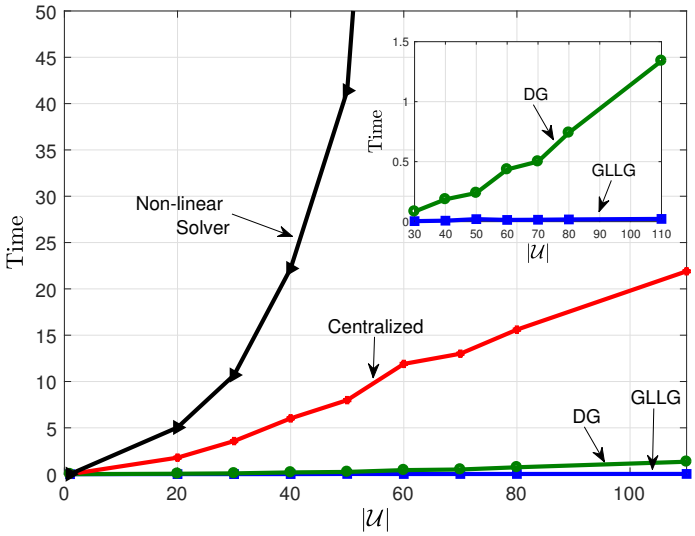

Figure 2.7: Computational complexity of our approaches and state-of-the-art algorithms.

execution times (in seconds) as a function of the number of users for a fixed network size $|\mathcal{B}| = 57$ and $|\mathcal{O}| = 4$ operators. The results confirm that the algorithms of [76] and [74] are impractical, especially if we take into account that they have to be triggered every time the channel quality of a user changes. By contrast, the execution time of our Distributed Greedy algorithm remains very low, and it remains even lower for our GLLG approach (due to the constraint that GLLG imposes on the number of handovers).

2.7.5 Impact of non-uniform load distributions

All the results shown so far have been based on the RWP mobility model, which is known to distribute load uniformly across space. To understand the impact of non-uniform load distributions, we have evaluated the capacity savings

over a baseline (static slicing with enhanced user association) under the SLAW model [71], which is a non-uniform human walk mobility model. To show different levels of non-uniformity, we have parameterized the SLAW model with five configurations of increasing non-uniformity, from $C1$ to $C5$, whose parameters $\{$waypoints, clustering range, alpha distance, inverse self-similarity$\}$ are set as follows: $C1 = \{100, 20, 5, 0.95\}$, $C2 = \{85, 40, 4.5, 0.85\}$, $C3 = \{75, 60, 4, 0.75\}$, $C4 = \{65, 80, 3.5, 0.65\}$ and $C5 = \{50, 100, 3, 0.55\}$. The results, given in Fig. 2.8, show that (as expected) capacity savings decrease if loads are non-uniform, since when users concentrate around some areas the expected number of users per sector in those areas increases and thus multiplexing gains are reduced. However, the decrease is very gradual, which shows that non-uniformity has a limited impact.

The above experiment assumes that all operators follow the same mobility pattern. Alternatively, we may assume different patterns for different operators, which may be the case for instance if we consider services of different nature. To evaluate the performance under such case, we have run additional simulations in which each operator follows a different instance of the SLAW model, with different waypoints. The results, given in Figs. 2.9, show that in this case gains increase (rather than decrease) with non-uniformity, as each operator may have its users concentrated in different areas, thereby maximizing the benefit from resource sharing.

2.8 Conclusions

In this chapter we have addressed the problem of multi-tenant resource slicing. While there has been substantial work towards addressing this problem, most

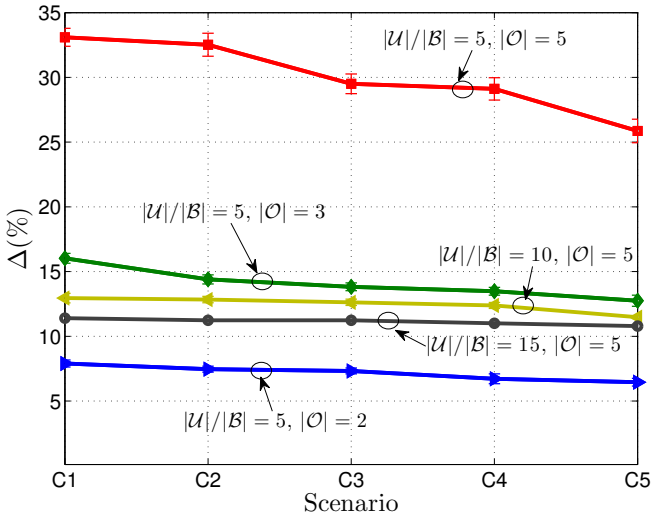

Figure 2.8: Capacity savings for different levels of non-uniformity under the SLAW mobility model.

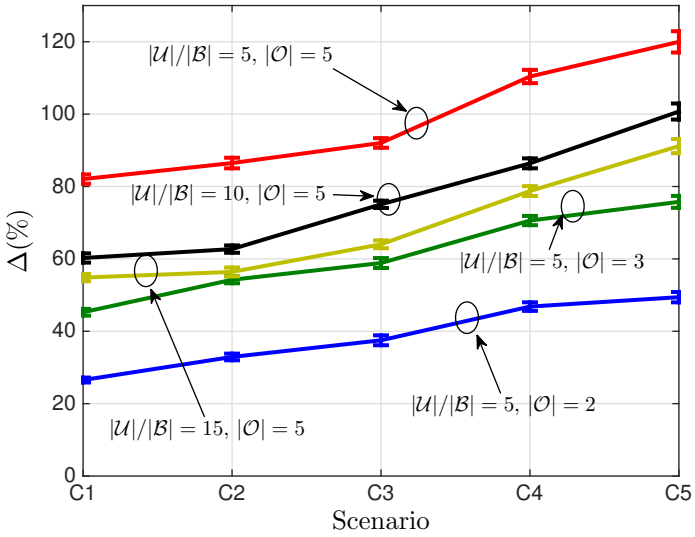

Figure 2.9: Capacity savings for different levels of non-uniformity when operators follow different patterns.

has focused on architectural issues, leaving algorithmic aspects open to consideration. The design of algorithms for dynamic resource sharing across slices is challenging as it involves user association decisions (a difficult problem per se) as well as multi-operator sharing policies. Our main contribution has been to show that, despite its complexity, it is possible to design practical solutions that scale to large networks and can track network load dynamics. Indeed, our analytical results provide strong evidence that the resulting allocations are near-optimal, and our simulations confirm robust benefits to operators (in terms of capacity savings) as well as to users (in terms of improved performance).

2.9 Proofs of chapter results

2.9.1 Proof of Theorem 1

For a given user association $\mathbf{x}$ the utility of operator o under SS is maximized when the resource blocks of each operator at each base station are equally distributed among the operator's users. This yields

$$\begin{aligned} &U_o(\mathbf{x}, \mathbf{f}^{\mathbf{S}}(\mathbf{x})) = \\ &= \sum_{b\in\mathcal{B}} \sum_{u\in\mathcal{U}_o} \frac{1}{|\mathcal{U}_o|} x_{ub} \log\left(\frac{1}{\sum_{b\in\mathcal{B}} \sum_{v\in\mathcal{U}_o} x_{vb}} \frac{s_o}{\sum_{o'\in\mathcal{O}} s_{o'}} c_{ub} \right) \\ &= \frac{1}{s_o} \sum_{b\in\mathcal{B}} \sum_{u\in\mathcal{U}_o} w_u x_{ub} \log\left(\frac{1}{\sum_{b\in\mathcal{B}} \sum_{v\in\mathcal{U}_o} x_{vb}} \frac{s_o}{\sum_{o'\in\mathcal{O}} s_{o'}} c_{ub} \right) \end{aligned}$$

where the weights are $w_u = \frac{s_o}{|\mathcal{U}_o|},\ u \in \mathcal{U}_o$.

If we multiply the numerator and denominator inside the $\log()$ by w_u, and take into account that $w_u = w_v$ for $u, v \in \mathcal{U}_o$ and $\sum_{o'\in\mathcal{O}} s_{o'} = 1$, the above can be

rewritten as

$$U_o(\mathbf{x}, \mathbf{f}^S(\mathbf{x})) = \frac{1}{s_o} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log(w_u c_{ub}) - \frac{1}{s_o} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log \left(\frac{\sum_{b \in \mathcal{B}} \sum_{v \in \mathcal{U}_o} w_v x_{vb}}{s_o} \right).$$

The utility of operator o with MORA allocation is given by

$$U_o(\mathbf{x}, \mathbf{f}^M(\mathbf{x})) = \frac{1}{s_o} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log \left(\frac{w_u c_{ub}}{\sum_{v \in \mathcal{U}} w_v x_{vb}} \right),$$

which can be rewritten as

$$U_o(\mathbf{x}, \mathbf{f}^M(\mathbf{x})) = \frac{1}{s_o} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log(w_u c_{ub}) - \frac{1}{s_o} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log \left(\sum_{v \in \mathcal{U}} w_v x_{vb} \right).$$

From the above, if we can show that

$$\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log \left(\frac{\sum_{b \in \mathcal{B}} \sum_{v \in \mathcal{U}_o} w_v x_{vb}}{s_o} \right) \geq \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_o} w_u x_{ub} \log \left(\sum_{v \in \mathcal{U}} w_v x_{vb} \right), \tag{2.9}$$

the theorem is proved.

To show the above, we consider the maximization of function $\sum_{b \in \mathcal{B}} y_b \log(x_b)$ over x_b subject to $\sum_{b \in \mathcal{B}} x_b = 1$. By applying Lagrange multipliers, it can be easily seen that this function is maximized for $x_b = y_b / \sum_{b' \in \mathcal{B}} y_{b'}$. Since both the left and right-hand sides of (2.9) conform to this constrained optimization problem, and the left-hand side of (2.9) corresponds to its optimal solution, the inequality of (2.9) follows. □

2.9.2 Proof of Theorem 2

The reduction is via the 3-dimensional matching problem which is known to be NP-complete. Recall that the 3-dimensional matching problem is stated as follows. Let us consider disjoint sets $C = \{c_1, \ldots, c_n\}$, $D = \{d_1, \ldots, d_n\}$ and $E = \{e_1, \ldots, e_n\}$, and a family $T = \{T_1, \ldots, T_m\}$ of triples with $|T_i \cap C| = |T_i \cap D| = |T_i \cap E| = 1$ for $i = 1, \ldots, m$, with $m \geq n$. The question is whether T contains a matching, i.e., a subfamily T' for which $|T'| = n$ and $\cup_{T_i \in T'} T_i = C \cup D \cup E$.

Our reduction is as follows. We call the triples that contain c_j *triples of type j*. Let t_j be the number of triples of type j for $j = 1, \ldots, n$. Base station i corresponds to the triples T_i for $i = 1, \ldots, m$. We create two types of users, element users and dummy users. We have $2n$ element users, $u \in \{1, \ldots, 2n\}$, corresponding to the $2n$ elements of $D \cup E$. There are $t_j - 1$ dummy users of type j for $j = 1, \ldots, n$. Note that the total number of dummy users is $m - n$, $u \in \{2n + 1, \ldots, m + n\}$. Element users can connect to the base stations that correspond to a triple that contains this element, with a transmission rate of R. Dummy users of type j can connect (also with a transmission of R) to the base stations that correspond to triples of type j. Element users have a weight $w_u = 1/(2m)$ and dummy users have a weight $w_u = 1/m$. We claim that a matching exists if and only if the network utility with the MORA criterion is $W = (n/m)\log(R/2) + ((m - n)/m)\log(R)$.

The value of the objective function is bounded above by the following opti-

mization problem:

$$\max_{\mathbf{f}} \sum_{u=1}^{2n} \frac{1}{2m} \log(f_u R) + \sum_{u=2n+1}^{m+n} \frac{1}{m} \log(f_u R),$$

subject to $\sum_{u=1}^{2n} f_u + \sum_{u=2n+1}^{m+n} f_u = m$, where f_u is the fraction of resources assigned to user u (the first term of the summation corresponds to the element users and the second term to the dummy users).

By applying the Lagrange multiplier method, it can be easily seen that the above optimization problem is solved when $f_u = 1/2$ for the element users and $f_u = 1$ for the dummy users. This gives an upper bound on W equal to $(n/m)\log(R/2) + ((m-n)/m)\log(R)$. This corresponds to a global maximum, and thus any other set of f_u values yields a smaller W.

Assume that there is a matching. For each $T_i = (c_j, d_k, e_l)$ in the matching, we associate element users d_k and e_l with base station i. For each j, this leaves $t_j - 1$ idle base stations corresponding to tripes of type j that are not in the matching. We associate the $t_j - 1$ dummy users of type j to these $t_j - 1$ base stations. This assignment has an objective function of $W = (n/m)\log(R/2) + ((m-n)/m)\log(R)$, which is equal to the upper bound given above. In case there is no matching, it is not possible to have the $2n$ element users sharing n base stations with $f_u = 1/2$ each, and therefore we cannot achieve the distribution of f_u values that maximizes W. According to the above result, this imples that we obtain a smaller W value. Therefore, a matching exists if and only if MORA gives $W = (n/m)\log(R/2) + ((m-n)/m)\log(R)$, which proves the theorem. □

2.9.3 Proof of Theorem 3

We prove the theorem by means of the following example. Let us consider a scenario with $|\mathcal{B}|$ base stations in which $|\mathcal{B}|^2$ users join the network. All users have the same weight and can associate with any of the $|\mathcal{B}|$ base stations with $c_{ub} = 1$. Independently of the criterion followed to associate new users, after all users have joined there must be a base station with at least $|\mathcal{B}|$ users. Now, suppose all users but these $|\mathcal{B}|$ leave the network. For this scenario, the network utility provided by the online algorithm is $W(\mathbf{x}', \mathbf{f}') = \sum_{i=1}^{|\mathcal{U}|} \frac{1}{|\mathcal{B}|} \log(\frac{1}{|\mathcal{B}|}) = -\log(|\mathcal{B}|)$. The optimal solution is that each user associates with a different base station, which yields $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) = \log(1)$. Thus, we have $W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - W(\mathbf{x}', \mathbf{f}') = \log(1) + \log(|\mathcal{B}|)$, which grows to ∞ as $|\mathcal{B}| \to \infty$.

2.9.4 Proof of Theorem 4

Since in an equilibrium of the Distributed Greedy algorithm, each user is associated with the base station that maximizes r_u, the following holds for all u:

$$\sum_{b \in \mathcal{B}} x'_{ub} w_u \log\left(\frac{w_u c_{ub}}{\sum_{v \in \mathcal{U}} x'_{vb} w_v}\right) \geq \sum_{b \in \mathcal{B}} x^*_{ub} w_u \log\left(\frac{w_u c_{ub}}{\sum_{v \in \mathcal{U}} x'_{vb} w_v + w_u}\right), \tag{2.10}$$

where the base station for which $x'_{ub} = 1$ is the one with which user u is associated under Distributed Greedy, and the base station for which $x^*_{ub} = 1$ is the one with which it is associated under the optimal allocation (i.e., $\mathbf{x}^* = \mathbf{x}^{MORA}$).

At the base station for which $x^*_{ub} = 1$ we have $\sum_{v \in \mathcal{U}} x^*_{vb} w_v \geq w_u$, so the

following also holds:

$$\sum_{b\in\mathcal{B}} x'_{ub} w_u \log\left(\frac{w_u c_{ub}}{\sum_{v\in\mathcal{U}} x'_{vb} w_v}\right) \geq \sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{w_u c_{ub}}{\sum_{v\in\mathcal{U}} x'_{vb} w_v + \sum_{v\in\mathcal{U}} x^*_{vb} w_v}\right).$$

Let us define the load at a base station as the sum of weights of the users at the base station, $l_b = \sum_{v\in\mathcal{U}} w_v x_{vb}$. Then, the above can be rewritten as

$$\sum_{b\in\mathcal{B}} x'_{ub} w_u \log\left(\frac{w_u c_{ub}}{l'_b}\right) \geq \sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{w_u c_{ub}}{l'_b + l^*_b}\right),$$

where l'_b and l^*_b are the load at base station b with the Distributed Greedy algorithm and the optimal allocation, respectively.

From the above it follows that

$$w_u \log(r_u(\mathbf{x}^*, \mathbf{f}^*)) - w_u \log(r_u(\mathbf{x}', \mathbf{f}')) \leq \sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{w_u c_{ub}}{l^*_b}\right) - \sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{w_u c_{ub}}{l'_b + l^*_b}\right),$$

where $\mathbf{f}^* = f^M(\mathbf{f}^*)$. The above can be expressed as

$$w_u \log(r_u(\mathbf{x}^*, \mathbf{f}^*)) - w_u \log(r_u(\mathbf{x}', \mathbf{f}')) \leq -\sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{l^*_b}{l'_b + l^*_b}\right).$$

Summing the above over all users yields

$$W(\mathbf{x}^*, \mathbf{f}^*) - W(\mathbf{x}', \mathbf{f}') \leq -\sum_{u\in\mathcal{U}} \sum_{b\in\mathcal{B}} x^*_{ub} w_u \log\left(\frac{l^*_b}{l'_b + l^*_b}\right).$$

From the above,

$$W(\mathbf{x}^*, \mathbf{f}^*) - W(\mathbf{x}', \mathbf{f}') \leq -\sum_{b\in\mathcal{B}} \log\left(\frac{l^*_b}{l'_b + l^*_b}\right)^{\sum_{u\in\mathcal{U}} x^*_{ub} w_u} =$$

$$-\sum_{b\in\mathcal{B}} \sum_{u\in\mathcal{U}} x'_{ub} w_u \log\left(\frac{l^*_b / l'_b}{1 + l^*_b / l'_b}\right)^{\frac{\sum_{v\in\mathcal{U}} x^*_{vb} w_v}{\sum_{v\in\mathcal{U}} x'_{vb} w_v}} =$$

$$-\sum_{b\in\mathcal{B}} \sum_{u\in\mathcal{U}} x'_{ub} w_u \log\left(\frac{l^*_b / l'_b}{1 + l^*_b / l'_b}\right)^{l^*_b / l'_b}.$$

Given that $(x/(1+x))^x > 1/e$ for $x \geq 0$, we obtain the following bound:

$$W(\mathbf{x}^*, \mathbf{f}^*) - W(\mathbf{x}', \mathbf{f}') \leq \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}} x'_{ub} w_u \log(e) = \log(e).$$

Since $\mathbf{x}^* = \mathbf{x}^{MORA}$ and $\mathbf{f}^* = \mathbf{f}^{MORA}$, this proves the first part of the theorem.

To find an instance for which the network utility difference between MORA and Distributed Greedy Algorithm is $\log(2)$, consider the following scenario. Consider a network with 2 base stations $\mathcal{B} = \{1, 2\}$ and 2 operators $\mathcal{O} = \{1, 2\}$ with equal shares, $s_1 = s_2 = 0.5$. Each operator has one user: User 1 belongs to Operator 1 and User 2 to Operator 2. Let the achievable rates be $c_{1,1} = c_{2,2} = R$ and $c_{1,2} = c_{2,1} = R/2$, i.e, user 1 sees a higher rate with base station 1 and user 2 with base station 2. Clearly, the optimal MORA solution is to associate user 1 with base station 1 and User 2 with base station 2, i.e., $x^M_{1,1} = 1$ and $x^M_{2,2} = 1$,. This leads to a network utility $W(\mathbf{x^M}, \mathbf{f^M}) = 0.5 \log(c_{1,1}) + 0.5 \log(c_{2,2}) = \log(R)$.

Distributed Greedy Algorithm only reassociates a user if this increases her rate. Let user 1 be associated with base station 2 and user 2 with base station 2. Since none of the two users can increase her rate by reassociating, they will not reassociate with the Distributed Greedy Algorithm, and hence this algorithm will result in a user association decision $\mathbf{x}'$ such that $x'_{1,2} = 1$ and $x'_{2,1} = 1$. This yields a network utility $W(\mathbf{x}', \mathbf{f}') = 0.5 \log(c_{1,2}) + 0.5 \log(c_{2,1}) = \log(R/2) = \log(R) - \log(2) = W(\mathbf{x^M}, \mathbf{f^M}) - \log(2)$, which proves the second part of the theorem. □

2.9.5 Proof of Theorem 5

The proof of the theorem is based on the following steps:

Step 1: we first show that while there is some user for which $r_u^{new} \geq e \cdot r_u^{old}$, $W(\mathbf{x}^i, \mathbf{f}^i)$ increases at each iteration until we converge to a region that satisfies $r_u^{new} \leq e \cdot r_u^{old}$ for all u.

Step 2: we then show that if $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$, it follows that $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{x}^{MORA}) - 2\log(e)$.

Step 3: we further prove that if a subsequent iteration i yields $r_u^{new} \geq e \cdot r_u^{old}$ for some user u, then it must be that $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{x}^{MORA}) - (2 + \max_u w_u)\log(e)$.

Step 4: finally, we prove that after an iteration such as the above, in the subsequent iterations $W(\mathbf{x}^i, \mathbf{f}^i)$ increases, until we converge once again to a region where $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$.

We next prove each of the above steps.

Step 1: *While there is some user for which $r_u^{new} \geq e \cdot r_u^{old}$, $W(\mathbf{x}^i, \mathbf{f}^i)$ increases at each iteration until we converge to a region that satisfies $r_u^{new} \leq e \cdot r_u^{old}$ for all u.*

To prove the above, we consider a variation of the Greedy Largest Gain in which a user only moves to a new location if $r_u^{new} \geq e \cdot r_u^{old}$, and show that this algorithm is guaranteed to converge. To show this, we prove that the network utility function $W(\mathbf{x}, \mathbf{f})$ is a generalized ordinal potential for the algorithm variation. Consider the i^{th} iteration in the algorithm corresponding to a reassociation of user

u, and let $(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})$ denote the configuration before this iteration and $(\mathbf{x}^i, \mathbf{f}^i)$ the configuration after the iteration. By construction of the algorithm, the following is satisfied:

$$r_u(\mathbf{x}^i, \mathbf{f}^i) \geq e \cdot r_u(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}).$$

Let b be the new base station user u associates with, and a her previous base station. Then,

$$\begin{aligned} W(\mathbf{x}^i, \mathbf{f}^i) - W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}) &= \sum_{v \in \mathcal{U}} x^i_{va} w_v \log \left(\frac{\sum_{y \in \mathcal{U}} x^i_{ya} w_y + w_u}{\sum_{y \in \mathcal{U}} x^i_{ya} w_y} \right) + \\ &\sum_{v \in \mathcal{U} \setminus \{u\}} x^i_{vb} w_v \log \left(\frac{\sum_{y \in \mathcal{U} \setminus \{u\}} x^i_{yb} w_y}{\sum_{y \in \mathcal{U} \setminus \{u\}} x^i_{yb} w_y + w_u} \right) + \\ &+ w_u \log(r_u(\mathbf{x}^i, \mathbf{f}^i)) - w_u \log(r_u(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})) \\ &= l^i_a \log \left(\frac{l^i_a + w_u}{l^i_a} \right) + l^{i-1}_b \log \left(\frac{l^{i-1}_b}{l^{i-1}_b + w_u} \right) \\ &+ w_u \log(r_u(\mathbf{x}^i, \mathbf{f}^i)) - w_u \log(r_u(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})). \end{aligned}$$

Since $l^i_a \log \left(\frac{l^i_a + w_u}{l^i_a} \right) \geq 0$, we have

$$\begin{aligned} W(\mathbf{x}^i, \mathbf{f}^i) - W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}) &\geq w_u \log \left(\frac{l^{i-1}_b / w_u}{1 + l^{i-1}_b / w_u} \right)^{\frac{l^{i-1}_b}{w_u}} + w_u \log \left(\frac{r_u(\mathbf{x}^i, \mathbf{f}^i)}{r_u(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})} \right) \\ &> w_u \log(1/e) + w_u \log(e) = 0, \end{aligned} \tag{2.11}$$

so that $W(\mathbf{x}, \mathbf{f})$ is a generalized ordinal potential. This implies that the potential game corresponding to the algorithm variation has the finite improvement property; therefore, the algorithm variation converges in a finite number of iterations to a solution that satisfies $r^{new}_u \leq e \cdot r^{old}_u \ \forall u$. Also, from (2.11) it follows that $W(\mathbf{x}^i, \mathbf{f}^i) > W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})$, i.e., the network utility increases at each iteration.

As the Greedy Largest Gain algorithm always selects the user with the largest r_u^{new}/r_u^{old}, it will select a user for which $r_u^{new} \geq e \cdot r_u^{old}$, as long as there is one that satisfies this condition, and hence will follow the same steps as the algorithm variation that we have considered above. This implies that there will be some iteration i in which the Greedy Largest Gain algorithm will reach a solution $(\mathbf{x}^i, \mathbf{f}^i)$ that satisfies $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$ and, until reaching this solution, $W(\mathbf{x}^i, \mathbf{f}^i)$ will increase at each iteration.

Step 2: *If* $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$*, it follows that* $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{x}^{MORA}) - 2\log(e)$.

Let $(\mathbf{x}^i, \mathbf{f}^i)$ be the solution at the i^{th} iteration which satisfies $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$. Equation (2.10) for this solution can be rewritten as

$$\sum_{b \in \mathcal{B}} x_{ub}^i w_u \log\left(\frac{w_u c_{ub}}{\sum_{v \in \mathcal{U}} x_{vb}^i w_v}\right) \geq \sum_{b \in \mathcal{B}} x_{ub}^{MORA} w_u \log\left(\frac{w_u c_{ub}}{\sum_{v \in \mathcal{U}} x_{vb}^i w_v + w_u}\right) - w_u \log(e).$$

Starting from the above equation and applying the same reasoning as in the proof of Theorem 4 yields $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - 2\log(e)$.

Step 3: *If a subsequent iteration* i *yields* $r_u^{new} \geq e \cdot r_u^{old}$ *for some user* u*, then it must be that* $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{x}^{MORA}) - (2 + \max_u w_u)\log(e)$.

Let us that for some iteration i of the algorithm such that it holds $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$ for the solution before this iteration, and $r_u^{new} \leq e \cdot r_u^{old}$, for some u, for the solution after the iteration. Let $(\mathbf{x}^{i-1}, \mathbf{f}^{i-1})$ be the solution before iteration i

and $(\mathbf{x}^i, \mathbf{f}^i)$ the solution after the iteration. As we have seen above, for the former it holds $W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - 2\log(e)$. Let us consider that at iteration i user u moves to base station b. Then,

$$W(\mathbf{x}^i, \mathbf{f}^i) - W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}) \geq \sum_{v \in \mathcal{U}} x_{vb}^{i-1} w_v \log\left(\frac{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t}{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t + w_u}\right) =$$

$$= w_u \log\left(\frac{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t / w_u}{1 + \sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t / w_u}\right)^{\frac{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t}{w_u}}$$

$$\geq -w_u log(e) \geq -\max_u w_u log(e).$$

Thus,

$$W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - (2 + \max_u w_u)\log(e).$$

Step 4: *After an iteration such as the above, in the subsequent iterations* $W(\mathbf{x}^i, \mathbf{f}^i)$ *increases, until we converge once again to a region where* $r_u^{new} \leq e \cdot r_u^{old}\ \forall u$.

Let us consider that before iteration i there is some u for which $r_u^{new} \geq e \cdot r_u^{old}$. Then,

$$W(\mathbf{x}^i, \mathbf{f}^i) - W(\mathbf{x}^{i-1}, \mathbf{f}^{i-1}) \geq w_u \log(r^{new}) - w_u \log(r^{old}) +$$
$$\sum_{v\in\mathcal{U}} x_{vb}^{i-1} w_v \log\left(\frac{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t}{\sum_{t\in\mathcal{U}} x_{tb}^{i-1} w_t + w_u}\right) > w_u \log(e) - w_u \log(e) \geq 0.$$

Therefore, if at some iteration we get $r_u^{new} \geq e \cdot r_u^{old}$ for some u, then for that iteration it will hold $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - (2 + \max_u w_u)\log(e)$, and from this point on $W(\mathbf{x}^i, \mathbf{f}^i)$ is going to increase until we reach $W(\mathbf{x}^i, \mathbf{f}^i) \geq W(\mathbf{x}^{MORA}, \mathbf{f}^{MORA}) - 2\log(e)$ again. □

Chapter 3

Optimizing Network Slicing via Virtual Resource Pool Partitioning

Managing a network consisting in a collection of distributed resources is simple when the load is deterministic and/or predictable, since through network capacity planning one can in principle provision only the neccesary resources achieving optimal utilization. However, when serving a spatially distributed set of mobile customers/demands, this becomes a challenging task. Fluctuations of the network's load may result in inadequate or low capacity utilization if the network does not adapt to these changes. Ideally, network managment should be performed so that resources behave as if they were a single resource with pooled capacities; this is referred to as *resource pooling* [105].

Achieving perfect resource pooling requires the ability to seamlessly shift/ transfer either load or capacity across resources. While "perfect" resource pooling is often unachievable in practice, particularly in the context of wireless networks, techniques such as load balancing and dynamic resource allocation, among others [56], have been traditionally designed to enable a distributed network to appear as *virtual resource pool* which achieves improved statistical multiplexing and resource utilization. When considering a multi-tenant network, virtual resource pools should

not only provide the performance/utilization expected from pooled resources, but should also ensure each tenant the desire degree of isolation and protection from other tenants' traffic as ass the ability to differentiate its customers performance.

As discussed in Chapter 1, in this thesis we consider the optimization *Virtual Resource Pools* (VRPs) and associated joint dynamic resource allocation mechanisms among tenants/slices. Although aggregating resources into VRPs may increase statistical multiplexing through more flexible resource allocation, it also degrades the ability to differentiate the precise allocations given to tenants. In this chapter, we consider optimally partitioning a set of resources into a collection of VRPs which enhances statistical multiplexing while minimizing degradation in the ability of the system to differentiate and isolate slices. To that end, we present an optimization problem, Optimal VRP Partitioning (OVP), aimed at guiding the selection of a partition that realizes the best tradeoffs between these two objectives while ensuring a relaxed notion of isolation among the slices.

3.1 Related Work

As discussed earlier, achieving resource pooling by seamlessly transfering either load or capacity across resources is unrealistic in a wireless network. However, resource pooling can still be achieved through different techniques (for a taxonomy of these techniques in wireless networks the reader is referred to [56]), which we classify as being as one of two types: "*load allocation*" and "*resource allocation*" mechanisms. Load allocation may correspond to dynamic routing policies, such as join the shortest queue, see e.g., [44, 84] or queue with the smallest

expected delay, e.g., [68]. Such mechanims have proven to be very effective resource pooling enablers. Generalizations of these policies, such as the "supermarket model" [79], that consider that users can join the less loaded out of d random resources achieve exponential improvements in the maximum resource load when $d = 2$ with respect of $d = 1$. 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.

Regarding resource allocation, 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 [11] and Generalized Processor Sharing [90] have seen wide applicability in wireless networks. However, managing resource allocation independently at each resource does enable pooling across resources, see e.g. [19, 111]. Consequently, researchers have considered the idea of performing resource allocation across resources [16, 19, 74] and exhibited the effectiveness of resource allocation in enabling resource pooling. The above-mentioned approaches address single-tenant networks, in contrast to our work where we focus on slicing and sharing of resources among multiple tenants. Recently, some extensions of resource allocation mechanisms for muti-tenant wireless networks were studied in [13, 43, 72, 76, 102]. The reader is referred to [93] for a survey on resource slicing techniques for virtual wireless networks.

These multi-tenant resource allocation mechanisms are targeted to analyze the case of 'elastic' users, i.e., whose sojourn time is dependent on the experienced

rate (e.g., users completing a file transfer). In this thesis, instead, we are concerned on 'inelastic' customers whose network activity is independent of their resource allocation (e.g., video, voice and other rate-adaptive user sessions). In Chapter 2 (and in [21, 111]) we proposed a simple Share-Constrained Proportionally Fair mechanism with the premise that tenants are allocated a share of a pool of resources, which is to be redistributed dynamically depending on the system loads. In [111] we showed that this mechanism achieves improved statistical multiplexing, resulting in capacity savings versus per-resource mechanisms such as GPS. Moreover, in [111] we characterized these gains, suggesting that the spatial distribution of the loads impacts the perceived gains as well as the degree of tenants' isolation and performance variability.

Some works have demonstrated that, under mild conditions, using a combination of load and resource mechanisms, a network may indeed behave as a resource pool. For example, [68] shows that in stochastic networks with dynamic routing, in a heavy traffic Brownian motion regime, the network behaves as a single pool. [57] shows a similar result for a network operating under a fair bandwidth sharing resource allocation policy, in a heavy elastic traffic regime in which the average load placed on each resource is approximately equal to its capacity. Similarly, [60] consider a heavy traffic network where resources are shared according to the proportional fairness criterion. Systems where the distribution of users takes a product form allow one to characterize the closeness of the system to resource pooling while [54] quantify the performance benefits of using multi-path flow control to enable resource pooling in stochastic networks.

In this chapter, we do not limit ourselves to heavy load regimes for elastic traffic and we go beyond analyzing how close to a resource pool the network operates. We study how to partition the resources to create virtual pools of jointly allocated resources that work as close as possible to a single pool, considering the tenants per resource shares/requests, mean and variability of the loads and resource capacities. Although optimal network partitioning has the been object of studies for decades in several contexts with different applications [49,87,95,103], this work is, to the authors best knowledge, the first attempt at devising an strategy for optimal VRP partitioning and associated joint resource management.

3.2 Chapter organization

The rest of this chapter is organized as follows. We introduce our system model and the notion of a Virtual Resource Pool (VRP) in Section 3.3, along with some benchmark partitions, including fine grain GPS based resource sharing and coarse grain Complete Pooling (CP). Section 3.4 introduces the Optimal VRP Partitioning problem, i.e., an attempt at determining how the network infrastructure provider should choose a VRP partition of the available resources which ensures multi-tenant isolation while meeting architectural and geographical constraints. In Section 3.5, we show that the Optimal VRP Partitioning problem is NP-Hard and we propose a polynomial time greedy algorithm that determines near optimal partitions. In Section 3.6 we present a performance analysis that characterizes the statistical multiplexing vs differentiation tradeoffs in such networks. Before drawing concluding remarks in Section 3.8, we present a numerical evaluation based

on simulations, which complements the analysis Section 3.7. Proofs of theoretical results in this chapter have been relegated to Section 3.9.

3.3 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 with capacities given by

$$\mathbf{c} = (c_b : \ b \in \mathcal{B}), \tag{3.1}$$

shared by a set $\mathcal{O} = \{1, 2, \ldots, |\mathcal{O}|\}$ of $|\mathcal{O}|$ network tenants (also denoted as *network slices*).

The tenants' traffic load is assumed to be stochastic and the distribution of the random vector $\mathbf{N} = (N_b^o : b \in \mathcal{B}, o \in \mathcal{O})$ characterizes the marginal distribution of the number of active users on the network and

$$\boldsymbol{\rho} = (\rho_b^o : b \in \mathcal{B}, o \in \mathcal{O}) \tag{3.2}$$

denotes the mean loads. The traffic load state at a certain instant is represented by

$$\mathbf{n} = (n_b^o : b \in \mathcal{B}, o \in \mathcal{O}), \tag{3.3}$$

where n_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

$$\mathbf{s} = (\mathbf{s}^o : o \in \mathcal{O}) \quad \text{where} \quad \mathbf{s}^o = (s_{b_1}^o, s_{b_2}^o, \ldots, s_{b_{|\mathcal{B}|}}^o). \tag{3.4}$$

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

$$s_b \triangleq \sum_{o \in \mathcal{O}} s_b^o \leq 1. \tag{3.5}$$

This assumption is made without loss of generality, since the shares/demands correspond to relative quantities across entities contending for resources, and can always be normalized.

3.3.1 Virtual Resource Pools and resource allocation

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

$$\mathcal{P} = \{P_i \mid i = 1, \ldots, |\mathcal{P}|\}. \tag{3.6}$$

Each of the subsets $P_i \subset \mathcal{B}$ of the partition will act as a VRP [21, 111]. The idea underlying multi-tenant sharing of a virtual pool is as follows. From a resource allocation perspective, tenants have a fixed share 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

$$s^o(P_i) \triangleq \sum_{b \in P_i} s_b^o. \tag{3.7}$$

Note that the sum of shares over a pool are no longer 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

$$n^o(P_i) \triangleq \sum_{b \in P_i} n_b^o \tag{3.8}$$

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.,*

$$w^o(\mathbf{n}, P_i) = \frac{s^o(P_i)}{n^o(P_i)}, \quad \forall o \in \mathcal{O},\ P_i \in \mathcal{P}. \tag{3.9}$$

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 a user of slice o is given by

$$f_b^o(\mathbf{n}, P_i) = \frac{w^o(\mathbf{n}, P_i)}{\sum\limits_{v \in \mathcal{O}} w^v(\mathbf{n}, P_i) \cdot n_b^v \cdot \mathbf{1}(n_b^v > 0)}. \tag{3.10}$$

Given that the total capacity of the resource is c_b when the system is in state $\mathbf{n}$ all n_b^o users of slice o at resource $b \in P_i$ are allocated the same rate [1] *$r_b^o(\mathbf{n}, P_i)$ given by*

$$r_b^o(\mathbf{n}, P_i) = c_b \cdot f_b^o(\mathbf{n}, P_i). \tag{3.11}$$

We note that the notion of a VRP represents an abstraction. Indeed, since its underlying physical resources might be at different spatial locations they may not be

[1] Despite in some cases, e.g. wireless networks, the user average peak achievable rate depends on its channel quality, we average out these variations since they occur in shorter time scales than our pooling.

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.

3.3.2 Benchmark allocations

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

1. *Generalized Processor Sharing (GPS)* [90]: This corresponds to the partition of the resources into VRPs each with a single resource each, i.e., $\mathcal{P}^{GPS} = \{\{b\} : b \in \mathcal{B}\}$.

2. *Complete Pooling (CP)*: This corresponds to the partition with a single into a VRP containing all of the resources, i.e., $\mathcal{P}^{CP} = \{\mathcal{B}\}$.

3.3.3 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 as follows*

$$\tilde{\mathbf{s}}^o(P_i) = (\tilde{s}^o_b(P_i) : b \in P_i), \text{ where } \tilde{s}^o_b(P_i) \triangleq \frac{s^o_b}{s^o(P_i)},$$
$$\tilde{\mathbf{n}}^o(P_i) = (\tilde{n}^o_b(P_i) : b \in P_i), \text{ where } \tilde{n}^o_b(P_i) \triangleq \frac{n^o_b}{n^o(P_i)}.$$

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

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

where $s = \sum_{o\in\mathcal{O}, b\in\mathcal{B}} s^o_b$ *is the total share and 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 $\rho = \sum_{o\in\mathcal{O}, b\in\mathcal{B}} \rho^o_b$ *denotes the total system mean load.*

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

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

We adopt the convention that $0/0 = 1$ *if* $n^o_b = n^o(P_i) = 0$. *Note that* $\hat{\mathbf{g}}(\mathbf{n}, \mathcal{P})$ *can also be interpreted as a mixture of the load distributions* $\tilde{n}^o(P_i)$ *with weights* $\hat{s}^o_{P_i}$. [2]

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

$$\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P}) = (\hat{g}_b(\boldsymbol{\rho}, P_i) \ : \ b \in \mathcal{B}) \quad \textit{where} \quad \hat{g}_b(\boldsymbol{\rho}, P_i) \triangleq \sum_{o\in\mathcal{O}} \hat{s}^o_{P_i}\, \tilde{\rho}^o_b(P_i). \qquad (3.12)$$

[2] 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 $\mathcal{P}$.

3.4 VRP partitioning

The main goal of this study is focused to decide how the network infrastructure provider should choose a partition $\mathcal{P}$ of VRPs. Creating such VRPs of many resources enables the ability to absorb bursty traffic variations by exploiting statistical multiplexing – consequently improving the users expected performance. However, pooling may reduce the ability of guarantee each slice a desired degree of protection, e.g., by strictly enforcing s_b^o, the per slices shares, at each resource. Moreover, geographical and architectural network constraints may need to be incorporated which limit the resources that can be pooled together. We propose to have the Network Infrastructure Provider (NIP) to choose a VRP partition based on maximizing an associated expected network utility subject to constraints which ensure inter-slice isolation and incorporate architectural/geographical system constraints.

3.4.1 Stochastic network utility

The optimal VRP partition will be set to maximize a certain network statistic, 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 to generate a meaningful 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 [61], 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 partition P_i, i.e.,

$$U^o(P_i) = \mathbb{E}\left[\sum_{b\in P_i} \frac{N_b^o}{\mathbb{E}[N^o(P_i)]} \log\left(r_b^o(\mathbf{N}, P_i)\right)\right]$$
$$= \mathbb{E}\left[\sum_{b\in P_i} \frac{N_b^o}{\rho^o(P_i)} \log(r_b^o(\mathbf{N}, P_i))\right].$$

To deal with the case where the number of active users is zeros, i.e., $N_b^o = 0$ we have used the convention (see e.g., [25]) $0 \cdot \log(0) \triangleq 0$. Recall that a "typical" users here should be viewed as a randomly selected user of slice o on partition 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 partition's resources.

Then, the overall expected network utility is given by a weighted combination of the slices' utilities per partition. 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 $\rho^o(P_i)\frac{s^o(P_i)}{\rho^o(P_i)} = s^o(P_i)$, given an overall utility

$$\mathcal{U}(\mathcal{P}) = \sum_{P_i\in\mathcal{P}} \sum_{o\in\mathcal{O}} \hat{s}_{P_i}^o \, U^o(P_i) = \sum_{o\in\mathcal{O}} \mathcal{U}^o(\mathcal{P}). \tag{3.13}$$

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

$$\mathcal{U}^o(\mathcal{P}) = \sum_{P_i\in\mathcal{P}} \hat{s}_{P_i}^o \, U^o(P_i) \tag{3.14}$$

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' typical utility.

3.4.2 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 $\sum_o s_b^o \leq 1$). However, in a multi-resource network, the inability of the resource allocation to adapt to the traffic variations indicates that fair allocation schemes may consider a network-wide view [19] 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 performance isolation. Hereby, we define a VRP notion of protection.

Definition 5. ***(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*

$$\sum_{b \in P_i} n_b^o \log \left(c_b \cdot f_b^o(\mathbf{n}, P_i) \right) \geq \sum_{b \in P_i} n_b^o \log \left(c_b \cdot \frac{s_b^o}{n_b^o} \right) \tag{3.15}$$

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. (3.15) *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 partition) is presented in the following lemma.

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

$$\mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) = -\sum_{b \in P_i} \frac{s_b^o}{s^o(P_i)} \log\left(\frac{s_b^o}{s^o(P_i)}\right) \geq \log(s(P_i)), \tag{3.16}$$

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 demand on the partition.*

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

$$\mathcal{C}_p^o = \{\mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) \geq \log(s(P_i)), \forall P_i \in \mathcal{P}\} \tag{3.17}$$

We note that protection only depends on the share distribution of slice o, and aggregated shares of the slices on each pool, and some 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 [31]

$$0 \leq \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{o}}(P_i)\right) \leq \log(|P_i|).$$

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 6. *(**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*

$$\mathcal{C}_p = \bigcap_{o \in \hat{\mathcal{O}}} \mathcal{C}_p^o. \tag{3.18}$$

The condition in Eq. (3.15) is displayed in Figure 3.1 for a VRP composed by 2 resources, as a function of the share distribution of the slice, given that the total

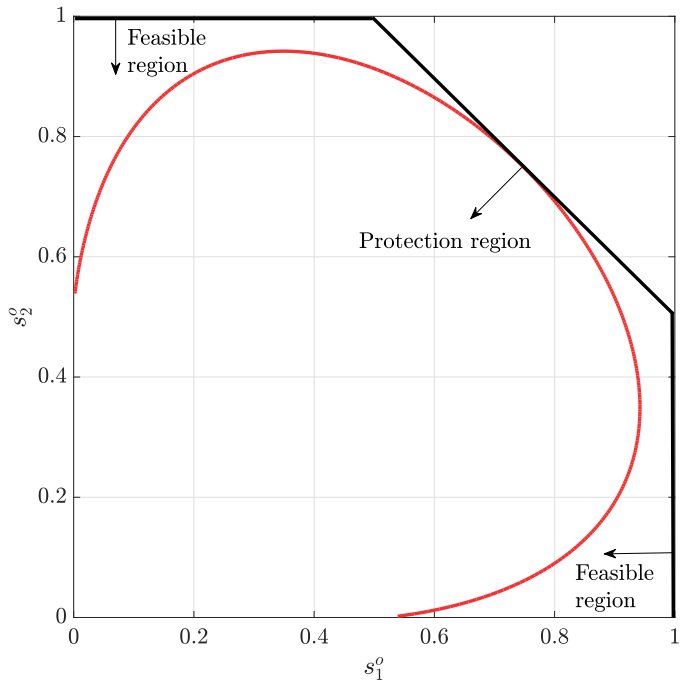

Figure 3.1: Protection range for different values of s_b^o and $s(P_i) - s^o(P_i) = 0.5$.

share of the other slices is equal to $s(P_i) - s^o(P_i) = 0.5$. The feasible region determines the possible values in the share distribution of slice o such that the condition in Eq. (3.6) is satisfied.

3.4.3 Design constraints

Even with the protection constraints satisfied, some partitions may be impractical/inefficient for the NIP. Realizing VRPs requires an exchange of information within the resources, in order to establish the joint resource allocation since it is necessary that every resource in a VRP knows the total number of active users

of each slice. This exchange of information, may impose some architectural or design constraints. For instance, a virtual controller may have capacity to coordinate a maximum number of resources or the delay for information sharing may make virtual pooling of distant resources unfeasible.

3.4.3.1 Pooling management capacity constraints

To capture design constraints associated with the limitations of the architecture in terms of pooling management capacity, we will define the following constraint

$$\mathcal{C}_c = \{ \mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid |P_i| \leq \bar{K}, \forall P_i \in \mathcal{P} \}. \tag{3.19}$$

where $\bar{K}$ represents the maximum number of resources that the NIP can pool together given the network information sharing capacity.

3.4.3.2 Connectivity and locality constraints

In some settings, it may be desirable to create VRPs 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 possible architectural requirement on the partition could be that $G_i(N, E)$ are connected subgraphs, ensuring that paths to distribute information are available and/or the associated nodes are geographical networks to

each other. We will abstract these constraints as follows

$$\mathcal{C}_l = \{\mathcal{P} \in \mathcal{P}_{\mathcal{B}} \mid p(G_i(P_i, E_i)) = 1,\ \forall P_i \in \mathcal{P}\}. \tag{3.20}$$

where $p(G_i(P_i, E_i))$ is equal to 1 if the subgraph is connected. Note that, although abstracted here for clarity purposes, the connected component restriction can be formally presented as a set of linear constraints by using an exponential set of combinations of the edges [55] or by introducing flow variables [82]. Some geographical/local constraints are reasonable to be expected from the original graph and can be enclosed by the proper setting of the edges. For instance, it is expected that in wireless networks base stations can belong to the same pool if they share a cell edge, resulting in a planar graph or if they share an direct communication link (e.g., an X2 interface) which will result in a non-fully connected graph, while in some other environments there may be no constraints imposed resulting into a fully connected graph.

3.4.4 Optimal VRP Partitioning

Joining the constraints from previous subsections, we can define the partition constraints as

$$\mathcal{C} = \mathcal{C}_p \cap \mathcal{C}_c \cap \mathcal{C}_l \tag{3.21}$$

where $\mathcal{C}_p,\ \mathcal{C}_c,\ \mathcal{C}_l$ are defined in Eqs. (3.18), (3.19) and (3.20) respectively. We can write the optimal spatial pooling problem as the following optimization problem.

Definition 7. ***(Optimal VRP Partitioning Problem (OVP))*** *The Optimal VRP Par-*

tition is given by

$$\max_{\mathcal{P}} \quad \{\, \mathcal{U}(\mathcal{P}) \,\big|\, \mathcal{P} \in \mathcal{C} \,\}. \tag{3.22}$$

Unfortunately, finding the solution of OVP is a complex problem from several reasons: (i) the possible number of partitions that need to be considered increase exponentially and (ii) evaluating the utility function implies finding the expected value of a non-linear function of random variables, which is per se a hard problem. Sections 3.5 and 3.6 discuss how to solve these two issues, respectively.

3.5 Algorithm Design

As already mentioned in Section 3.4.4, the possible number of feasible of partitions that need to be considered in order to find the optimal pooling increase exponentially with the number of resources (in accordance to the Bell numbers [17]). 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, where it shows that the problem is NP-Hard.

Theorem 6. *Optimal VRP Partitioning is NP-Hard.*

3.5.1 Greedy algorithm for OVP

In order to overcome the high complexity of an exhaustive search associated with OVP, we propose a greedy algorithm based on the idea of cost-benefit greedy algorithm [65].

The algorithm is initialized with by 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., $\mathcal{H}(\mathcal{P}) = \left(\sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} H(\tilde{\mathbf{s}}^o(P_i)) \right)^{-1}$. Therefore, the gain over cost ratio of joining pools P_i and P_j, is

$$\delta\mathcal{U}(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}}) = \frac{\mathcal{U}(\hat{\mathcal{P}}^{i,j}) - \mathcal{U}(\hat{\mathcal{P}})}{\mathcal{H}(\hat{\mathcal{P}}^{i,j})}$$
$$\text{where } \hat{\mathcal{P}}^{i,j} = \{\hat{\mathcal{P}} \setminus \{P_i, P_j\}\} \cup \{P_i \cup P_j\}$$

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 $\mathcal{U}(\hat{\mathcal{P}}^{i,j})$, which can be performed by using Monte Carlo sampling methods or via appropriate approximations, see e.g., Section 3.6.

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 $\mathcal{P}^{CP}$. The pseudocode for the proposed algorithm is exhibited in Algorithm 2.

3.5.2 Greedy algorithm performance

The proposed algorithm is ensured to complete with a worst-case time complexity of $O(|\mathcal{B}|^3)$, since it can do at most $|\mathcal{B}| - 1$ merges before reaching the stopping condition, each of which requires one to check at most $|\mathcal{B}|^2$ possible merges.

Algorithm 2: Greedy OVP

Input: $\mathbf{s}, \mathbf{c}, \boldsymbol{\rho}, \mathcal{G}$
Output: $\hat{\mathcal{P}}$
Initialization: $\hat{\mathcal{P}} = \mathcal{B}$
while $|\hat{\mathcal{P}}| > 1$ **do**
  for $P_i \in \hat{\mathcal{P}}$ **do**
    for $P_j \neq P_i \in \hat{\mathcal{P}}$ **do**
      $\hat{\mathcal{P}}^{i,j} = \{\hat{\mathcal{P}} \setminus \{P_i, P_j\}\} \cup \{P_i \cup P_j\}$
      if $\hat{\mathcal{P}}^{i,j} \in \mathcal{C}$ **then**
        $\delta\mathcal{U}(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}}) = \frac{\mathcal{U}(\hat{\mathcal{P}}^{i,j}) - \mathcal{U}(\hat{\mathcal{P}})}{\mathcal{H}(\hat{\mathcal{P}}^{i,j})}$
      else
        $\delta\mathcal{U}(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}}) = 0$
  if $\max(\delta\mathcal{U}(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}})) \leq 0$ **then**
    return $\hat{\mathcal{P}}$
  else
    $i^*, j^* = \arg\max\left(\delta\mathcal{U}(\hat{\mathcal{P}}^{i,j}, \hat{\mathcal{P}})\right)$
    $\hat{\mathcal{P}} = \{\hat{\mathcal{P}} \setminus \{P_{i^*}, P_{j^*}\}\} \cup \{P_{i^*} \cup P_{j^*}\}$

Observing that the most complex operation of algorithm is the computation of the the expected network utility of a VRP; the algorithm can be greatly improved by storing and reusing VRP utilities. By doing this, the first iteration of the algorithm requires to compute $\binom{|\mathcal{B}|}{2}$ pool utilities, while the i^{th} iteration only requires the computation of $(i-1)$ pool utilities of the new merged VPP with each of the remaining subsets, using the precached pool utilities to evaluate the rest of required utilities. Therefore, the worst-case number of VRP utilities computations required is

$$\binom{|\mathcal{B}|}{2} + \sum_{i=2}^{|\mathcal{B}|}(i-1) = |\mathcal{B}|^2 - |\mathcal{B}| = O(|\mathcal{B}|^2).$$

This makes the greedy algorithm feasible at the time scales we envisage optimization of VRP partitioning being recomputed, e.g., hourly/daily. The numerical eval-

uations performed in Section 3.7.1 show that the algorithm successfully finds the optimal partition in most of the cases at a much lower computational complexity than brute force evaluation.

3.6 Utility approximation and analysis

One of the challenges with optimizing the proposed expected network utility $\mathcal{U}(\mathcal{P})$ is the evaluation of the function itself, i.e., Eq. (3.13). As mentioned earlier, this involves computing the expected value of a nonlinear function over the distribution of $\mathbf{N}$, which itself needs to be estimated or modeled.

In this study, we will use a simple model for the number of active users per slice and resource. We assume that customers of each slice o arrive at each resource b according to a Poisson process with rate γ_b^o. Upon arrival, each customer has an independent sojourn time with mean μ_b^o. Customer mobility is assumed to follow a fixed routing matrix, which captures either a departure from the network or a handoff to another resource. In our considered model, both the sojourn time and the mobility of a customer are assumed to be independent of the rate allocation received by the customer throughout its sojourn. Thus, our setting might be viewed as associated with a "well engineered" network supporting inelastic and/or rate adaptive customers, where a customer utility increases as she receives a greater rate, but its departure time is fixed. Examples of rate adaptive sessions are, for example, live video streaming and phone and video calls.

As explained in [111] this model corresponds to a multi-class (multi-tenant) network of M/GI/∞ queues, which has a product-form stationary distribution [59]

where the number of customers of slice o at station b are mutually independent and Poisson distributed with a mean ρ_b^o which depends on the arrival rates, the sojourn times and the routing matrix.

This traffic model is relatively simple in that it is only necessary to consider one parameter per slice and resource, i.e., the mean intensities ρ_b^o for the Poisson distributed number of active users that slice o has in station b in steady state. This assumption is formalized in the following

Assumption 1. ***(Poisson and independent loads network)*** *We assume a network supporting a stochastic number of active users* $\mathbf{N} = (N_b^o : o \in \mathcal{O}, b \in \mathcal{B})$ *independent across slices and resources and Poisson distributed with finite and positive means* $\boldsymbol{\rho} = (\rho_b^o : b \in \mathcal{B}, o \in \mathcal{O})$, *i.e., where* $N_b^o \sim$ *Poisson*(ρ_b^o).

Under Assumption 1 one can develop an approximation for the expected network utility via a Taylor expansion. This result is stated in the following proposition.

Proposition 1. *Under Assumption 1 the expected network utility* $\mathcal{U}(\mathcal{P})$ *for a partition* $\mathcal{P}$ *is equal to:*

$$\mathcal{U}(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} \hat{u}_b^o(\boldsymbol{\varrho}_b^o, P_i) + \sum_{P_i \in \mathcal{P}} \sum_{\substack{o,o' \in \mathcal{O} \\ b,b' \in P_i}} \frac{\sigma_{b,b'}^{o,o'}}{2} \frac{\partial^2 \left(\hat{u}_b^o(\mathbf{x}, P_i)\right)}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} + R, \qquad (3.23)$$

where the function $\hat{u}_b^o(\mathbf{x}, P_i)$ *is given by*

$$\hat{u}_b^o(\mathbf{x}, P_i) = \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \left(1 - e^{-\rho_b^o}\right) \, x_b^o \, \log\left(r_b^o(\mathbf{x}, P_i)\right),$$

and $r_b^o(\mathbf{x}, P_i)$ *is the extension of Eq.* (3.11) *to continuous arguments. The vectors* $\boldsymbol{\varrho}_b^o = (\varrho_{b,b'}^{o,o'} : o' \in \mathcal{O}, b' \in \mathcal{B})$ *for each* $o \in \mathcal{O}$ *and* $b \in \mathcal{B}$ *are given by*

$$\varrho_{b,b'}^{o,o'} = \begin{cases} \rho_{b'}^{o'} \left(1 - e^{-\rho_{b'}^{o'}}\right)^{-1} & \textit{if } o' = o, b' = b; \\ \rho_{b'}^{o'} & \textit{otherwise}. \end{cases}$$

and

$$\sigma_{b,b'}^{o,o'} = \begin{cases} \varrho_{b,b'}^{o,o'} \left(1 - \frac{\rho_b^o}{e^{\rho_b^o}}\right) & \textit{if } o' = o, b' = b; \\ \varrho_{b,b'}^{o,o'} & \textit{otherwise}. \end{cases}$$

Finally, R *corresponds to the expected value of the remainder terms in the Taylor approximations.*

In the sequel, we will use the approximation of Eq. (3.23), ignoring the remainder term R (which is quite complex, see Eq. (3.31)), to rank the different partitions. This approximation is easier to evaluate than Monte Carlo sampling the distribution of $\mathbf{N}$ to compute the expected network utility.

Still, to better understand the characteristics of partitions that lead to high expected network utility, we consider a sequence of networks, indexed by β, as follows.

Assumption 2. ***(Linear scaling)*** *Consider 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 number 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.,*

$$N_b^{o,(\beta)} \sim \textit{Poisson}(\beta \cdot \rho_b^o)$$

and the resource capacities $\mathbf{c}^{(\beta)} = \beta\mathbf{c}$ *such that* $c_b^{(\beta)} = \beta c_b$. *We let* $\mathcal{U}^{(\beta)}(\mathcal{P})$ *denote the expected network utility, given Eq.* (3.13)*, of the* β^{th} *network.*

Theorem 7. *Under Assumption 2, the expected network utility under partition* $\mathcal{P}$ *is given by*

$$\mathcal{U}^{(\beta)}(\mathcal{P}) = \log\left(\frac{c}{\rho}\right) + \mathrm{D}(\mathcal{P}) - \mathrm{M}(\mathcal{P}) - \frac{S(\mathcal{P})}{\beta} + o\left(\frac{1}{\beta}\right), \tag{3.24}$$

where $\mathrm{D}(\mathcal{P}) = \mathrm{D}_{KL}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right)$ *and* $\mathrm{M}(\mathcal{P}) = \mathrm{D}_{KL}\left(\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P})||\hat{\mathbf{c}}\right)$; *where* D_{KL} *stands for the Kullback-Liebler divergence [66].*

Also,

$$\mathrm{S}(\mathcal{P}) = \langle \hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P})\rangle + \langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P})\rangle \tag{3.25}$$

where $\mathbf{q}(\mathcal{P}) = \left((\rho^o(P_i))^{-1} : o \in \mathcal{O}, P_i \in \mathcal{P}\right), \quad \mathbf{h}(\mathcal{P}) = (h_b^o(\mathcal{P}) : o \in \mathcal{O}, b \in \mathcal{B}),$

$$h_b^o(\mathcal{P}) = \sum_{b' \in \mathcal{B}} \frac{\partial^2 \left(\bar{g}_{b'}(\mathbf{x}, P_i)\, \log\left(\hat{g}_{b'}(\mathbf{x}, P_i)\right)\right)}{\partial (x_b^o)^2}\bigg|_{\boldsymbol{\rho}_b^o}$$

and $\bar{g}_{b'}(\mathbf{x}, P_i) = \sum_{v \in \mathcal{O}} \frac{\hat{s}^v(P_i)}{\rho^v(P_i)} x_{b'}^v.$

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

Remark 2. The utility $\mathcal{U}^{(\beta)}(\mathcal{P})$ 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. (3.24) in more detail

1. The **perfect pooling utility** $\log(c/\rho)$ corresponds to the utility of a system where the total network capacity c is pooled and equally divided among its mean total number of users ρ across all slices.

2. 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 and loads

$$\hat{\mathbf{s}}(\mathcal{P}) = \left(\frac{s^o(P_i)}{s} : o \in \mathcal{O}, P_i \in \mathcal{P}\right),$$

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

distributions coincide. When the two distributions diverge, the term increases resulting in slice differentiation gains relative to $\log(c/\rho)$.

3. 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 and normalized capacity distributions

$$\hat{\mathbf{g}}(\boldsymbol{\rho}, \mathcal{P}) = \left(\hat{g}_b(\rho, P_i) = \sum_{o \in \mathcal{O}} \hat{s}^o_{P_i} \tilde{\rho}^o_b(P_i) : b \in \mathcal{B}\right),$$
$$\hat{\mathbf{c}} = \left(\frac{c_b}{c} : b \in \mathcal{B}\right),$$

are equal, otherwise the losses increase as they diverge.

4. 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 the Poisson distribution as $\beta \to \infty$ the number of active users concentrates around the mean.

Let us briefly consider various network scenarios. First, a symmetric network, second a share, load and capacities proportional network and finally, the general setting.

(Symmetric networks) In this setting we have that for all $o \in \mathcal{O}, b \in \mathcal{B}$,

$$\rho_b^o = \frac{\rho}{|\mathcal{O}||\mathcal{B}|},\ s_b^o = \frac{s}{|\mathcal{O}||\mathcal{B}|} \text{ and } c_b = \frac{c}{|\mathcal{B}|}.$$

Under the linear scaling the expected network utility is given by

$$\mathcal{U}^{(\beta)}(\mathcal{P}) = \log\left(\frac{c}{\rho}\right) - \frac{1}{\beta\rho}\left(|\mathcal{B}| + |\mathcal{P}| \cdot (|\mathcal{O}| - 1)\right) + o\left(\frac{1}{\beta}\right). \tag{3.26}$$

In this scenario, $\mathrm{D}(\mathcal{P}) = \mathrm{M}(\mathcal{P}) = 0$, so the network acts as a pool of resources where each user receives on average equal fraction of the sum network capacity c, i.e., a typical user has a utility of $\log(c/\rho)$. However, the network experiences stochastic pooling losses with respect to $\log(c/\rho)$ due to variability in the load. These losses that are minimized when $\mathcal{P} = \mathcal{B} = \mathcal{P}^{(CP)}$, i.e., $|\mathcal{P}| = 1$, as presented in Eq. (3.26).

The network utility gain of CP versus GPS is

$$\Delta^{GPS}(\mathcal{P}) = \mathcal{U}^{(\beta)}(\mathcal{P}^{CP}) - \mathcal{U}^{(\beta)}(\mathcal{P}^{GPS}) \approx \frac{(|\mathcal{O}| - 1)(|\mathcal{B}| - 1)}{\beta\rho} + o\left(\frac{1}{\beta}\right),$$

which increases with the number of resources and the number of slices and decrease as the mean network load $\beta\,\rho$ increases.

(Proportional networks) A similar trend can be observed in a setting where the traffic loads are proportional to the shares, i.e., $\rho_b^o = \gamma s_b^o, \forall o \in \mathcal{O}, b \in \mathcal{B}$ and the

load per station is proportional to the network capacities $\rho_b = \sum_{o \in \mathcal{O}} \rho_b^o = \delta c_b$. Since $\mathrm{D}(\mathcal{P}) = \mathrm{M}(\mathcal{P}) = 0$, the utility is then given by

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

In this scenario, and the stochastic pooling losses $\mathrm{S}\left(\mathcal{P}\right)$ take a more complex form that benefits from the aggregation of stations into virtual pools, alike in the symmetric network to exploit statistical multiplexing.

(General network) For the general case, the expected network 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 $\mathrm{D_{KL}}\left(\hat{\mathbf{s}}(\mathcal{P}) || \hat{\boldsymbol{\rho}}(\mathcal{P})\right) = 0$ 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

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

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. However, in cellular networks, certain resource capacities transferability can be achieved in several ways as for example by having a C-RAN [26] that use its computational capabilities to perform Baseband Unit Pool Planning to align the capacities [106] and/or by appropriate admission control of the number of active users.

2. If resource capacities were transferable among resources or were engineered to coincide with the share weighted mean traffic loads, then $\mathrm{M}(\mathcal{P}) = 0$ and the expected network utility is given by

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

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*

$$\bar{\mathcal{U}} = \log\left(c/\rho\right) + \mathrm{D}_{KL}\left(\hat{\mathbf{s}}(\mathcal{P}^{GPS})||\hat{\boldsymbol{\rho}}(\mathcal{P}^{GPS})\right).$$

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

Fact 3. *The term* $\mathrm{S}(\mathcal{P}) = \langle \hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P}) \rangle + \langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P}) \rangle$, *where* $\langle \hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P}) \rangle$ *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 the main observations regarding the character of optimal VRP partitioning.

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 a differentiation.

3.7 Performance evaluation

Next we evaluate optimal VRP partitions through a set of numerical evaluations. We will first study a collection of small scenarios for which the optimal

partitions can be computed and the results are easy to interpret. This is followed by a set of more realistic and larger network scenarios, to understand the potential gains of our pooling solution. To evaluate the performance, instead of looking at the utility difference, which is hard to interpret in terms of performance improvement, we will define the *effective capacity savings*.

Definition 8. *We define the* ***effective capacity savings*** $\Delta(\mathcal{P}, \hat{\mathcal{P}})$ *of a VRP partition* $\mathcal{P}$ *with respect to* $\hat{\mathcal{P}}$ *as the extra percent of capacity at each resource that the network under VRP partition* $\hat{\mathcal{P}}$ *would require in order to achieve the same expected network utility as under* $\mathcal{P}$*.*

3.7.1 Numerical evaluation of synthetic scenarios

We shall first consider optimal VRP pooling for a stochastic system with 6 resources, fully interconnected (i.e., where no geographical constraints are imposed) and 3 slices and where the number of active users N_b^o are stochastic following a Poisson distribution with mean ρ_b^o. Initially, we will consider both shares and capacity distributions uniform. Unless otherwise specified, the shares of the slices are $s_b^o = 1/5$ for all slices and resources and the capacities for all stations are set to $c_b = L \cdot 10$.

We will consider four different scenarios, that vary in their mean load distributions and where L is a configurable parameter that determines the mean load per station. The illustrative scenarios used in this subsection are detailed next and a depiction for $L = 4$ is displayed in Figure 3.2, where each stacked bar represents the mean load per slice at each resource.

1. **Uniform loads:** all slices have the same mean load at all resources.

2. **Complementary loads:** slices' mean load distributions are complementary with $\eta\%$ of their load is concentrated in 2 out of the 6 resources. Unless otherwise specified, the value of $\eta = 95\%$

3. **Aligned hotspots:** slices' mean load distributions are aligned and concentrated in 2 hotspot resources, that accumulate 50% of the load.

4. **Mixed hotspots:** combination of complementary loads and hotspots where the mean load distributions are given by

$$\boldsymbol{\rho}^1 \approx \frac{L}{4} \cdot (0.03, 2.98, 0.44, 0.72, 2.05, 1.78)$$

$$\boldsymbol{\rho}^2 \approx \frac{L}{4} \cdot (2.74, 0.58, 0.03, 3.46, 0.67, 0.53)$$

$$\boldsymbol{\rho}^3 \approx \frac{L}{4} \cdot (1.16, 1.62, 1.08, 0.03, 2.18, 1.92)$$

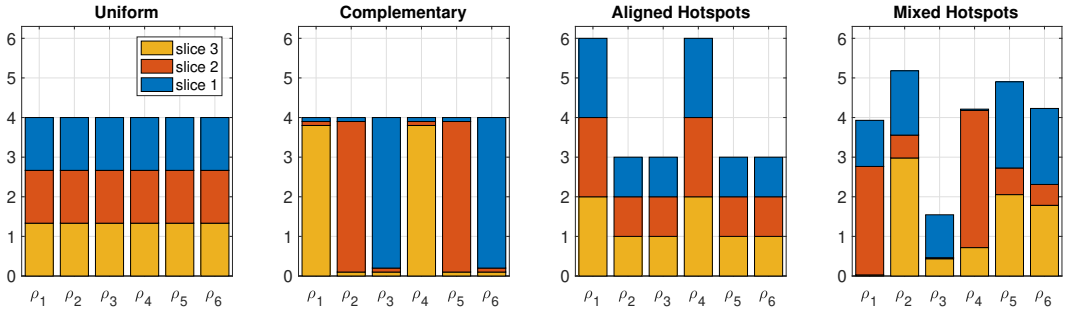

Figure 3.2: Load distributions of the illustrative scenarios for $L = 4$.

3.7.1.1 Optimal partitions for uniform shares

Given the small size of these network scenarios one can easily evaluate the optimal VRP partition by evaluating all possible partitions. Our greedy algorithm, was able to determine the optimal partition for all cases, evaluated at different load values ($2 \leq L \leq 40$), except for the case with $L = \{9, 10, 11\}$ associated with the mixed hotspot scenario. The resulting optimal partitions are as follows

1. **Uniform loads:** The optimal partitions, for all values of L is $\mathcal{P}^{CP}$, as expected since this scenario coincides with the symmetric case discussed in Section 3.6.

2. **Complementary loads:** The optimal solution, for all values of L is to create VRPs with **similar shares/loads distributions**, i.e., $\mathcal{P}^* = \{\{1, 4\}, \{2, 5\}, \{3, 6\}\}$. Joining resources with identical load distribution is beneficial in this case, since it enables the network to achieve higher differentiation gains, and having a misalignment loss of 0 while exploiting statistical multiplexing.

3. **Aligned hotspots:** The optimal solution, for all values of L is again to create VRPs with **similar share/load distributions**, i.e., $\mathcal{P}^* = \{\{1, 4\}, \{2, 3, 5, 6\}\}$. Again, this allows to achieve higher differentiation gains, although now there are losses due to load misalignment loss.

4. **Mixed hotspots:** In this case, there are different solutions depending on L. For low values of load where $L \leq 11$, the optimal solution is $\mathcal{P}^* = \{\{1, 4\}, \{2, 5, 6\}, \{3\}\}$ while for $L > 11$, $\mathcal{P}^* = \{\{1\}, \{3\}, \{4\}, \{2, 5, 6\}\}$.

We can see that, the resources with similar load distributions, i.e., 2,5,6 are joined together in any case and 3, since is not like any other resource is isolated. By contrast, 1 and 4 do have relatively similar distributions, as the variability of the load decreases, the statistical multiplexing load does not compensate for the differentiation losses and so resources are not pooled together.

3.7.1.2 Pooling capacity savings

Although the optimal partitions provide an intuition towards the character of good VRP partitions, it is unclear what performance gains are brought by optimal VRP pooling.

Figure 3.3 shows the effective capacity savings as a function of the mean load per station (L) for the 4 scenarios studied. As might be expected, as the mean load increases, part of the gain (corresponding to the multiplexing gains) disappear so higher gains are found for moderate levels of L. Indeed, as L gets close to 0, the gains over GPS also reduce since the probability of not having any user is increased, reducing the likelihood that the dynamic allocation in the virtual pool differ at all from GPS. The minimum gains are attained in the aligned and uniform loads scenarios, where maximum savings of 5.5% and 7.5% are achieved, respectively. The complementary loads scenario exhibits capacity savings up to a 12.5% while the mixed hotspots reaches more than 25% gain for low loads.

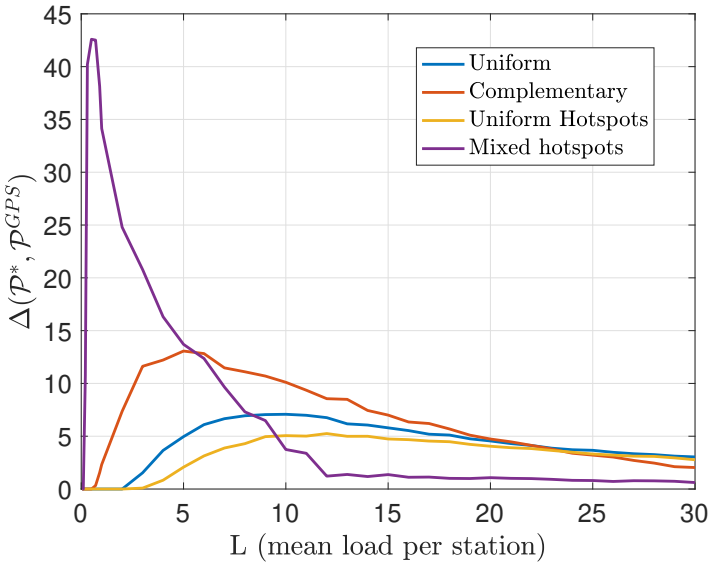

Figure 3.3: Pooling capacity savings of optimal VRP partition vs GPS for 4 network scenarios and varying load L.

3.7.1.3 Optimal partitions for shares/loads proportional networks

Next, we will consider same load and capacity scenarios assuming that the shares s_b^o, instead of being uniform are proportional to the load distributions. More formally, for all $o \in \mathcal{O}$ and $b \in \mathcal{B}$ the shares are given by $s_b^o = \gamma \cdot \rho_b^o$. Without taking into account the protection constraints, the optimal partition for all scenarios given any value of $L > 3$ is $\mathcal{P}^* = \mathcal{P}^{CP}$, i.e., to create **a single pool** with all 6 stations together. Since shares and loads are proportional, all the first order gains are independent of the partition as discussed in Section 3.6 and the gains only come from multiplexing gains, and therefore are smaller compared with the previous scenarios.

As discussed in Remark 1, the non-uniformity of the shares can result in a protection constraint violation, which depends on the relative share distribution and

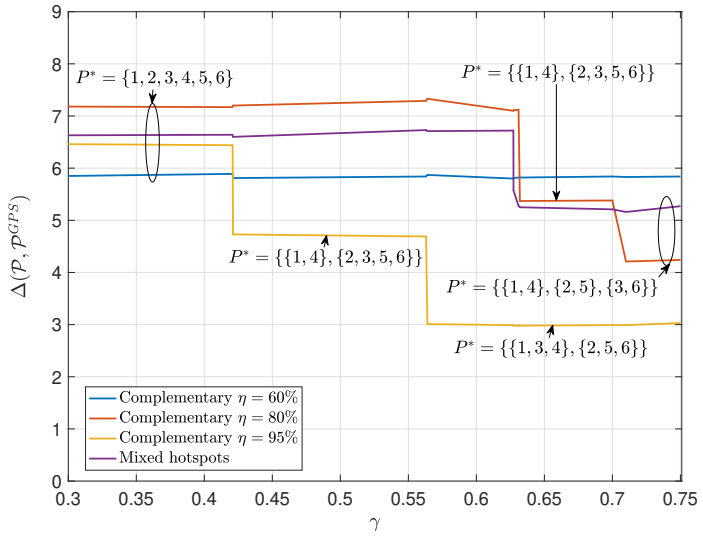

Figure 3.4: Capacity savings of optimal VRP partition vs GPS with proportional rate-share scenarios as a function of the total share for L=5.

the total share (controlled by the parameter γ), according to Eq. (3.16). While for the uniform and aligned hotspots scenarios, $\mathcal{P}^* = \mathcal{P}^{CP}$ is always feasible this is not the case for the complementary and mixed load scenarios.

Figure 3.4 exhibits the capacity savings as well as the optimal partitions as function of γ for 3 different complementary scenarios, with $\eta = \{60, 80, 95\}$ as well as the mixed hotspot scenario. We can observe that if the overall share, i.e., γ is low, $\mathcal{P}^{CP}$ is feasible but as γ grows the optimal feasible partition changes including several VRPs and a capacity savings reduction is expected, resulting into a decreased gain in favor of the slices protection as shown in Figure 3.4.

3.7.2 Performance evaluation in realistic scenarios

To complement the previous results, 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 [18] and (ii) SLAW model [71] which is a human walk based mobility model which generates uneven load distributions across space. These combinations of users generate stations with different load distributions. We explored 4 different scenarios described next and for which the average load distributions per resource are displayed in Figure 3.5:

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

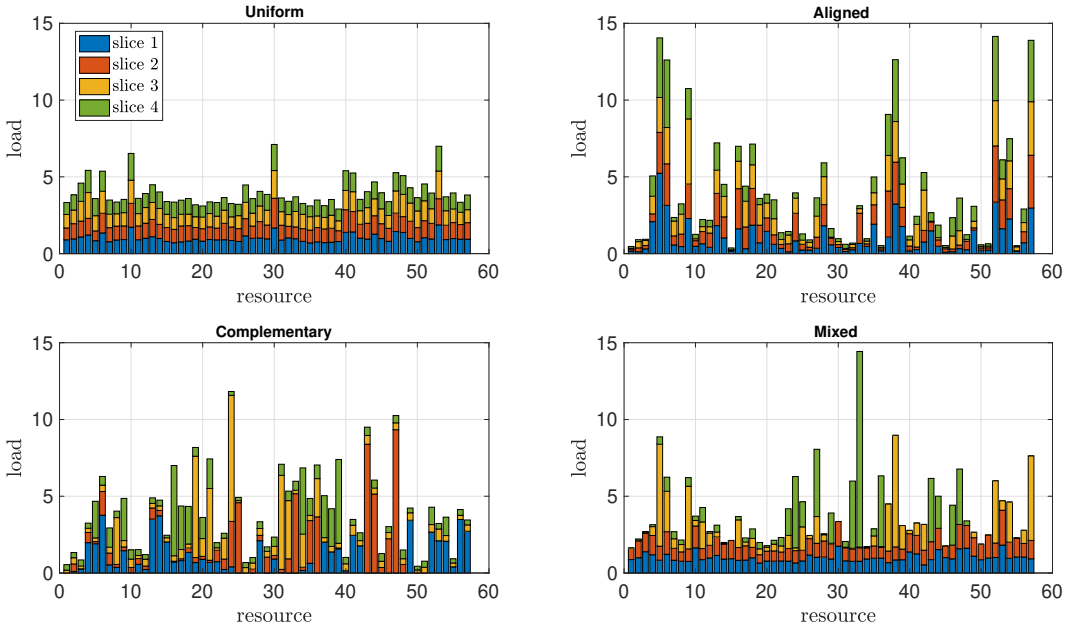

Figure 3.5: Load distribution of the illustrative scenarios.

3.7.2.1 Capacity savings for uniform shares

In Figure 3.6, we display the capacity savings of the optimal VRP partition versus GPS and CP for the 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 scenarios and the gains can go up to 50%. With respect to CP, the capacity savings (except for the uniform case) are very high since creating a big partition with all the resources eliminate the ability of slices to differentiate, resulting in resource allocations which exploit statistical multiplexing but are not able differentiate slices' users performance.

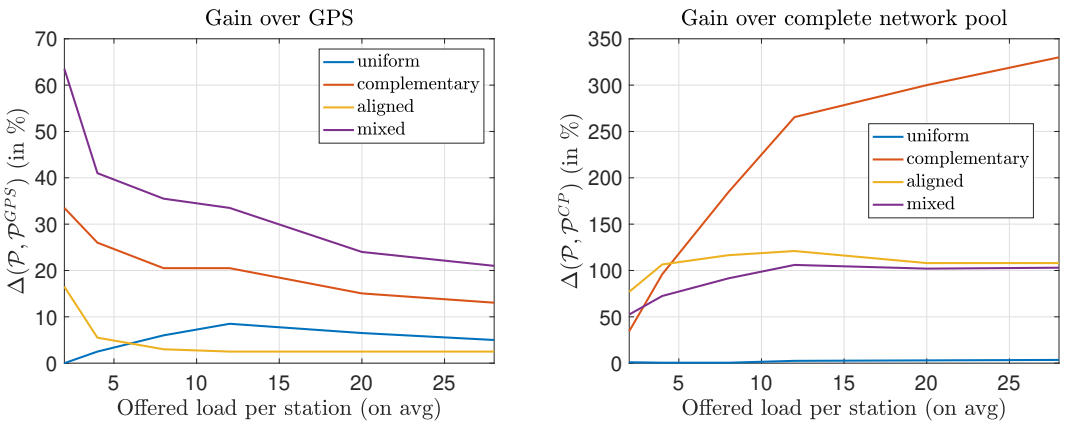

Figure 3.6: Capacity savings for the different scenarios vs GPS and CP for uniform shares as a function of the mean offered load per station

3.7.2.2 User utility in proportional shares/loads scenarios

To complement the previous results, we evaluated scenarios with homogeneous resource capacities $\mathbf{c} = \mathbf{1}$ and 4 slices. Each slice o requests an equal share per mean customer load denoted by γ^o, i.e.,

$$s_b^o = \gamma^o \rho_b^o \quad \text{where} \quad \begin{cases} \gamma^1 = 0.25/\bar{\rho}_b, \\ \gamma^2 = 0.18/\bar{\rho}_b, \\ \gamma^3 = 0.12/\bar{\rho}_b, \\ \gamma^4 = 0.09/\bar{\rho}_b. \end{cases}$$

where $\bar{\rho}_b = \max_{b \in \mathcal{B}} \sum_{o \in \mathcal{O}} \rho_b^o$. So we have slices which attempt to differentiate their customer's performance through requesting different shares per mean loads across the network. The network is evaluated for the 4 previously described load distributions – uniform, aligned, complementary and mixed; with an average offered load per station and slice of $5/4$, i.e., the total offered load of $L = 5$.

In the four scenarios evaluated, the optimal partition corresponds to Com-

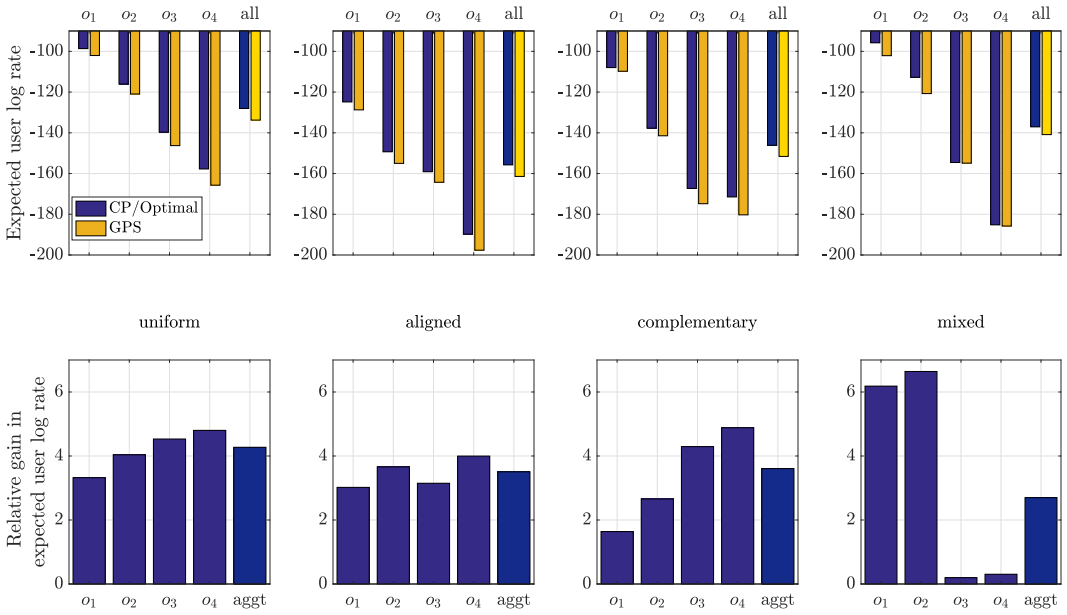

Figure 3.7: Expected user utilities and relative gains of VRP partitioning

plete Pooling. For these scenarios we evaluated the expected user log rate per slice, i.e.,

$$\mathcal{R}^o(\mathcal{P}) = \mathbb{E}\left[\sum_{P_i \in \mathcal{P}} \sum_{b \in \mathcal{B}} N_b^o \log(r_b^o(\mathbf{N}, P_i))\right]$$

as well as the overall expected user log rate $\mathcal{R} = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \mathcal{R}^o(\mathcal{P})$; for both CP and GPS. The results for this metric across the 4 scenarios and 4 slices are displayed in Figure 3.7 (top). From the results, we can conclude that, as expected, the expected user log rate utility increases with γ^o, both for CP and GPS and for all scenarios. Moreover, the gains obtained by CP are positive for every slice, i.e., no slice is harmed to benefit other.

To understand which slices see a greater gain under optimal VRP partition-

ing, we evaluated the relative gain of CP versus GPS as

$$\Delta\mathcal{R}^o(\mathcal{P}^{CP}, \mathcal{P}^{GPS}) = 100 \cdot \frac{\mathcal{R}^o(\mathcal{P}^{CP}) - \mathcal{R}^o(\mathcal{P}^{GPS})}{|\mathcal{R}^o(\mathcal{P}^{GPS})|} \quad (\%),$$

as well as the aggregate relative gain

$$\Delta\mathcal{R}(\mathcal{P}^{CP}, \mathcal{P}^{GPS}) = 100 \cdot \frac{\mathcal{R}(\mathcal{P}^{CP}) - \mathcal{R}(\mathcal{P}^{GPS})}{|\mathcal{R}(\mathcal{P}^{GPS})|} \quad (\%),$$

The results for the aggregate relative gain across the 4 scenarios and 4 slices, are displayed in Figure 3.7 (bottom). We can observe that, if all slices have the same load distributions (as in the case of the uniform scenario), the relative gain is decreasing with γ^o, indicating that slices with smaller shares benefit more from statistical multiplexing. For the scenarios where the loads per station are not equal per slice, despite all of them having a similar aggregated gain, the distribution of the load impacts on the distribution of the gain per slice. While in the case of complementary loads the same trend as in the uniform loads scenario hold, in the case of mixed loads, even slices 3 and 4 (which have complementary loads) have smaller γ^o, slices 1 and 2 (which have uniform loads) experience most of the relative gain.

3.8 Conclusions

In this chapter, we have addressed the problem how to optimally partition the set of parallel network resources shared by multiple tenants into VRPs. Our results indicate that in general pooling resources with similar shares and load distributions is beneficial since it does not harm the differentiation ability of slices while

allowing the pool to reap benefits from statistical multiplexing. Moreover, the analytical and numerical evaluations demonstrate that an adequate crafting of the partition provides substantial (up to 40%) performance improvements when compared to static GPS per resource.

3.9 Proofs of chapter results

3.9.1 Proof of Lemma 1

The VRP protection condition from Equation (3.15) can be rewritten as

$$\sum_{b\in P_i} n_b^o \log\left(\frac{n_b^o}{s_b^o}\frac{\frac{s^o(P_i)}{n^o(P_i)}}{\sum\limits_{v\in\mathcal{O}} s^v(P_i)\cdot\frac{n_b^v}{n^v(P_i)}}\right)\geq 0,\quad \forall P_i\in\mathcal{P}.$$

or equivalently

$$\sum_{b\in P_i} n_b^o \log\left(s_b^o\Big(1+\frac{a_b^o(P_i)}{s^o(P_i)\tilde{n}_b^o(P_i)}\Big)\right)\leq 0 \tag{3.27}$$

where $\mathbf{a}^o(P_i)=\{a_b^o(P_i): b\in P_i\}$ and $a_b^o(P_i)\triangleq\sum\limits_{v\in\mathcal{O},v\neq o} s^v(P_i)\tilde{n}_b^v(P_i)$.

Claim 1. *The slice o utility in the pool is minimized when the weights of the other slices $\mathbf{a}^o(P_i)$ are distributed in proportion to the user loads, i.e.,*

$$a_b^{o,*}(P_i)=\tilde{n}_b^o(P_i)\cdot(s(P_i)-s^o(P_i)).$$

Proof. Finding the weights $\mathbf{a}^o(P_i)$ is equivalent to solving the maximization problem

$$\max_{\mathbf{a}}\sum_{b\in P_i} n_b^o\log\left(n_b^o+a_b^o(P_i)\frac{n^o(P_i)}{s^o(P_i)}\right);$$
$$\text{s.t.:}\sum_{b\in P_i} a_b^o(P_i)=\sum_{o'\in\mathcal{O},o'\neq o} s^{o'}(P_i).$$

We can construct the Lagrangian function as:

$$\mathcal{L}(\mathbf{a}^o(P_i), \lambda) = \sum_{b \in P_i} n_b^o \log\left(n_b^o + a_b^o(P_i)\frac{n^o(P_i)}{s^o(P_i)}\right) - \lambda\left(\sum_{b \in P_i} a_b^o(P_i) - \sum_{o' \neq o} s^{o'}(P_i)\right).$$

Using first order optimality conditions,

$$\frac{\partial \mathcal{L}(\mathbf{a}^o(P_i), \lambda)}{\partial \lambda} = \left(\sum_{b \in P_i} a_b^o(P_i) - \sum_{o' \in \mathcal{O}, o' \neq o} s^{o'}(P_i)\right) = 0 \tag{3.28}$$

and

$$\frac{\partial \mathcal{L}(\mathbf{a}^o(P_i), \lambda)}{\partial a_b^o(P_i)} = \frac{n_b^o \frac{n^o(P_i)}{s^o(P_i)}}{(n_b^o + a_b^o(P_i)\frac{n^o(P_i)}{s^o(P_i)})} - \lambda = \frac{1}{\frac{s^o(P_i)}{n^o(P_i)} + a_b^o(P_i)/n_b^o} - \lambda = 0.$$

As a result, for any pair $a_b^o(P_i), a_v^o(P_i)$:

$$\frac{s^o(P_i)}{n^o(P_i)} + a_b^o(P_i)/n_b^o = \frac{s^o(P_i)}{n^o(P_i)} + a_v^o(P_i)/n_v^o \implies a_b^o(P_i)/n_b^o = a_v^o(P_i)/n_v^o$$

and simplifying for any $a_v^o(P_i)$ in Eq. (3.28):

$$a_v^o(P_i) \sum_{b \in P_i} \frac{n_b^o}{n_v^o} = \frac{a_v^o(P_i)}{n_v^o} n^o(P_i) = \sum_{o' \in \mathcal{O}, o' \neq o} s^{o'}(P_i) \tag{3.29}$$

and the claim holds. □

Substituting in Eq. (3.27) $a_b^o(P_i)$ by $\tilde{n}_b^o(P_i)\,(s(P_i) - s^o(P_i))$ and dividing the left and right hand sides by $n_{P_i}^o$ the condition becomes

$$\begin{aligned}
0 &\geq \sum_{b \in P_i} \tilde{n}_b^o \log\left(\frac{s_b^o}{s^o(P_i)} s(P_i)\right), \\
&= \sum_{b \in P_i} \tilde{n}_b^o \log\left(\frac{s_b^o}{s^o(P_i)}\right) + \sum_{b \in P_i} \tilde{n}_b^o \log\left(s(P_i)\right), \\
&= \sum_{b \in P_i} \tilde{n}_b^o \log\left(\frac{s_b^o}{s^o(P_i)}\right) + \log\left(s(P_i)\right).
\end{aligned}$$

If we assume that the slice load is proportional to the share, i.e., $s_b^o/s^o(P_i) = n_b^o/n^o(P_i)$ the condition can be rewritten as:

$$0 \geq \sum_{b \in P_i} \frac{s_b^o}{s^o(P_i)} \log\left(\frac{s_b^o}{s^o(P_i)}\right) + \log\left(s(P_i)\right),$$

so finally, $\mathrm{H}\left(\tilde{\mathbf{s}}^o(P_i)\right) \geq \log(s(P_i))$. □

3.9.2 Proof of Theorem 6

The proof consists in a restriction of "exact cover by 3 sets" (X3C) problem.

Definition 9. ***X3C [39]***

INSTANCE: A finite set X with $|X| = 3q$ and a collection C of 3-element subsets of X.

QUESTION: Does C contains an exact cover for X, that is, a subcollection $C' \subset C$ such that every element of X occurs in exactly one member of C'?

We show that OVP is a general version of X3C by showing a method to build an instance of OVP from any instance of X3C for which finding an OVP solution would solve X3C. We consider a network of $|X| + |C|$ resources, for each subset $C_j = \{i, k, l\} \in C$ we create a set of resources $q_i \in \mathcal{Q}$ and for each element $X_i \in X$ a set of resources $b_i \in \mathcal{B}$. The underlying network graph $\mathcal{G}$ is constructed as follows, if $i \in C_j$ we add a link between q_j and b_i and no link is created otherwise. An instance of the topology mapping can can be seen in Figure 3.8.

We assume that architectural constraints limit the number of resources per partition to 4, i.e., $\bar{K} = 4$ and we will define 3 slices with the following demands

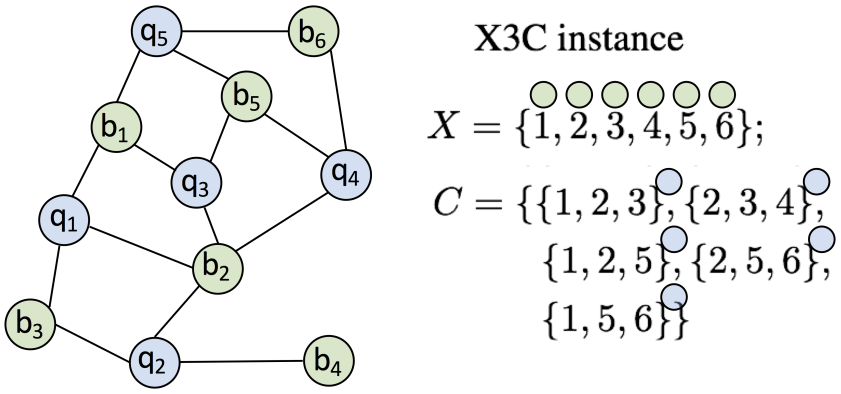

Figure 3.8: Instance of OVP/X3C topology mapping.

and station capacities

$$s_b^o = \begin{cases} 0.09, & o = 1, b \in \mathcal{B} \\ 1/2, & o = 1, b \in \mathcal{Q} \\ 1/4, & \text{otherwise} \end{cases} \quad \text{and} \quad c_b = \begin{cases} 100, & b \in \mathcal{B} \\ 200, & b \in \mathcal{Q} \end{cases}.$$

Moreover, the number of active users N_b^o in the equivalent system is deterministic with value $100\, s_b^o - 1$ and only slice 1 desires protection. Given the chosen demands and $\bar{K} = 4$, it is easy to show that the protection constraints impose that the only feasible subsets used to create partitions are composed by 1 element or by 3 resources from $\mathcal{B}$ and 1 resource from $\mathcal{Q}$, in particular

$$\{b_i, b_k, b_l, q_j \text{ such that } i, k, l \in C_j\}.$$

This is true since the only possible combinations of connected components of four or less resources are displayed in Table 3.1.

Additionally, it is also direct to observe that the global utility will be maximized when the number of subsets of the optimal partition is minimized, since

$$
\begin{cases}
\text{4 res. :} \begin{cases} \text{3 from } \mathcal{B}, \text{1 from } \mathcal{Q}: & \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{1}}(P_i)\right) - \log(s(P_i)) \approx 0.0142 \geq 0 & \text{(protected)} \\ \text{2 from } \mathcal{B}, \text{2 from } \mathcal{Q}: & \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{1}}(P_i)\right) - \log(s(P_i)) \approx -0.0366 \leq 0 & \text{(unprotected)} \end{cases} \\
\text{3 res. :} \begin{cases} \text{2 from } \mathcal{B}, \text{1 from } \mathcal{Q}: & \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{1}}(P_i)\right) - \log(s(P_i)) \approx -0.0179 \leq 0 & \text{(unprotected)} \\ \text{2 from } \mathcal{Q}, \text{1 from } \mathcal{B}: & \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{1}}(P_i)\right) - \log(s(P_i)) \approx -0.0307 \leq 0 & \text{(unprotected)} \end{cases} \\
\text{2 res. :} \begin{cases} \text{1 from } \mathcal{B}, \text{1 from } \mathcal{Q}: & \mathrm{H}\left(\tilde{\mathbf{s}}^{\mathbf{1}}(P_i)\right) - \log(s(P_i)) \approx -0.0366 \leq 0 & \text{(unprotected)} \end{cases} \\
\text{1 resource :} \begin{cases} \text{always protected} \end{cases}
\end{cases}
$$

Table 3.1: Table with the protection for all possible combinations of protection constraints for the problem.

for every possible subset of 4 elements, the total utility from the partition subset is ≈ 0.6326 while the utility if this subset is break into 4 subsets of 1 resource each is ≈ 0.6320. Clearly, any optimal solution for OVP would reveal whether or not there is a solution for the X3C problem, by observing the final partition. If the final partition does not contain any resource b_i in a separate subset, that implies a "YES" answer to X3C, while if any b_i has been left alone, that implies a "NO" answer to the X3C problem. Given that X3C is NP-Complete, it can not exist a polynomial time algorithm to solve OVP, and the proof is complete. □

3.9.3 Proof of Proposition 1

As defined in Eq. (3.13), our expected network utility is given by:

$$
\begin{aligned}
\mathcal{U}(\mathcal{P}) &= \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \sum_{b \in P_i} \frac{\hat{s}^o_{P_i}}{\rho^o(P_i)} \mathbb{E}\left[N^o_b \log(r^o_b(\mathbf{N}, P_i))\right], \\
&= \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \sum_{b \in P_i} \frac{\hat{s}^o_{P_i}}{\rho^o(P_i)} P(N^o_b \geq 1)\, \mathbb{E}\left[N^o_b \log(r^o_b(\mathbf{N}, P_i)) \middle| N^o_b \geq 1 \right],
\end{aligned}
$$

where the second equality follows from the convention that when $N_b^o = 0$ and thus $r_b^o(\mathbf{N}, P_i) = 0$, the utility is 0, i.e., $0 \cdot \log(0) = 0$.

We prove the proposition by taking a Taylor expansion of the utility functions and compute the conditional expected utility of slice o at resource b, i.e., $\mathbb{E}\left[N_b^o \log(r_b^o(\mathbf{N}, P_i)) \middle| N_b^o \geq 1\right]$. We define the utility of slice o at resource b as

$$U_b^o(\mathbf{n}, P_{i(b)}) \triangleq n_b^o \log(r_b^o(\mathbf{n}, P_i)) = n_b^o \log\left(c_b \frac{s^o(P_i)/n^o(P_i)}{\sum\limits_{o' \in \mathcal{O}} s^{o'}(P_i) \frac{n_b^{o'}}{n^{o'}(P_i)} \mathbf{1}(n_b^{o'} > 0)}\right),$$

$$= -n_b^o \log\left(n_b^o + \frac{n^o(P_{i(b)})}{s^o(P_{i(b)})} \sum_{o' \in \mathcal{O}, o' \neq o} \frac{s^{o'}(P_{i(b)})\, n_b^{o'}}{n^{o'}(P_{i(b)})} \mathbf{1}(n_b^{o'} > 0)\right) + n_b^o \log(c_b),$$

where $P_{i(b)}$ is the subset in the partition $\mathcal{P}$ that contains b. We define the continuous extension of $U_b^o(\cdot, P_{i(b)}) : \mathbb{N}^{|\mathcal{O}| \times |\mathcal{B}|} \to \mathbb{R}$ to $\mathbb{R}_+^{|\mathcal{O}| \times |\mathcal{B}|}$ as follows

$$u_b^o(\mathbf{n}, P_{i(b)}) \triangleq -n_b^o \log\left(n_b^o + \frac{n^o(P_{i(b)})}{s^o(P_{i(b)})} \sum_{\substack{o' \in \mathcal{O} \\ o' \neq o}} \frac{s^{o'}(P_{i(b)})\, n_b^{o'}}{n^{o'}(P_{i(b)})} \mathbf{1}(n_b^{o'} > 0)\right) + n_b^o \log(c_b).$$

The continuous extension $u_b^o(\mathbf{n}, P_{i(b)})$ is defined for all $\mathbb{R}_+^{|\mathcal{O}| \times |\mathcal{B}|}$ except where $n^o(P_i) = 0$. Unfortunately, this extension is discontinuous due to the indicator functions $\mathbf{1}(n_b^{o'} > 0)$. To have a continuous and differentiable approximation, we define a ε-perturbed version of $u_b^o(\mathbf{n}, P_{i(b)})$ as follows

$$u_b^{o,(\varepsilon)}(\mathbf{n}, P_{i(b)}) \triangleq -n_b^o \log\left(n_b^o + \frac{n^o(P_{i(b)})}{s^o(P_{i(b)})} \sum_{o' \in \mathcal{O}, o' \neq o} \frac{s^{o'}(P_{i(b)})\, n_b^{o'}}{n^{o'}(P_{i(b)}) + \varepsilon}\right) + n_b^o \log(c_b),$$

which is a continuous and infinitely differentiable function except when $n^o(P_i) = 0$. However, since we will be conditioning on $N_b^o > 0$, which implies $N^o(P_i) > 0$ this

discontinuous point does not affect our result. We define the ε-perturbed version of our overall network utility as follows

$$u^{(\varepsilon)}(\mathcal{P}) \triangleq \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \sum_{b \in P_i} \frac{\hat{s}^o_{P_i}}{\rho^o(P_i)} P(N^o_b \geq 1)\, \mathbb{E}\left[u^{o,(\varepsilon)}_b(\mathbf{N}, P_i) \middle| N^o_b \geq 1 \right].$$

To obtain an expression for the overall expected utility $u^{(\varepsilon)}(\mathcal{P})$ for the ϵ-perturbed rate allocation network, we shall approximate $\mathbb{E}\left[u^{o,(\varepsilon)}_b(\mathbf{N}, P_{i(b)}) \middle| N^o_b \geq 1 \right]$. Recall that, according to Assumption 1, the random vector $\mathbf{N} = (N^o_b : o \in \mathcal{O}, b \in \mathcal{B})$ representing the active number of users are mutually independent Poisson random variables, which conditioned on $N^o_b \geq 1$ have means [99] $\boldsymbol{\varrho}^o_b = (\varrho^{o,o'}_{b,b'} : o \in \mathcal{O}, b \in \mathcal{B})$,

$$\varrho^{o,o'}_{b,b'} = \mathbb{E}[N^{o'}_{b'} | N^o_b \geq 1] = \begin{cases} \rho^o_b (1 - e^{-\rho^o_b})^{-1}, & o' = o, b' = b; \\ \rho^{o'}_{b'}, & \text{otherwise} \end{cases},$$

According to Taylor's Theorem [46], for the function $u^{o,(\varepsilon)}_b(\mathbf{n}, P_{i(b)})$ which is thrice differentiable we have that for any point $\mathbf{n} = (n^o_b : o \in \mathcal{O}, b \in \mathcal{B})$

$$\begin{aligned} u^{o,(\varepsilon)}_b(\mathbf{n}, P_{i(b)}) = u^{o,(\varepsilon)}_b(\boldsymbol{\varrho}^o_b, P_{i(b)}) + \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_{i(b)}}} (n^{o'}_{b'} - \varrho^{o,o'}_{b,b'}) \frac{\partial (u^{o,(\varepsilon)}_b(\boldsymbol{x}, P_{i(b)}))}{\partial x^{o'}_{b'}} \Bigg|_{\boldsymbol{\varrho}^o_b} \\ + \frac{1}{2} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_{i(b)}}} \sum_{\substack{o'' \in \mathcal{O} \\ b'' \in P_{i(b)}}} (n^{o'}_{b'} - \varrho^{o,o'}_{b,b'})(n^{o''}_{b''} - \varrho^{o,o'}_{b,b'}) \frac{\partial^2 (u^{o,(\varepsilon)}_b(\boldsymbol{x}, P_{i(b)}))}{\partial x^{o'}_{b'}\, \partial x^{o''}_{b''}} \Bigg|_{\boldsymbol{\varrho}^o_b} \\ + \frac{1}{6} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_{i(b)}}} \sum_{\substack{o'' \in \mathcal{O} \\ b'' \in P_{i(b)}}} \sum_{\substack{o''' \in \mathcal{O} \\ b''' \in P_{i(b)}}} (n^{o'}_{b'} - \varrho^{o,o'}_{b,b'})(n^{o''}_{b''} - \varrho^{o,o'}_{b,b'})(n^{o'''}_{b'''} - \varrho^{o,o'''}_{b,b'''}) \frac{\partial^3 (u^{o,(\varepsilon)}_b(\mathbf{x}, P_{i(b)}))}{\partial x^{o'}_{b'}\, \partial x^{o''}_{b''}\, \partial x^{o'''}_{b'''}} \Bigg|_{\boldsymbol{\xi}(\mathbf{n})} \end{aligned}$$

where $\boldsymbol{\xi}(\mathbf{n})$ is a point in the segment connecting $\boldsymbol{\varrho}^o_b$ and $\mathbf{n}$.

After taking expectation in both sides of the Taylor expansion of the function $u_b^{o,(\varepsilon)}(\mathbf{n}, P_{i(b)})$ and using the mutually independence of $\mathbf{N}$ conditioned on $N_b^o \geq 1$, we have that

$$\mathbb{E}\left[u_b^{o,(\varepsilon)}(\mathbf{N}, P_{i(b)})\middle| N_b^o \geq 1\right] = u_b^{o,(\varepsilon)}(\boldsymbol{\varrho}_b^o, P_{i(b)}) + \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_{i(b)}}} \frac{\mathrm{Var}\left(N_{b'}^{o'} | N_b^o \geq 1\right)}{2} \frac{\partial^2 (u_b^{o,(\varepsilon)}(\mathbf{x}, P_{i(b)}))}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} + R_b^{o,(\varepsilon)},$$

where

$$R_b^{o,(\varepsilon)} = \frac{1}{6} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_{i(b)}}} \sum_{\substack{o'' \in \mathcal{O} \\ b'' \in P_{i(b)}}} \sum_{\substack{o''' \in \mathcal{O} \\ b''' \in P_{i(b)}}} \mathbb{E}\left[(N_{b'}^{o'} - \varrho_{b,b'}^{o,o'})(N_{b''}^{o''} - \varrho_{b,b'}^{o,o'})(N_{b'''}^{o'''} - \varrho_{b,b'''}^{o,o'''}) \cdot \frac{\partial^3 (u_b^{o,(\varepsilon)}(\mathbf{x}, P_{i(b)}))}{\partial x_{b'}^{o'} \, \partial x_{b''}^{o''} \, \partial x_{b'''}^{o'''}}\bigg|_{\boldsymbol{\xi}(\mathbf{N})} \middle| N_b^o \geq 1\right].$$

Given the Poisson assumption on $\mathbf{N}$ the conditional variances can be computed based on the original Poisson mean loads $\boldsymbol{\rho}$ and are given by [99],

$$\sigma_{b,b'}^{o,o'} = \mathrm{Var}\left(N_{b'}^{o'} | N_b^o \geq 1\right) = \begin{cases} \frac{\rho_b^o}{1 - e^{-\rho_b^o}} \left(1 - \frac{\rho_b^o}{e^{\rho_b^o}}\right), & o' = o, b' = b; \\ \rho_{b'}^{o'}, & \text{otherwise} \end{cases}.$$

Using the Taylor expansions of $\mathbb{E}\left[u_b^{o,(\varepsilon)}(\mathbf{N}, P_{i(b)})\middle| N_b^o \geq 1\right]$ for all $o \in \mathcal{O}, b \in \mathcal{B}$, the final expression for the expected ε-perturbed utility is given by

$$\begin{aligned} u^{(\varepsilon)}(\mathcal{P}) = & \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \sum_{b \in P_i} (1 - e^{-\rho_b^o}) \, u_b^{o,(\varepsilon)}(\boldsymbol{\varrho}_b^o, P_{i(b)}) \\ & + \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \sum_{b \in P_i} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_i}} (1 - e^{-\rho_b^o}) \frac{\sigma_{b,b'}^{o,o'}}{2} \frac{\partial^2 (u_b^{o,(\varepsilon)}(\mathbf{x}, P_{i(b)}))}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} \\ & + \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \sum_{b \in P_i} (1 - e^{-\rho_b^o}) \, R_b^{o,(\varepsilon)} \end{aligned}$$

However, our goal is to compute the expression for the unperturbed rate allocation mechanism. To that end, let us consider any decreasing sequence ε_n of perturbed networks such that $\varepsilon_n \downarrow 0$. For every $\mathbf{n}$, $u_b^{o,(\varepsilon_n)}(\mathbf{n}, P_{i(b)}) \to u_b^o(\mathbf{n}, P_{i(b)})$, and the ϵ_n-perturbed sequence is monotone increasing with n, i.e.,

$$u_b^{o,(\varepsilon_n)}(\mathbf{n}, P_{i(b)}) \leq u_b^{o,(\varepsilon_{n+1})}(\mathbf{n}, P_{i(b)}).$$

Therefore, according to the monotone convergence theorem as $\varepsilon_n \downarrow 0$ the ε-perturbed conditional expected utility converges to the original expected utility

$$\mathbb{E}\left[u_b^{o,(\varepsilon_n)}(\mathbf{N}, P_{i(b)})\middle| N_b^o \geq 1\right] \uparrow \mathbb{E}\left[U_b^o(\mathbf{N}, P_{i(b)})\middle| N_b^o \geq 1\right].$$

and consequently it holds true that $\lim_{\varepsilon_n \downarrow 0} u^{(\varepsilon_n)}(\mathcal{P}) = \mathcal{U}(\mathcal{P})$.

Moreover given that $\boldsymbol{\varrho}_b^o$ is a strictly positive vector, according to Assumption 1 we have that

$$\lim_{\varepsilon_n \downarrow 0} u_b^{o,(\varepsilon_n)}(\boldsymbol{\varrho}_b^o, P_{i(b)}) = u_b^o(\boldsymbol{\varrho}_b^o, P_{i(b)})$$

and computing the derivatives one can show that for all $o' \in \mathcal{O}$ and $b' \in P_{i(b)}$

$$\lim_{\varepsilon_n \to 0} \frac{\partial^2(u_b^{o,(\varepsilon_n)}(\mathbf{x}, P_{i(b)}))}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} = \frac{\partial^2(u_b^o(\mathbf{x}, P_{i(b)}))}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o}$$

Therefore,

$$\begin{aligned}\mathcal{U}(\mathcal{P}) = \lim_{\varepsilon_n \downarrow 0} u^{(\varepsilon_n)}(\mathcal{P}) = &\sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} (1 - e^{-\rho_b^o}) \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} u_b^o(\boldsymbol{\varrho}_b^o, P_i) \\ &+ \sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_i}} (1 - e^{-\rho_b^o}) \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \frac{\sigma_{b,b'}^{o,o'}}{2} \frac{\partial^2 \left(u_b^o(\mathbf{x}, P_i)\right)}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} \\ &+ \lim_{\varepsilon_n \downarrow 0} \sum_{P_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \sum_{b \in P_i} (1 - e^{-\rho_b^o}) R_b^{o,(\varepsilon_n)}\end{aligned}$$

and redefining $\hat{u}_b^o(\mathbf{x}, P_i) = \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \left(1 - e^{-\rho_b^o}\right) u_b^o(\mathbf{x}, P_i)$

$$\mathcal{U}(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} \hat{u}_b^o(\boldsymbol{\varrho}_b^o, P_i) + \sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} \sum_{\substack{o' \in \mathcal{O} \\ b' \in P_i}} \frac{\sigma_{b,b'}^{o,o'}}{2} \frac{\partial^2 \left(\hat{u}_b^o(\mathbf{x}, P_i)\right)}{(\partial x_{b'}^{o'})^2}\bigg|_{\boldsymbol{\varrho}_b^o} + R, \quad (3.30)$$

where

$$\begin{aligned} R &= \lim_{\varepsilon_n \to 0} \sum_{P_i \in \mathcal{P}} \sum_{\substack{o \in \mathcal{O} \\ b \in P_i}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \left(1 - e^{-\rho_b^o}\right) R_b^{o,(\varepsilon_n)} \\ &= \frac{1}{6} \sum_{P_i \in \mathcal{P}} \sum_{\substack{o,o',o'',o''' \in \mathcal{O} \\ b,b',b'',b''' \in P_i}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \left(1 - e^{-\rho_b^o}\right) \mathbb{E}\Bigg[(N_{b'}^{o'} - \varrho_{b,b'}^{o,o'})(N_{b''}^{o''} - \varrho_{b,b'}^{o,o'})(N_{b'''}^{o'''} - \varrho_{b,b'''}^{o,o'''}) \cdot \\ &\qquad \cdot \lim_{\varepsilon_n \to 0} \frac{\partial^3 (u_b^{o,(\varepsilon)}(\mathbf{x}, P_{i(b)}))}{\partial x_{b'}^{o'} \, \partial x_{b''}^{o''} \, \partial x_{b'''}^{o'''}}\bigg|_{\boldsymbol{\xi}(\mathbf{N})} \Bigg| N_b^o \geq 1 \Bigg]. \end{aligned} \quad (3.31)$$

Given that $\mathcal{U}(\mathcal{P})$ is bounded, then given Eq. (3.30) R is well-defined.

3.9.4 Proof of Theorem 7

The proof of this theorem goes along the lines of the proof of Proposition 1. The sequence of linearly scaled networks described in Assumption 2 we define a new stochastic load process $\mathbf{X}^{(\beta)}$ where, $\mathbf{X}^{(\beta)} = \frac{\mathbf{N}^{(\beta)}}{\beta}$ and recall that $\mathbf{c}^{(\beta)} = \beta \mathbf{c}$. We shall again extend the notation defined for the discrete vector $\mathbf{N}$ to the continuous vector $\mathbf{X}^{(\beta)}$.

Using these we can rewrite the expected network utility as follows.

$$\begin{aligned}
\mathcal{U}^{(\beta)}(\mathcal{P}) &= \sum_{\substack{P_i\in\mathcal{P}\\ o\in\mathcal{O}}} \hat{s}^o_{P_i}\mathbb{E}\left[\sum_{b\in P_i}\frac{N_b^{o,(\beta)}}{\beta\,\rho^o(P_i)}\log\left(c_b^{(\beta)}\cdot f_b^o(\mathbf{N}^{(\beta)},P_i)\right)\right]\\
&= \sum_{\substack{o\in\mathcal{O}\\ P_i\in\mathcal{P}\\ b\in P_i}} s^o(P_i)\mathbb{E}\left[\frac{N_b^{o,(\beta)}}{\beta\rho^o(P_i)}\,\log\left(\frac{\hat{s}^o(P_i)\;c_b^{(\beta)}}{N^{o,(\beta)}(P_i)\;\hat{g}_b(\mathbf{N}^{(\beta)},P_i)}\right)\right]\\
&= \sum_{\substack{o\in\mathcal{O}\\ P_i\in\mathcal{P}\\ b\in P_i}} \frac{s^o(P_i)}{\rho^o(P_i)}\mathbb{E}\left[\frac{N_b^{o,(\beta)}}{\beta}\,\log\left(\frac{\hat{s}^o(P_i)\;c_b}{\frac{N^{o,(\beta)}(P_i)}{\beta}\;\hat{g}_b(\frac{\mathbf{N}^{(\beta)}}{\beta},P_i)}\right)\right]\\
&= \sum_{\substack{o\in\mathcal{O}\\ P_i\in\mathcal{P}\\ b\in P_i}} \frac{s^o(P_i)}{\rho^o(P_i)}\mathbb{E}\left[X_b^{o,(\beta)}\,\log\left(\frac{\hat{s}^o(P_i)\;c_b}{X^{o,(\beta)}(P_i)\;\hat{g}_b(\mathbf{X}^{(\beta)},P_i)}\right)\right]\\
&= \mathbb{E}\left[u(\mathcal{P},\mathbf{X}^{(\beta)})\right] \qquad (3.32)
\end{aligned}$$

where $X_b^{o,(\beta)}$ for all $o\in\mathcal{O}, b\in\mathcal{B}$ has mean

$$\mathbb{E}[X_b^{o,(\beta)}] = \mathbb{E}\left[\frac{N_b^{o,(\beta)}}{\beta}\right] = \rho_b^o,$$

and variance

$$\mathrm{Var}(X_b^{o,(\beta)}) = \mathrm{Var}\left(\frac{N_b^{o,(\beta)}}{\beta}\right) = \frac{\mathrm{Var}\left(N_b^{o,(\beta)}\right)}{\beta^2} = \frac{\rho_b^o}{\beta}.$$

We can, therefore consider that our linear scaling is equivalent to using the stochastic fluid model in [62], which can be thought as $\beta\to\infty$ as a law of large number approximation for the stochastic network model. Using again a Taylor expansion

$$\mathcal{U}^{(\beta)}(\mathcal{P}) = u(\mathcal{P},\rho) + \sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\mathrm{Var}(X_b^{o,(\beta)})\cdot\left.\frac{\partial^2(u(\mathcal{P},\mathbf{x}))}{(\partial x_b^o)^2}\right|_{\boldsymbol{\rho}} + o\left(\frac{1}{\beta}\right)$$

The $o(1/\beta)$ is given by the fact that the third central moment of $X_b^{o,(\beta)}$ for all b and o is equal to

$$\begin{aligned}
\mathbb{E}\left[\left(X_b^{o,(\beta)} - \mathbb{E}\left[X_b^{o,(\beta)}\right]\right)^3\right] &= \mathbb{E}\left[\left(X_b^{o,(\beta)}\right)^3\right] - 3\,\mathbb{E}\left[X_b^{o,(\beta)}\right]\,\mathbb{E}\left[\left(X_b^{o,(\beta)}\right)^2\right] + 2\left(\mathbb{E}\left[X_b^{o,(\beta)}\right]\right)^3 \\
&= \mathbb{E}\left[\left(X_b^{o,(\beta)}\right)^3\right] - 3\left(\frac{(\rho_b^o)^2}{\beta} + (\rho_b^o)^3\right) + 2\,(\rho_b^o)^3 \\
&= \frac{1}{\beta^3}\mathbb{E}\left[\left(N_b^{o,(\beta)}\right)^3\right] - 3\frac{(\rho_b^o)^2}{\beta} - (\rho_b^o)^3 \\
&= \frac{1}{\beta^3}\left(\beta^3\,(\rho_b^o)^3 + 3\beta^2\,(\rho_b^o)^2 + \beta\,(\rho_b^o)\right) - 3\frac{(\rho_b^o)^2}{\beta} - (\rho_b^o)^3 = \frac{1}{\beta^2}\rho_b^o
\end{aligned}$$

and therefore the error is in $o(1/\beta)$ since

$$\lim_{\beta\to\infty} \frac{m_b^o\,\mathbb{E}\left[\left(X_b^{o,(\beta)} - \mathbb{E}\left[X_b^{o,(\beta)}\right]\right)^3\right]}{1/\beta} = \lim_{\beta\to\infty} \frac{\frac{1}{\beta^2}\rho_b^o}{1/\beta} = 0$$

which holds since the third derivative $m_b^o = \frac{\partial^3(u(\mathcal{P},\mathbf{X}^{(\beta)}))}{(\partial x_b^o)^3}\Big|_{\boldsymbol{\rho}}$ is finite and independent of β. The same argument can be used in order to prove the same argument for the rest of higher order derivatives.

Moreover, given our pool resource allocation mechanism and normalizing over the total mean loads and capacities

$$\begin{aligned}
u(\mathcal{P},\rho) &= \sum_{\substack{P_i\in\mathcal{P}\\ o\in\mathcal{O}}}\sum_{b\in P_i} \hat{s}^o(P_i)\,\tilde{\rho}_b^o(P_i)\log\left(\frac{c_b\frac{s^o(P_i)}{\rho^o(P_i)}}{\sum\limits_{o'\in\mathcal{O}} s^{o'}(P_i)\tilde{\rho}_b^{o'}(P_i)}\right) \\
&= \sum_{\substack{P_i\in\mathcal{P}\\ o\in\mathcal{O}}} \hat{s}^o(P_i)\,\log\left(\frac{\hat{s}^o(P_i)}{\rho^o(P_i)}\right) - \sum_{\substack{P_i\in\mathcal{P}\\ b\in P_i}} \hat{g}_b(\boldsymbol{\rho},P_i)\,\log\left(\frac{\hat{g}_b(\boldsymbol{\rho},P_i)}{c_b}\right) \\
&= \sum_{\substack{P_i\in\mathcal{P}\\ o\in\mathcal{O}}} \hat{s}^o(P_i)\,\log\left(\frac{\hat{s}^o(P_i)}{\rho\cdot\hat{\rho}^o(P_i)}\right) - \sum_{\substack{P_i\in\mathcal{P}\\ b\in P_i}} \hat{g}_b(\boldsymbol{\rho},P_i)\,\log\left(\frac{\hat{g}_b(\boldsymbol{\rho},P_i)}{c\cdot\hat{c}_b}\right) \\
&= \log\left(\frac{c}{\rho}\right) + \mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{s}}(\mathcal{P})||\hat{\boldsymbol{\rho}}(\mathcal{P})\right) - \mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{g}}(\boldsymbol{\rho},P_i)||\hat{\mathbf{c}}\right).
\end{aligned}$$

Regarding the second order term,

$$\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\frac{\rho_b^o}{\beta}\cdot\frac{\partial^2(u(\mathcal{P},\mathbf{x}))}{(\partial x_b^o)^2}\bigg|_{\boldsymbol{\rho}} = \frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\rho_b^o\cdot\sum_{\substack{P_i\in\mathcal{P}\\ v\in\mathcal{O}\\ d\in P_i}}\frac{\hat{s}^v(P_i)}{\rho^v(P_i)}\frac{\partial^2 x_d^v\,\log\left(\frac{\hat{s}^v(P_i)\,c_d}{x^v(P_i)}\right)}{\partial(x_b^o)^2}\bigg|_{\boldsymbol{\rho}}$$

$$-\frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\rho_b^o\cdot\sum_{\substack{P_i\in\mathcal{P}\\ v\in\mathcal{O}\\ d\in P_i}}\frac{\hat{s}^v(P_i)}{\rho^v(P_i)}\frac{\partial^2 x_d^v\,\log\left(\hat{g}_b(\mathbf{x},P_i)\right)}{\partial(x_b^o)^2}\bigg|_{\boldsymbol{\rho}}$$

$$= -\frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\rho_b^o\cdot\sum_{\substack{P_i\in\mathcal{P}\\ d\in P_i}}\frac{\hat{s}^o(P_i)}{\rho^o(P_i)}\frac{\partial^2 x_d^o\,\log\left(x^o(P_i)\right)}{\partial(x_b^o)^2}\bigg|_{\boldsymbol{\rho}}$$

$$-\frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\rho_b^o\cdot\sum_{b'\in\mathcal{B}}\frac{\partial^2\bar{g}_{b'}(\mathbf{x},P_i)\,\log\left(\hat{g}_{b'}(\mathbf{x},P_i)\right)}{\partial(x_b^o)^2}\bigg|_{\boldsymbol{\rho}}$$

$$= -\frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ P_i\in\mathcal{P}}}\frac{\hat{s}^o(P_i)}{\rho^o(P_i)} - \frac{1}{\beta}\sum_{\substack{o\in\mathcal{O}\\ b\in\mathcal{B}}}\rho_b^o\cdot\sum_{b'\in\mathcal{B}}\frac{\partial^2\bar{g}_{b'}(\mathbf{x},P_i)\,\log\left(\hat{g}_{b'}(\mathbf{x},P_i)\right)}{\partial(x_b^o)^2}\bigg|_{\boldsymbol{\rho}}$$

where $\hat{g}_b(\mathbf{x},P_i)=\sum_{o\in\mathcal{O}}\hat{s}^o(P_i)\tilde{x}_b^o(P_i)$ and $\bar{g}_b(\mathbf{x},P_i)=\sum_{o\in\mathcal{O}}\hat{s}^o(P_i)\frac{x_b^o}{\rho^o(P_i)}$. □

3.9.5 Proof of Fact 2

We will show $\mathrm{D}(\mathcal{P})-\mathrm{D}(\mathcal{P}^{GPS})\leq 0$ as follows

$$\mathrm{D}(\mathcal{P})-\mathrm{D}(\mathcal{P}^{GPS}) = \sum_{P_i\in\mathcal{P}}\sum_{o\in\mathcal{O}}\frac{s^o(P_i)}{s}\log\left(\frac{s^o(P_i)}{s}\frac{\rho}{\rho^o(P_i)}\right) - \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}}\frac{s_b^o}{s}\log\left(\frac{s_b^o}{s}\frac{\rho}{\rho_b^o}\right)$$

$$= \sum_{P_i\in\mathcal{P}}\sum_{o\in\mathcal{O}}\frac{\sum_{b\in\mathcal{P}_i}s_b^o}{s}\log\left(\frac{s^o(P_i)}{s}\frac{\rho}{\rho^o(P_i)}\right) - \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}}\frac{s_b^o}{s}\log\left(\frac{s_b^o}{s}\frac{\rho}{\rho_b^o}\right)$$

$$= \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}}\frac{s_b^o}{s}\log\left(\frac{s^o(P_{i(b)})}{s}\frac{\rho}{\rho^o(P_{i(b)})}\right) - \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}}\frac{s_b^o}{s}\log\left(\frac{s_b^o}{s}\frac{\rho}{\rho_b^o}\right)$$

where $P_{i(b)}$ is the VRP in $\mathcal{P}$ that contains station b. Joining the logarithms

$$\begin{aligned}
\mathrm{D}(\mathcal{P}) - \mathrm{D}(\mathcal{P}^{GPS}) &= \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}} \frac{s_b^o}{s} \log\left(\frac{s^o(P_{i(b)})}{s}\frac{\rho}{\rho^o(P_{i(b)})}\right) - \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}} \frac{s_b^o}{s} \log\left(\frac{s_b^o}{s}\frac{\rho}{\rho_b^o}\right) \\
&= \sum_{b\in\mathcal{B}}\sum_{o\in\mathcal{O}} \hat{s}(P_i^{GPS}) \log\left(\frac{1}{\hat{s}(P_i^{GPS})}\hat{s}^o(P_{i(b)})\frac{\rho_b^o}{\rho^o(P_{i(b)})}\right) \\
&= \sum_{\substack{b\in\mathcal{B}\\ o\in\mathcal{O}}} \hat{s}(P_i^{GPS}) \log\left(\frac{\hat{l}_b^o(\mathcal{P})}{\hat{s}(P_i^{GPS})}\right) = -\mathrm{D}_{\mathrm{KL}}\left(\hat{\mathbf{s}}(\mathcal{P}^{GPS})||\hat{\boldsymbol{l}}(\mathcal{P})\right) \leq 0
\end{aligned}$$

where $\hat{\boldsymbol{l}}(\mathcal{P}) = (l_b^o(\mathcal{P}) : o \in \mathcal{O}, b \in \mathcal{B})$ and $l_b^o = \hat{s}^o(P_i)\tilde{\rho}_b^o(P_i)$ and the last inequality holds given the positivity of the Kullback-Leibler divergence. □

3.9.6 Proof of Fact 3

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

To show that $\langle\hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P})\rangle$ is maximized when $\mathcal{P} = \mathcal{P}^{GPS}$ we will show that

$$\langle\hat{\boldsymbol{s}}(\mathcal{P}^{GPS}), \mathbf{q}(\mathcal{P}^{GPS})\rangle - \langle\hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P})\rangle \geq 0$$

Note that:

$$\begin{aligned}
\langle \hat{\boldsymbol{s}}(\mathcal{P}^{GPS}), \mathbf{q}(\mathcal{P}^{GPS}) \rangle - \langle \hat{\boldsymbol{s}}(\mathcal{P}), \mathbf{q}(\mathcal{P}) \rangle &= \sum_{o \in \mathcal{O}, b \in \mathcal{B}} \frac{s_b^o}{s\, \rho_b^o} - \sum_{o \in \mathcal{O}, P_i \in \mathcal{P}} \frac{\hat{s}_{P_i}^o}{\rho^o(P_i)} \\
&= \sum_{o \in \mathcal{O}, b \in \mathcal{B}} \frac{s_b^o}{s\, \rho_b^o} - \sum_{o \in \mathcal{O}, P_i \in \mathcal{P}} \frac{\sum_{b \in P_i} s_b^o}{s\, \rho^o(P_i)} \\
&= \frac{1}{s} \left(\sum_{o \in \mathcal{O}, b \in \mathcal{B}} \frac{s_b^o}{\rho_b^o} - \sum_{\substack{o \in \mathcal{O} \\ P_i \in \mathcal{P}}} \sum_{b \in P_i} \frac{s_b^o}{\rho^o(P_i)} \right) \\
&= \frac{1}{s} \left(\sum_{o \in \mathcal{O}, b \in \mathcal{B}} \left(\frac{s_b^o}{\rho_b^o} - \frac{s_b^o}{\rho^o(P_i)} \right) \right) \geq 0
\end{aligned}$$

Finally, to see that $\langle \boldsymbol{\rho}, \mathbf{h}(\mathcal{P}) \rangle = 0$ when $\mathcal{P} = \mathcal{P}^{GPS}$, it is enough to observe that $\hat{g}_b(\mathbf{x}, \{b\}) = \frac{1}{s} \sum_{o \in \mathcal{O}} s_b^o$, therefore independent of x, making every h_b^o term equal zero.

□

Part II

Competitive Resource Allocation

Chapter 4

Competitive Slices: Network Slicing Games

This chapter [1] presents a solution for multi-tenant network slicing where slices are competitive and tend to optimize how they distribute their share amongst their users to satisfy their own needs. This mechanism enables tenants to reap the performance benefits of sharing, while retaining the ability to customize their own users allocation. This setting results in a network slicing game in which each tenant/slice exhibit strategic behavior, by adjusting its preferences depending on perceived congestion at resources, so as to maximize its own utility.

We show that, under appropriate conditions, the game associated with such strategic behavior converges to a Nash equilibrium. Further, at the Nash equilibrium, a tenant can always achieve the same, or better, performance than under a static partitioning of resources irrespective on how other slices behave hence providing the same level of protection as such static partitioning. We further analyze the efficiency and fairness of the resulting allocations, providing tight bounds for the price of anarchy and envy-freeness.

[1] Publications based on this chapter: Pablo Caballero, Albert Banchs, Gustavo de Veciana, and Xavier Costa- Perez. Network slicing games: Enabling customization in multi-tenant networks. In IEEE INFOCOM 2017 and under review for ACM/IEEE Transactions on Networking. All co-authors contributed equally.

Our analysis and extensive simulation results confirm that the mechanism provides a comprehensive practical solution to realize network slicing resource allocation. The situation where, in addition to the resource allocation, the user association is decided by the slices is, unfortunately, not ensured to converge to a Nash Equilibrium suggesting that this decision should perhaps remain in the control of the network infrastructure provider. Our theoretical results also fill a gap in the literature regarding the analysis of such resource allocation models for the case of strategic players.

4.1 Related work

The resource allocation mechanism informally described above corresponds to a Fisher market, which is a standard framework in economics. In such markets, buyers (in our case slices) have fixed budgets (in our case network shares) and (according to their preferences) bid for resources within their budget, which are then allocated to buyers proportionally to their bids. Analysis of the Fisher market shows that, as long as buyers are price-taking (i.e., they do not anticipate the impact of their bids on the price – in our case, the impact of the slices' preferences on the overall congestion), the Nash equilibrium is socially optimal, and distributed algorithms can be easily devised to reach it [110]. This assumption may be reasonable for markets where the impact of a single buyer on a resource's price is negligible, but does not apply to our case where a relatively small number of active tenants might be sharing resources.

There is a substantial literature on Fisher markets with strategic buyers,

which, as will be studied in this chapter, anticipate the impact of their bids [34]. The analysis, so far, has been limited to the case of buyers with *linear utility* functions of the allocated resources, which can lead to extremely unfair allocations. While such utility functions may be suitable for goods, they are not an appropriate model for tenants wishing to customize allocations amongst their customers. This chapter includes a comprehensive analysis for a wide set of slice utility functions, including the convergence of best response dynamics and other results which to our knowledge are new.

A related resource allocation model often considered in the networking field is the so-called 'Kelly's mechanism' [61]; this mechanism allocates resources to players proportionally to their bids and, assuming that they are price-taking, converges to a social optimum. Follow-up work has considered price-anticipating players in this setting; for example, [52] analyze efficiency losses, while [107] devise a scalar-parametrized modification that is once again socially optimal for price-anticipating players. However, in Kelly's mechanism players respond to their payoff (given by the utility minus cost) whereas in our model tenants' behavior is only driven by their utilities (since they have a fixed budget: the network share). Consequently, results on the analysis of Kelly's mechanism are not applicable to our setting.

Table 4.1 capture the main resource allocation models for this problem highlighting some of the most relevant work for each problem and situating the contribution of this work.

From a more practical angle, multi-tenant sharing has been studied from

	price taking	**price anticipating**	
	scalar bid	**scalar bid**	**vector bid**
non fixed budget	[61] Kelly's mechanism (conv, efficiency)	VCG-Kelly mechanism [107] Hajek/Yang [53] Johari/Tsitsiklis	Johari/Tsitsiklis [52] Efficiency of congestion games
	concave utilities	**linear utilities**	**concave utilities**
fixed budget	[110] Zhang (convergence)	[34] Zhang (conv, efficiency)	*This work* (conv, efficiency)

Table 4.1: Resource allocation models.

different points of view, including planning, economics, coverage, performance, etc. [35, 73]. This chapter focuses specifically on the design of algorithms for resource sharing among tenants, which has been previously addressed by [43, 76, 77, 92]. The work of [92] considers sharing via a bid-based auction, which may incur substantial overhead and complexity; in contrast, our approach relies on fixed (pre-negotiated) network shares. The works of [43, 76, 77] also fix a network share per slice, but consider approaches where the infrastructure makes centralized decisions on the resources allocated to each tenant's customers; hence, these approaches do not enable tenants to make their own decisions on how to allocate resources to their customers.

Network slicing has emerged as a desirable feature for 5G [86]. 3GPP has started work on defining requirements for network slicing [8], whereas the Next Generation Mobile Network (NGMN) alliance has identified network sharing among slices as a key issue [85]. In spite of these efforts, most of the work so far

has addressed architectural aspects with only a limited focus on resource allocation algorithms [96, 113]. To the best of our knowledge, this is the first work investigating how to enable tenants to customize their allocations in a dynamic slicing model; there is wide consensus that such an ability to customize tenants' allocations is needed to efficiently satisfy their very diverse requirements (see, e.g., [10] for examples of vertical tenants).

4.2 Chapter organization

The rest of the chapter is organized as follows. After introducing our system model (Section 4.3), we show that with the resource sharing model under study, each slice has the ability to achieve the same or better utility than under static resource slicing irrespective of how the other slices behave, which confirms that this model effectively protects slices from one another (Section 4.4.1). Next we show that if tenants exhibit strategic behavior (i.e, optimize their utilities), then (i) a Nash equilibrium exists under mild conditions; and (ii) the system converges to such an equilibrium when tenants sequentially take their best response (Sections 4.4.2 and 4.4.3). The resulting efficiency and fairness among tenants are then studied, providing: (i) a tight bound on the Price of Anarchy of the system, and (ii) a bound on the Envy-freeness (Section 4.5). Our results are validated via simulation, confirming that the approach provides substantial gains, protects network slices from each other, operates close to optimal performance and is effectively envy-free (Section 4.6). The proof of the theoretical results for this chapter are provided in Section 4.8.

4.3 System model

We consider a wireless network consisting of a set of resources $\mathcal{B}$ (the base stations or sectors) shared by a set of network slices $\mathcal{O}$ (the tenants). At a given point in time, the network supports a set of users $\mathcal{U}$ (the customers or devices), which can be subdivided into subsets $\mathcal{U}_b$ (the users at base station b), $\mathcal{U}^o$ (the users of slice o) and $\mathcal{U}_b^o$ (their intersection). We further assume that a user $u \in \mathcal{U}$ has a mean peak capacity c_u depending on the choice of modulation and coding at the base station it is associated with. For any user u, we let $b(u)$ denote the base station it is currently associated with.

4.3.1 Resource allocation model

As indicated in the introduction, we focus on a well-established resource sharing model known in economics as a Fisher market. Hereafter, we will refer to this model as the 'Share-Constrained Proportional Allocation' (SCPA) mechanism.

In our setting, each slice o is allocated a network share s_o (corresponding to its budget) such that $\sum_{o \in \mathcal{O}} s_o = 1$. The slice is at liberty in turn to distribute its share amongst its users, assigning them weights (corresponding to the bids): w_u for $u \in \mathcal{U}^o$, such that $\sum_{u \in \mathcal{U}_o} w_u = s_o$. We let $\mathbf{w}^o = (w_u \colon u \in \mathcal{U}_o)$ be the weights of slice o, $\mathbf{w} = (w_u \colon u \in \mathcal{U})$ those of all slices and $\mathbf{w}^{-o} = (w_u \colon u \in \mathcal{U} \setminus \mathcal{U}^o)$ the weights of all users excluding those of slice o.

We shall assume users are allocated a fraction of resources at their base

station proportionally to their weights w_u. Thus the rate of user u is given by

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

where $l_b(\mathbf{w}) = \sum_{u \in \mathcal{U}_b} w_u$ denotes the overall load at b (recall that c_u is the achievable rate if the user had the entire base station to itself).

To implement the above resource allocation, a slice needs to communicate the weights of its users $\mathbf{w}^o$ to the infrastructure. When selecting its weights, we assume that the slice is aware of the overall load at each base station (indeed, a slice could infer these by varying its users' weights and observing the resulting resource allocations).[2]

In the case where a slice o is the only one with users at a given base station b, we shall assume that the slice's users are allocated the entire capacity at that base station independent of their weights. Thus, such a slice would set $w_u = 0$ for these users, allowing them to receive all the resources of this base station without consuming any share. To avoid dealing with this special case, and without loss of generality, we will make the following assumption for the rest of the chapter.

Assumption 3. *(Competition at all resources) We assume that all resources have active users from at least two slices.*

[2]It is worth noting that, with the SCPA mechanism under study, the weights of a given tenant are not disclosed to the others, which only see the overall load at each base station.

4.3.2 Network slice utility and service differentiation

Network slices may support services and customers of different types and needs. Alternatively, competing slices with similar customer types may wish to differentiate the service they provide. To that end, we assume each network slice has a *private* utility that reflects the benefit obtained by the slice from a given allocation and is given by

$$U^o(\mathbf{w}) = \sum_{u \in \mathcal{U}^o} \phi_u f_u(r_u(\mathbf{w})),$$

where ϕ_u is the relative priority of user u, with $\phi_u \geq 0$ and $\sum_{u \in \mathcal{U}_o} \phi_u = 1$, and $f_u(\cdot)$ is a (concave) utility function associated with the user. In the sequel, we will often focus on the following well-known class of utility functions [80].

Definition 10. *A network slice o has a homogenous α_o-fair utility if for all $u \in \mathcal{U}^o$ we have that*

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

Thus, in our setting, a slice is free to choose different fairness criteria in allocating resources across its users, by selecting the appropriate α_o parameter. Note that $\alpha_o = 1$ corresponds to the widely accepted proportional fairness criterion, while $\alpha_o = 2$ corresponds to potential delay fairness, $\alpha_o \to \infty$ to max-min fairness and $\alpha_o = 0$ to linear sum utility.

A slice can also 'strategically' optimize the weight allocation of its users to maximize its own utility. We will consider such strategic behavior of weight allocations in Section 4.4.

4.3.3 Baseline allocations

Next we introduce two natural resource allocation comparative baselines: socially optimal allocations and static slicing.

Socially Optimal Allocations (SO) If slices were to share their utility functions with a centralized authority, one could in principle consider a socially optimal allocation of weights and resources. These would be given by the maximizer to the *overall network utility* $U(\mathbf{w})$ given by (see [76]):

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

Note that (as in [76]) we have weighted the slices' utilities to reflect their shares (thus prioritizing those with higher shares). We shall denote the resulting optimal weight and resource allocations under the socially optimal allocations by $\mathbf{w}^*$ and $\mathbf{r}^* = (r_u^* : u \in \mathcal{U})$, respectively.

Static Slicing (SS) By static slicing (also known as static splitting [30]) we refer to a complete partitioning of resources based on the network shares $s_o, o \in \mathcal{O}$. In this setting, each slice o receives a fixed fraction s_o of each resource and can

unilaterally optimize its weight allocation as follows:

$$\begin{aligned}\max_{\mathbf{w}^0 \geq 0} \quad & U^o(\mathbf{w}^o) = \sum_{u \in \mathcal{U}^o} \phi_u f_u(r_u(\mathbf{w}^o)) \\ \text{s.t.} \quad & r_u(\mathbf{w}^o) = \frac{w_u}{\sum_{v \in \mathcal{U}^o_{b(u)}} w_v} s_o c_u \quad \forall u \in \mathcal{U}^o \\ & \sum_{u \in \mathcal{U}^o} w_u = s_o,\end{aligned}$$

where we have abused notation to indicate that, in this case, U^o and r_u depend only on $\mathbf{w}^o$. We shall denote the resulting optimal weight and resource allocations under static slicing for all slices by $\mathbf{w}^{ss}$ and $\mathbf{r}^{ss} = (r_u^{ss} : u \in \mathcal{U})$ respectively, where

$$r_u^{ss} = \frac{w_u^{ss}}{\sum_{v \in \mathcal{U}^o_{b(u)}} w_v^{ss}} s_o c_u \quad \forall u \in \mathcal{U}^o, \forall o \in \mathcal{O}. \tag{4.1}$$

4.4 Strategic behavior and Nash Equilibrium

Under the SCPA resource allocation model, it is reasonable to assume that a player (network slice) would choose to adjust its weights so as to optimize its utility (and thus the service delivered to its customers). Since the resources allocated to a user depend on the weight allocations of the other slices, such behavior would be predicated on the aggregate weight of the other slices at each resource. From the point of view of slice o, the overall load at resource b can be decomposed as

$$l_b(\mathbf{w}) = a_b^o(\mathbf{w}^{-o}) + d_b^o(\mathbf{w}^o)$$

where

$$a_b^o(\mathbf{w}^{-o}) = \sum_{o' \in \mathcal{O} \setminus \{o\}} \sum_{u \in \mathcal{U}_b^{o'}} w_u \text{ and } d_b^o(\mathbf{w}^o) = \sum_{u \in \mathcal{U}_b^o} w_u,$$

correspond to the aggregate weight of the other slices and that of slice o, respectively. As indicated in Section 4.3, we assume $\boldsymbol{a}^o(\mathbf{w}^{-o}) = (a_b^o(\mathbf{w}^{-o}) : b \in \mathcal{B})$ are readily available to slice o.

4.4.1 Gain over Static Slicing

We first analyze if strategic behavior on the part of network slices may result in allocations that are worse that those under static slicing. Note that static slicing provides complete isolation among slices but potentially poor utilization. A critical question is whether dynamic sharing, which achieves a higher resource utilization, also provides the same level of protection. This is confirmed by the following result.[3]

Lemma 2. *Consider slice o and any feasible weight allocation $\mathbf{w}^{-o}$ for other slices satisfying the network share constraints. Then, there exists a weight allocation $\mathbf{w}^o$ for slice o, possibly dependent on $\mathbf{w}^{-o}$, such that the resulting weight allocation $\mathbf{w}$ satisfies $r_u(\mathbf{w}) \geq r_u^{ss}$ for all $u \in \mathcal{U}_o$.*

This lemma is easily shown by choosing $\mathbf{w}^o$ such that

$$w_u = \frac{w_u^{ss}}{\sum_{u \in \mathcal{U}_{b(u)}^o} w_u^{ss}} \frac{a_{b(u)}^o(\mathbf{w}^{-o})}{\sum_{b' \in \mathcal{B}_o} a_{b'}^o(\mathbf{w}^{-o})} s_o, \ \forall u \in \mathcal{U}^o$$

where $\mathcal{B}^o$ is the set of base stations where slice o has users. The intuitive interpretation for this choice is that by distributing its weights proportionally to the load at each base station, slice o can achieve the same resource allocation as static slicing

[3]The proofs of all lemmas and theorems are provided in the Appendix.

at each base station. Further, by redistributing these allocations amongst its user in the same manner as static slicing, it achieves at least as much rate per user.

It follows immediately from this lemma that under the SCPA resource allocation model, if all slices exhibit strategic behavior attempting to maximize their utilities, they necessarily achieve a higher utility than under static slicing.

Theorem 8. *If the game where each network slice maximizes its utility has a Nash equilibrium, then each slice achieves a higher utility than under static slicing.*

Note this result does not require slices to have homogenous or concave utilities, just that they be increasing in the users' rate allocations.

4.4.2 Existence and uniqueness of Nash Equilibrium

Next we study whether there exists a Nash equilibrium (NE) under which no slice can benefit by unilaterally changing its weight allocation. To that end, we first characterize the best response of a slice.

Given the weights of the other slices, $\mathbf{w}^{-o}$, the best response of slice o is the unique maximizer $\mathbf{w}^o$ of its utility, i.e.,

$$\begin{aligned} \max_{\mathbf{w}'^o \geq 0} & \sum_{u \in \mathcal{U}_o} \phi_u f_u \left(\frac{w'_u c_u}{a^o_{b(u)}(\mathbf{w}^{-o}) + d^o_{b(u)}(\mathbf{w}'^o)} \right) \\ \text{s.t} & \sum_{u \in \mathcal{U}_o} w'_u = s_o. \end{aligned}$$

The following lemma characterizes the best response for a network slice with homogenous α_o-fair utility (see [34] for the best response when $\alpha_o = 0$).

Lemma 3. *Suppose slice o has a homogeneous α_o-fair utility (with $\alpha_o > 0$). Given the weights of the other slices $\mathbf{w}^{-o} > 0$, slice o's best response $\mathbf{w}^o$ is the unique solution to the following nonlinear set of equations:*

$$w_u = \frac{\beta_u \frac{\left(a^o_{b(u)}(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_{b(u)}(\mathbf{w}^{-o}) + d^o_{b(u)}(\mathbf{w}^o)\right)^{\frac{2}{\alpha_o}-1}}}{\sum_{v \in \mathcal{U}_o} \beta_v \frac{\left(a^o_{b(v)}(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_{b(v)}(\mathbf{w}^{-o}) + d^o_{b(v)}(\mathbf{w}^o)\right)^{\frac{2}{\alpha_o}-1}}} s_o, \quad \forall u \in \mathcal{U}^o, \tag{4.2}$$

where $\beta_u := (\phi_u)^{\frac{1}{\alpha_o}} (c_u)^{\frac{1}{\alpha_o}-1}$.

Note that slice o need only know $\boldsymbol{a}^o(\mathbf{w}^{-o})$ to compute its best response. Building on this characterization, we will study the game in which all slices choose to allocate their weights based on their best response. The following theorem proves that this game admits a Nash equilibrium, i.e., there is a weight allocation $\mathbf{w}$ such that no slice can improve its utility by modifying its weights unilaterally.[4]

Theorem 9. *Suppose all slices have homogenous α_o-fair utilities (with possibly different $\alpha_o > 0$). Then, there exists a (not necessarily unique) Nash equilibrium satisfying* (4.2) *for each slice.*

The above theorem covers any finite α_o value, but leaves out the case $\alpha_o \to \infty$, which yields a utility function $U^o(\mathbf{w}) = \min_{u \in \mathcal{U}^o} (r_u(\mathbf{w}))$ and corresponds to max-min fairness. The following lemma shows that in this case the existence of a NE is not guaranteed.

[4]The existence of a NE had already been proven by [110] for the case $\alpha_o = 0 \ \forall o$. Here we extend this result to any combination of α_o values.

Lemma 4. *Let $U^o(\mathbf{w}) = \min_{u \in \mathcal{U}^o}(r_u(\mathbf{w}))$ for two or more slices. Then, the existence of a NE cannot be guaranteed.*

4.4.3 Convergence of Best Response dynamics

Below we will consider a best response game wherein slices realize their best responses in rounds, either (i) updating their weights ($\mathbf{w}^o$) sequentially, one at a time and in the same fixed order, in response to the other slices' weights ($\boldsymbol{a}^o$); or (ii) having all slides update their weights simultaneously in each round in response to the other slices' weights in the previous round.

Theorem 10. *If slices have homogeneous α_o-fair utilities, possibly with different $\alpha_o \in [1, 2]$ for $o \in \mathcal{O}$, then the best response game converges to a Nash equilibrium. This result holds both for sequential and for simultaneous updates.*

Note that the value of α_o impacts a slice's best response and consequently the game dynamics. As seen in Lemma 3, the best response weights are proportional to:

$$w_u \propto g(a_b^o, d_b^o) := \frac{(a_b^o)^{\frac{1}{\alpha_o}}}{(a_b^o + d_b^o)^{\frac{2}{\alpha_o}-1}},$$

where we have suppressed the dependency of a_b^o on $\mathbf{w}^{-o}$ and d_b^o on $\mathbf{w}^o$. The function $g(\cdot, \cdot)$ has different properties depending on α_o which are shown in Table 4.2. The regime where $1 \leq \alpha_o \leq 2$, considered in Theorem 10, is of particular interest since it includes proportional ($\alpha_o = 1$) and potential delay ($\alpha_o = 2$) fairness. It is known that convergence is not ensured when $\alpha_o = 0$ for all slices (see [34]); for other regimes, we resort to the simulations results of Section 4.6, which suggest

convergence for any $\alpha_o > 0$ since they are different problems in nature and therefore the analysis requires a distinct approach.

	$\alpha_o = 0$	$0 < \alpha_o < 1$	$1 \leq \alpha_o \leq 2$	$2 < \alpha_o < \infty$
g w.r.t. d_b^o	linear	convex	convex	concave
g w.r.t. a_b^o	linear	convex	concave	concave
NE existence	✓ [34]	✓Theorem 9 for heterogeneous α_o		
convergence	× [34]	✓simulations	✓Theorem 10	✓simulations

Table 4.2: Impact of α_o on slice's Best Responses.

Perhaps surprisingly, the above result is quite challenging to show. The key challenge lies in the "price-anticipating" aspect of the best response, in which players anticipate the impact of their own allocation (indeed, as mentioned in the introduction, there are very few results in the literature on the convergence of price-anticipating best response dynamics).

4.5 Performance bounds analysis

In this section, we analyze the performance of the Nash equilibrium in terms of two standard metrics for efficiency and fairness: (i) the *price of anarchy*, which gives the loss in overall utility resulting from slices' strategic behavior, and (ii) *envy-freeness*, which captures the degree to which a slice would prefer another slice's allocations across the network resources. We will focus on the case where slice utilities are 1-fair homogeneous i.e., $U^o(\mathbf{w}) = \sum_{u \in \mathcal{U}^o} \phi_u \log(r_u(\mathbf{w}))\ \forall o \in \mathcal{O}$ – a widely accepted case leading to the well-known proportionally fair allocations.

4.5.1 Efficiency: Price of Anarchy

The following result characterizes the socially optimal allocation of resources defined in Section 4.3.3 (see [74]).

Fact 1. For slices with 1-fair homogenous utilities, the socially optimal allocation of resources $\mathbf{w}^*$ is such that $w_u^* = \phi_u s_o, \ \ \forall u \in \mathcal{U}^o$ and $\forall o \in \mathcal{O}$.

The following theorem bounds the difference between the *overall network utility* resulting from the socially optimal allocation, $U(\mathbf{w}^*)$, and that obtained at a Nash equilibrium of the SCPA resource allocation mechanism, $U(\mathbf{w})$ – the proof is provided in the Appendix.

Theorem 11. *If all slices have* 1*-fair homogenous utilities, then the Price of Anarchy (PoA) associated with a Nash equilibrium* $\mathbf{w}$ *satisfies*

$$PoA := U(\mathbf{w}^*) - U(\mathbf{w}) \leq \log(e).$$

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

Note that, with 1-fair utilities, if we increase the capacity of all resources by a factor Δc, we have a utility increase of $\log(\Delta c)$. Thus, the performance improvement achieved by the socially optimal allocation over SCPA is (in the upper bound) equivalent to having a capacity e times larger, i.e., almost the triple capacity. While there are some (pathological) cases in which such a bound can be achieved, our simulation results show that for practical scenarios the actual performance difference between the two allocations is much smaller, confirming that (for $\alpha_o = 1$) the flexibility gained with the SCPA mechanism comes at a very small price in performance.

4.5.2 Fairness: Envy-freeness

Next we consider a Nash equilibrium $\mathbf{w}$ and analyze whether a slice, say o, with utility $U^o(\mathbf{w})$, might have a better utility if it were to exchange its resources with those of another slice, say o'. To that end, we denote by $\tilde{\mathbf{w}}$ the resulting weight allocation when the users of slices o and o' exchange their allocated resources. It is easy to see that $\tilde{\mathbf{w}}^o$ is such that

$$\tilde{w}_u^o = \frac{\phi_u}{\sum_{v \in \mathcal{U}_b^o} \phi_v} d_b^{o'}(\mathbf{w}) \text{ for all } b \in \mathcal{B} \text{ and all } u \in \mathcal{U}_b^o, \tag{4.3}$$

i.e., slice o takes the aggregate weight of o' at base station b under the Nash equilibrium, $d_b^{o'}(\mathbf{w})$, and allocates it proportionally to its user priorities. Clearly, $\tilde{\mathbf{w}}^{o'}$ is defined similarly and the remaining slices weights remain unchanged under $\tilde{\mathbf{w}}$.

We define the envy of slice o for o''s resources under the Nash equilibrium $\mathbf{w}$ by

$$E^{o,o'} := U^o(\tilde{\mathbf{w}}) - U^o(\mathbf{w}).$$

Note that envy is a "directed" concept, i.e., it is defined from slice o's point of view. When $E^{o,o'} \leq 0$, we say slice o is not envious. The following theorem provides a bound on $E^{o,o'}$.

Theorem 12. *Consider a slice o with 1-fair homogeneous utilities and the remaining slices $\mathcal{O} \setminus \{o\}$ with arbitrary slice utilities. Consider a slice o' such that $s_o = s_{o'}$. Let $\mathbf{w}$ denote a Nash equilibrium and $\tilde{\mathbf{w}}$ denote the resulting weights when o and o' exchange their resources. Then, the envy of slice o for o' satisfies*

$$E^{o,o'} = U^o(\tilde{\mathbf{w}}) - U^o(\mathbf{w}) \leq 0.060.$$

Furthermore, there is a game instance where $0.041 \leq E_{o,o'}$.

Given that, if one increases the rates of all users by a factor Δr this yields a utility increase of $\log(\Delta r)$, one can interpret this result as saying that, by exchanging resources with o', slice o may see a gain equivalent to increasing the rate of all its users by a factor between 4.1% and 6.1% (given by the lower and upper bounds of the above theorem). This is quite low and, moreover, simulation results show that in practical settings there is actually (almost) never any envy, confirming that our system is (practically) envy-free.

4.6 Performance Evaluation

Next, we evaluate the performance of the SCPA resource allocation mechanism via simulation. The mobile network scenario considered is based on the IMT-A evaluation guidelines for dense 'small cell' deployments [1], which consider base stations with an intersite distance of 200 meters in a hexagonal cell layout with 3 sector antennas.[5] The network size $|\mathcal{B}|$ is 57 sectors and, unless otherwise stated, users move according to the Random Waypoint Model (RWP). Users' Signal Interference to Noise Ratio ($\overline{\text{SINR}}_u$) is computed based on physical layer network model specified in [1] (which includes path loss, shadowing, fast fading and antenna gain) and user association follows the strongest signal policy. The achievable rate for users, c_u, are determined based on the thresholds reported in [7]. For all our simu-

[5]Note that, in this setting, users associate with sectors rather than the base stations we used in the mechanism description and analysis.

lation results, we obtained 95% confidence intervals with relative errors below 1% (not shown in the figures).

4.6.1 Overall performance

Throughout the chapter we have used *static slicing* and the *socially optimal* resource allocations as our baselines. To confirm our analytical results and gain additional insights, we have evaluated the performance of the SCPA mechanism versus these two baselines via simulation. As an intuitive metric for comparison, we have used the extra capacity required by these baseline schemes to deliver the same performance as SCPA: (i) *Gain over SS*: additional resources required by *static slicing* to provide the same utility as SCPA (in %); and (ii) *Loss versus SO*: additional resources required by SCPA to provide the same utility as the *socially optimal* allocation (in %). Note that the latter metric is closely related to the *Price of Anarchy* analyzed in Section 4.5.1.

The results shown in Figure 4.1 are for different user densities ($|\mathcal{U}|/|\mathcal{B}|$) and different slice utilities (α_o parameter). As expected, the SCPA mechanism always has a gain over static slicing and a loss over the social optimal. However, for $\alpha_o = 1$ the loss is well below the bound given in Section 4.5.1. We further observe that performance is particularly good as long as α_o does not exceed 1 (Gain over SS up to 50% and Loss over SO below 5%), and it degrades mildly as α_o increases.

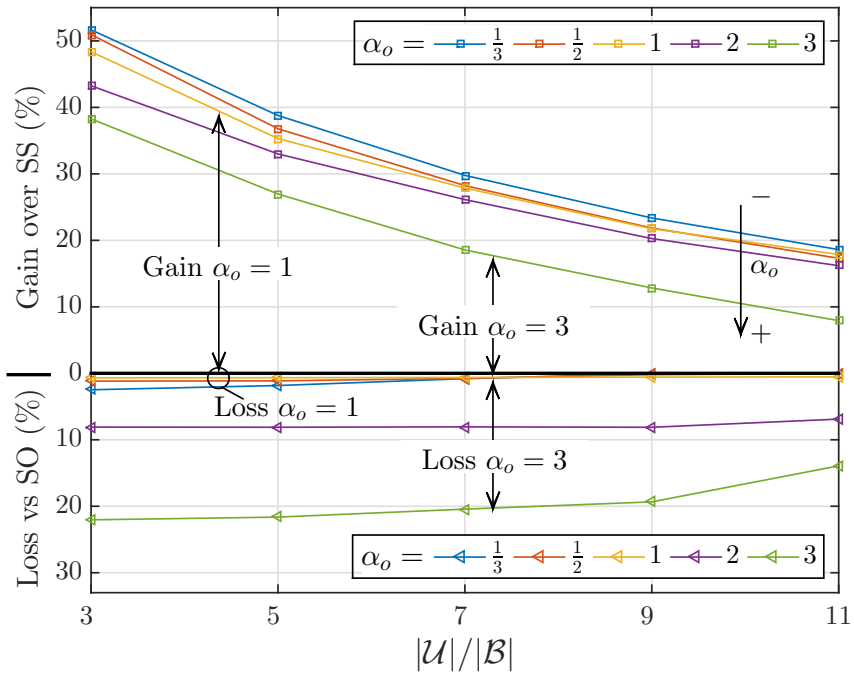

Figure 4.1: Average Gain over Static slicing and Loss against Social optimum for different scenarios.

4.6.2 Fairness

In addition to overall performance, it is of interest to evaluate the fairness of the resulting allocations. While in Section 4.5.2 we derived analytically a bound on the envy, we have further explored this via simulation by evaluating up to 10^7 randomly generated scenarios, with parameters drawn uniformly in the ranges: $|\mathcal{O}| \in [2, 12]$, $|\mathcal{B}| \in [10, 90]$, $|\mathcal{U}|/|\mathcal{B}| \in [3, 15]$, $\alpha_o \in [0.01, 30]$ and $\boldsymbol{\phi}$ vectors in the simplex. Our results show that $E^{o,o'} < 0$ holds for *all* the cases explored, confirming that in practice the system is envy-free.

4.6.3 Protection against other slices

One of the main objectives of our proposed framework is to enable slices to customize their resource allocations. This can be done by adjusting (i) the user priorities ϕ_u, and (ii) the parameter α_o, which regulates the desired level of fairness among the slice's users. In order to evaluate the impact that these settings have amongst slices, we simulated a scenario with three slices: Slice 1 has $\alpha_1 = 1$, Slice 2 has $\alpha_2 = 4$, and Slice 3 has α_3 with varying values. For simplicity, we set the priorities ϕ_u equal for all users.

Figure 4.2 shows the rate distributions of the 3 slices. We observe that the choice of α_3 is effective in adjusting the level of user fairness for Slice 3; indeed, as α_3 grows, the rate distribution becomes more homogeneous. Such customization at Slice 3 has a higher impact on Slice 1 than on Slice 2. This is the case because, as α_2 is quite large, the distribution of Slice 2's rates remains homogeneous, making the slice fairly insensitive to the choices of the other slices. As can be seen in the subplots, the utilities of Slices 1 and 2 are not only larger than the utility of static slicing, but remain fairly insensitive to α_3, showing that in both cases we have a good level of protection between slices.

4.6.4 Convergence speed

The existence of a Nash equilibrium and the convergence of Best Response Dynamics are essential for the system stability. While the existence of a Nash equilibrium has been proven for all α_o values, convergence has only been shown for $\alpha_o \in [1, 2]$. In order to confirm the convergence for other α_o values, we have con-

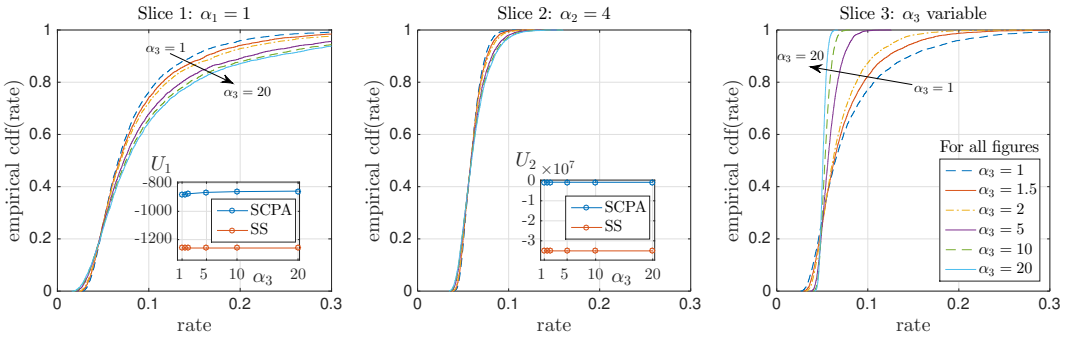

Figure 4.2: Impact of α_3 decision on the slice rate distributions.

ducted extensive simulations implementing sequential best response updates for up to 10^7 randomly generated scenarios within the same parameter space as in Section 4.6.2. Our results confirmed the convergence of the best response game in all cases. Moreover, they also showed that convergence speed mainly depends on α_o, while it is fairly insensitive to the user priorities and the network size. According to the results, convergence is very quick for $\alpha_o \leq 1$ (about 8 rounds) and increases slightly as α_o grows (about 16 rounds for $\alpha_o = 3$). The average number of rounds needed for the Best Response dynamics to converge are shown in Figure 4.3.

4.6.5 Impact of user mobility

The above results assume a Random Waypoint mobility model where users are (on average) uniformly distributed across space. To understand the impact of other user distributions, we evaluated the *Gain over SS* for the following user mobility patterns: (i) *uniform*: all slices with a uniform spatial load distribution; (ii) *overlapping hotspots*: all slices with the same non-uniform spatial load distribution; (iii) *non-overlapping hotspots*: different slices with different non-uniform spatial

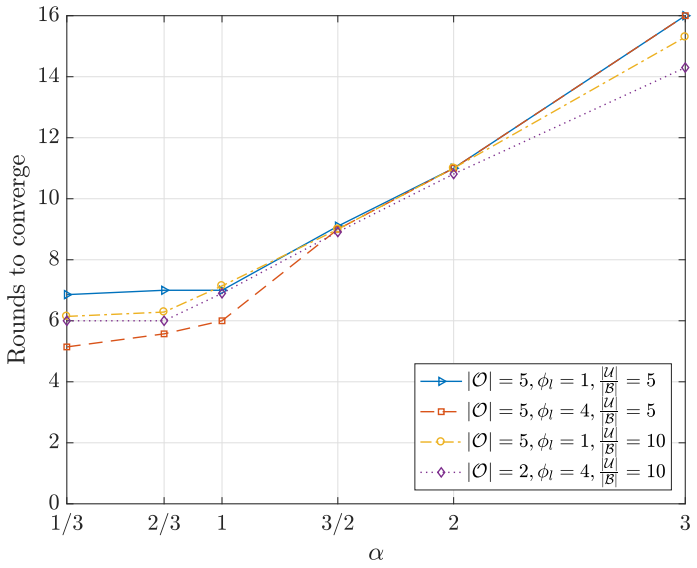

Figure 4.3: Average number of rounds until convergence for different scenarios.

load distributions; and (iv) *mixed*: half of the slices with a uniform spatial load distribution and the other half with a non-uniform one. In all cases, we have 4 slices with equal shares. The results, depicted in Figure 4.4, show that the gains are larger for scenarios with uneven and complementary traffic loads; indeed, in this case different slices need their resources at different base stations and thus there is a higher gain from dynamically sharing the resources. We further observe that larger α values result in smaller gain; this is because slices are less elastic with larger α, which limits the ability to exploit statistical multiplexing.

4.7 Conclusions

In this chapter we have analyzed a 'share-constrained proportional allocation' framework for network slicing. The framework allows slices to customize the

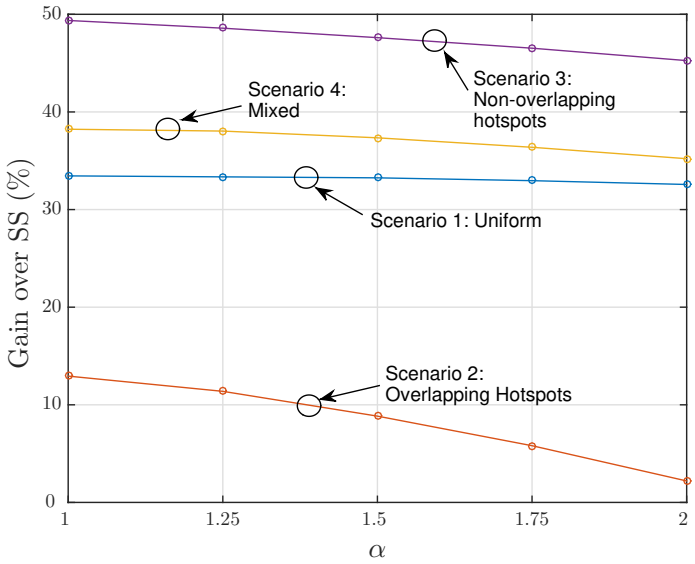

Figure 4.4: Gain over Static slicing for different traffic models and α values.

resource allocation to their users, leading to a *network slicing game* in which each slice reacts to the settings of the others. Our main conclusion is that the framework provides an *effective* and *implementable* scheme for dynamically sharing resources across slices. Indeed, this scheme involves simple operations at base stations and incurs a limited signaling between the slices and the infrastructure. Our results confirm system stability (*best response dynamics* converge), substantial gains over *static slicing*, and fairness of the allocations (*envy-freeness*). Moreover, as long as the majority of the slices do not choose α_o values larger than 1 (i.e., they do not all demand very homogeneous rate distributions), the overall performance is close to optimal (*price of anarchy* is very small). Thus, in this case the flexibility provided by this framework comes at no cost. If a substantial number of slices choose higher α_o's, then we pay a (small) price for enabling slice customization.

4.8 Proofs of chapter results

4.8.1 Proof of Lemma 2

Given the weight allocation under static slicing, $\mathbf{w}^{ss}$, and the weights of the other slices under dynamic sharing, $\mathbf{w}^{-o}$, we consider the following weight allocation for slice o:

$$w_u = \frac{w_u^{ss}}{\sum_{u \in \mathcal{U}_{b(u)}^o} w_u^{ss}} \frac{a_{b(u)}^o(\mathbf{w}^{-o})}{\sum_{b' \in \mathcal{B}_o} a_{b'}^o(\mathbf{w}^{-o})} s_o, \tag{4.4}$$

where $\mathcal{B}_o$ is the set of base stations where slice o has customers.

We define $\rho_u^o(\mathbf{w}^{ss})$ as the ratio between the weight of user u under static slicing and the sum of the weights of all the users of the same slice in the base station, i.e.,

$$\rho_u^o(\mathbf{w}^{ss}) \doteq \frac{w_u^{ss}}{\sum_{u \in \mathcal{U}_{b(u)}^o} w_u^{ss}}$$

where we have dropped the terms $\mathbf{w}^{-o}$ and $\mathbf{w}^{ss}$ from $a_{b(u)}^o(\mathbf{w}^{-o})$ and $\rho_u^o(\mathbf{w}^{ss})$ for readability purposes.

With the allocation given by (4.4), for two users u and u' of slice o it holds

$$\frac{w_u}{w_{u'}} = \frac{\rho_u^o}{\rho_{u'}^o} \frac{a_{b(u)}^o}{a_{b(u')}^o} \tag{4.5}$$

Furthermore, it also holds that

$$d_b^o = \sum_{u \in \mathcal{U}_b^o} w_u = \sum_{u \in \mathcal{U}_b^o} \rho_u^o \frac{a_{b(u)}^o}{\sum\limits_{b' \in \mathcal{B}_o} a_{b'}^o} s_o = \frac{a_{b(u)}^o}{\sum\limits_{b' \in \mathcal{B}_o} a_{b'}^o} s_o = \frac{w_u}{\rho_u^o}$$

for $u \in \mathcal{U}_b^o$. From the above expression, we have

$$\frac{\rho_u^o l_{b(u)}(\mathbf{w})}{\rho_{u'}^o l_{b(u')}(\mathbf{w})} = \frac{\rho_u^o \left(a_{b(u)}^o + d_{b(u)}^o \right)}{\rho_{u'}^o \left(a_{b(u')}^o + d_{b(u')}^o \right)} = \frac{\rho_u^o \left(a_{b(u)}^o + \frac{w_u}{\rho_u^o} \right)}{\rho_{u'}^o \left(a_{b(u')}^o + \frac{w_{u'}}{\rho_{u'}^o} \right)},$$

and combining this with (4.5):

$$\frac{\rho^o_u l_{b(u)}(\mathbf{w})}{\rho^o_{u'} l_{b(u')}(\mathbf{w})} = \frac{\rho^o_u \left(a^o_{b(u)} + \frac{a^o_{b(u)}}{a^o_{b(u')}} \frac{w_{u'}}{\rho^o_{u'}} \right)}{\rho^o_{u'} \left(a^o_{b(u')} + \frac{w_{u'}}{\rho^o_{u'}} \right)} = \frac{\rho^o_u a^o_{b(u)} \left(1 + \frac{w_{u'}}{a^o_{b(u')} \rho^o_{u'}} \right)}{\rho^o_{u'} a^o_{b(u')} \left(1 + \frac{w_{u'}}{a^o_{b(u')} \rho^o_{u'}} \right)} = \frac{\rho^o_u a^o_{b(u)}}{\rho^o_{u'} a^o_{b(u')}}$$

From the above,

$$w_u = \frac{w_u}{\sum_{u' \in \mathcal{U}^o} w_{u'}} s_o = \frac{s_o}{\sum_{u' \in \mathcal{U}^o} \frac{w_{u'}}{w_u}} = \frac{s_o}{\sum\limits_{u' \in \mathcal{U}^o} \frac{\rho^o_{u'} a^o_{b(u')}}{\rho^o_u a^o_{b(u)}}} = \frac{s_o}{\sum\limits_{u' \in \mathcal{U}^o} \frac{\rho^o_{u'} l_{b(u')}(\mathbf{w})}{\rho^o_u l_{b(u)}(\mathbf{w})}}$$

$$= \frac{\rho^o_u l_{b(u)}(\mathbf{w})}{\sum_{u' \in \mathcal{U}^o} \rho^o_{u'} l_{b(u')}(\mathbf{w})} s_o = \frac{\rho^o_u l_{b(u)}(\mathbf{w})}{\sum_{b' \in \mathcal{B}_o} l_{b'}(\mathbf{w})} s_o$$

Since $\mathcal{B}_o \subseteq \mathcal{B}$:

$$\sum_{b \in \mathcal{B}_o} l_b(\mathbf{w}) \leq \sum_{b \in \mathcal{B}} l_b(\mathbf{w}) = 1$$

and thus

$$w_u \geq \rho^o_u l_{b(u)}(\mathbf{w}) s_o,$$

from which

$$r_u(\mathbf{w}) = \frac{w_u}{l_{b(u)}(\mathbf{w})} c_u \geq \frac{\rho^o_u l_{b(u)}(\mathbf{w}) s_o}{l_{b(u)}(\mathbf{w})} c_u = \rho^o_u s_o c_u = r^{ss}_u.$$

The above holds for all $u \in \mathcal{U}$, which proves the lemma. □

4.8.2 Proof of Theorem 8

This result follows from Lemma 2. Given the configuration of the other slices, there exists a configuration for a given slice under which all its users obtain at least the same throughput as with static slicing, and thus the slice's utility with this configuration is at least as high. As a consequence, in a NE the slice will receive a utility no smaller than this value. □

4.8.3 Proof of Lemma 3

Let us start for $\alpha_o \neq 1$. The best response of slice o is given by

$$\mathbf{w}^o = \arg\max_{\mathbf{w}'^o} \sum_{u \in \mathcal{U}_o} \frac{\phi_u}{1-\alpha_o} \left(\frac{w'_u c_u}{a^o_{b(u)}(\mathbf{w}'^{-o}) + d^o_{b(u)}(\mathbf{w}'^o)} \right)^{1-\alpha_o}$$
$$\text{subject to: } \sum_{u \in \mathcal{U}_o} w'_u = s^o, \quad w'_u \geq 0, \quad \forall u \in \mathcal{U}_o$$

The Lagrangian for this optimization problem is given by

$$\mathcal{L}(\mathbf{w}, \lambda) = \sum_{u \in \mathcal{U}_o} \frac{\phi_u}{1-\alpha_o} \left(\frac{w_u c_u}{a^o_b(\mathbf{w}^{-o}) + d^o_b(\mathbf{w}'^o)} \right)^{1-\alpha_o} - \lambda_o \left(\sum_{u \in \mathcal{U}_o} w_u - s_o \right)$$

The partial derivative of the above function with respect to w_u is given by

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{w}, \lambda)}{\partial w_u} &= \frac{\phi_u c_u^{1-\alpha_o} \cdot (l_b(\mathbf{w}) - w_u)}{w_u^{\alpha_o} \cdot l_b(\mathbf{w})^{2-\alpha_o}} - \sum_{u' \in \mathcal{U}^o_{b(u)} \setminus \{u\}} \frac{\phi_{u'} c_{u'}^{1-\alpha_o} w_{u'}^{1-\alpha_o}}{l_b(\mathbf{w})^{2-\alpha_o}} - \lambda_o \\ &= \frac{\phi_u c_u^{1-\alpha_o} \cdot l_b(\mathbf{w})}{w_u^{\alpha_o} \cdot l_b(\mathbf{w})^{2-\alpha_o}} - \sum_{u' \in \mathcal{U}^o_{b(u)}} \frac{\phi_{u'} c_{u'}^{1-\alpha_o} w_{u'}^{1-\alpha_o}}{l_b(\mathbf{w})^{2-\alpha_o}} - \lambda_o \end{aligned}$$

From the above, for two users u, u' in the same base station we have

$$w_u^{\alpha_o} = \frac{\phi_u}{\phi_{u'}} \left(\frac{c_u}{c_{u'}} \right)^{1-\alpha_o} w_{u'}^{\alpha_o} \tag{4.6}$$

Combining the above two equations yields

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{w}, \lambda)}{\partial w_u} &= \frac{\phi_u \cdot c_u^{1-\alpha_o}}{w_u^{\alpha_o} \cdot l_b(\mathbf{w})^{2-\alpha_o}} \left(l_b(\mathbf{w}) - \sum_{u' \in \mathcal{U}^o_{b(u)}} \frac{\phi_{u'} c_{u'}^{1-\alpha_o}}{\phi_u c_u^{1-\alpha_o}} w_{u'}^{1-\alpha_o} w_u^{\alpha_o} \right) - \lambda_o \\ &= \frac{\phi_u \cdot c_u^{1-\alpha_o}}{w_u^{\alpha_o} \cdot l_b(\mathbf{w})^{2-\alpha_o}} \left(l_b(\mathbf{w}) - \sum_{u' \in \mathcal{U}^o_{b(u)}} w_{u'} \right) - \lambda_o. \end{aligned}$$

Equaling the above expression for two different users u and u' at different base stations, we obtain the following expression, which holds for any pair of users of slice o (at the same or different base stations):

$$\frac{w_u}{w_{u'}} = \frac{\beta_u}{\beta_v} \frac{\frac{\left(a^o_{b(u)}(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_{b(u)}(\mathbf{w}^{-o})+d^o_{b(u)}(\mathbf{w}^o)\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\left(a^o_{b(u')}(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_{b(u')}(\mathbf{w}^{-o})+d^o_{b(u')}(\mathbf{w}^o)\right)^{\frac{2}{\alpha_o}-1}}} \tag{4.7}$$

where $\beta_u := (\phi_u)^{\frac{1}{\alpha_o}}(c_u)^{\frac{1}{\alpha_o}-1}$. From the above, we obtain (4.2) by normalizing. In order to prove that the resulting non-linear system of equations has a unique solution, we proceed as follows. Let $d^o_b = \sum_{u\in\mathcal{U}^o_b} w_u$. From (4.6),

$$d^o_b = w_u \sum_{u'\in\mathcal{U}^o_b} \frac{\beta_{u'}}{\beta_u}$$

Combining the above with (4.7) yields

$$\frac{d^o_b}{d^o_{b'}} = \frac{\beta_u}{\beta_v} \frac{\sum_{v'\in\mathcal{U}^o_{b'}} \frac{\beta_{v'}}{\beta_v}}{\sum_{v'\in\mathcal{U}^o_b} \frac{\beta_{v'}}{\beta_u}} \frac{\frac{\left(a^o_b(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_b(\mathbf{w}^{-o})+d^o_b\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\left(a^o_{b'}(\mathbf{w}^{-o})\right)^{\frac{1}{\alpha_o}}}{\left(a^o_{b'}(\mathbf{w}^{-o})+d^o_{b'}\right)^{\frac{2}{\alpha_o}-1}}}$$

which is equivalent to

$$d^o_b \left(a^o_b(\mathbf{w}^{-o}) + d^o_b\right)^{\frac{2}{\alpha_o}-1} = K d^o_{b'} \left(a^o_{b'}(\mathbf{w}^{-o}) + d^o_{b'}\right)^{\frac{2}{\alpha_o}-1}$$

where K is some constant. Then, if we fix d^o_b to some positive value, there exists a unique positive value of $d^o_{b'}$ that satisfies the above equation. Indeed, the lhs of the equation will be fixed to some finite value larger than 0, while the rhs grows from 0 to ∞ as we increase $d^o_{b'}$. Moreover, the larger the value of d^o_b, the larger the

resulting $d^o_{b'}$, since both the lhs and the rhs of the equation are increasing functions of d^o_b and $d^o_{b'}$, respectively.

From the above, we can compute the $d^o_{b'}$ value of each base stations as a function of a single d^o_b. Once we have all $d^o_{b'}$ values, we can uniquely compute the user weights w_u, which are an increasing function of $d^o_{b'}$ (and thus of d^o_b). Inserting the resulting weights into $\sum_{u \in \mathcal{U}_o} w_u(d^o_b) = s_o$, we have an equation with a single unknown, d^o_b. This equation has a unique solution, as the lhs is an increasing function of d^o_b and the rhs is constant. Computing the resulting d^o_b value, and obtaining from this value the corresponding w_u values, we have a solution to the system. Since all relationships are bijective, this is the only solution of the system.

The case $\alpha_o = 1$ is proven employing a similar argument. Indeed, by repeating the same steps as above for for $U_o(\mathbf{w}) = \sum_{u \in \mathcal{U}_o} \phi_u \log r_u(\mathbf{w})$, it is easy to verify the best response for this case corresponds to the expression given by (4.2) for $\alpha_o = 1$. □

4.8.4 Proof of Theorem 9

Let $\mathbf{R}_o$ be the implicit function that denotes the best response of slice o, i.e., $\mathbf{w}^o = \mathbf{R}_o(\mathbf{w}^o, \mathbf{w}^{-o})$. We start by proving that $\mathbf{R}^o$ is a continuous and differentiable function for $a^o_b \neq 0 \ \forall b$. Note that $\mathbf{R}_o$ is a continuous and differentiable function of $\boldsymbol{a}^o = (a^o_b : b \in \mathcal{B}_o)$ when $\boldsymbol{w}^o$ is fixed. So it follows from the implicit function theorem that the best response is continuous and differentiable in $\boldsymbol{a}^o$. As $\boldsymbol{a}^o$ is continuous function of $\mathbf{w}^{-o}$, it follows that the best response is a continuous and differentiable function of the other slices' weights.

Now, let us define a perturbed game $G^{(\varepsilon)}$ for some $\varepsilon > 0$, in which there is an additional slice which places a weight of ε in each base station.[6] The existence of a NE for this perturbed game is guaranteed by the result of [94] since (i) the utility is a concave function, (ii) it is continuous (as given by the previous lemma),[7] and (iii) the game strategy space set of a slice, given by $\mathcal{S}^o = \{\mathbf{w}^o \in \mathbf{R}^o \mid \mathbf{w}^o \geq \mathbf{0}$ and $\sum_{u \in \mathcal{U}^o} w_u = s^o\}$, is compact and convex.[8] Moreover, Lemma 5 (provided with the supplementary material) shows that the weights in the NE must be greater than or equal to a value δ for any $\varepsilon < \varepsilon_{max}$.[9] Building on the result of this lemma, we proceed as follows. Let us consider a decreasing sequence $\varepsilon^k \to 0$, where $\varepsilon^0 \leq \varepsilon_{max}$, and let $\mathbf{w}^{(\varepsilon^k)}$ denote the Nash Equilibrium of game $G^{(\varepsilon^k)}$. Since the strategy space $\mathcal{S} = \mathcal{S}^1 \times ... \times \mathcal{S}^{|\mathcal{O}|}$ is a compact set, there exists a subsequence ε^{k_n} such that $\varepsilon^{k_n} \to 0$ and $\mathbf{w}^{(\varepsilon^{k_n})} \to \mathbf{w}^{(0)}$. Note that $w_u^{(\varepsilon^{k_n})} \geq \delta \ \forall u$ and therefore $w_u^{(0)} \geq \delta \ \forall u$. Now, let us define function

$$\mathbf{g}(\varepsilon, \mathbf{w}) = \mathbf{R}^{(\varepsilon)}(\mathbf{w}) - \mathbf{w}$$

where $\mathbf{R}^{(\varepsilon)}(\mathbf{w})$ is the best response to $\mathbf{w}$ in the game $G^{(\varepsilon)}$. Note that $\mathbf{g}(\varepsilon, \mathbf{w})$ is equal to zero at the NE of the perturbed game $G^{(\varepsilon^k)}$. Furthermore, from the above lemma we have that at this NE it holds $w_u > 0 \ \forall u$ (even as $\varepsilon \to 0$), and we

[6] In the perturbed game, the shares are rescaled so that they still sum 1, i.e., $\sum_{o \in \mathcal{O}} s_o + |\mathcal{B}|\varepsilon = 1$.

[7] Note that in the perturbed game $a_b^o \geq \varepsilon > 0$.

[8] The statement of the theorem of [94] requires that $U_o(\mathbf{w})$ is defined in the strategy space $\mathcal{S}^o$, which is not satisfied for $\alpha_o = 1$ since in this case $U_o(\mathbf{w}) \to -\infty$ when $w_u = 0$ for some $u \in \mathcal{U}_o$. However, the proof of this theorem only requires that the mapping $\mathbf{w}^o = R_o(\mathbf{w}^{-o})$ is defined in $\mathcal{S}^o$, which is satisfied for $\alpha_o = 1$ (and indeed this mapping never yields $w_u = 0$, as this would not maximize the slice's utility).

[9] Note that, as the sum of the weights of all players in the game is 1, i.e., $\sum_{o \in \mathcal{O}} s_o + |\mathcal{B}|\varepsilon = 1$, we have the following upper bound for ε: $\varepsilon \leq \hat{\varepsilon} = 1/|\mathcal{B}|$.

further have that this function is continuous and differentiable for $w_u > 0 \, \forall u$. Thus, $\mathbf{g}(\varepsilon^{k_n}, \mathbf{w}^{(\varepsilon^{k_n})}) = \mathbf{0}$ and

$$\lim_{\varepsilon^{k_n} \to 0} \mathbf{g}(\varepsilon^{k_n}, \mathbf{w}^{(\varepsilon^{k_n})}) = \mathbf{g}(0, \mathbf{w}^{(0)}) = \mathbf{0}$$

implying that indeed $\mathbf{w}^{(0)}$ exists and is a Nash Equilibrium of our (non perturbed) game. □

4.8.5 Proof of Lemma 4

We prove the lemma by showing that there exists a scenario for which there is no NE. Let us consider a network with two slices, 1 and 2, each of them with two users (u11 and u12 for slice 1 and u21 and u22 for slice 2) and with $s_1 = s_2 = 1/2$. Let users u11 and u21 be associated with one base station (base station 1) and the other two users (u12 and u22) associated to base station 2. Furthermore, let c_u of all users be equal to 1 except for u22, for which $c_u = 2$.

With utility functions $U^o(\mathbf{w}) = \min_{u \in \mathcal{U}^o} (r_u(\mathbf{w}))$, it follows that the best response of slice 1 to a given allocation of slice 2 satisfies

$$\frac{w_{11}}{w_{11} + w_{21}} = \frac{w_{12}}{w_{12} + w_{22}}. \tag{4.8}$$

Similarly, the following equation holds for the best response of slice 2:

$$\frac{w_{21}}{w_{11} + w_{21}} = 2\frac{w_{22}}{w_{12} + w_{22}}. \tag{4.9}$$

From (4.8) it follows that

$$\frac{w_{21}}{w_{11} + w_{21}} = \frac{w_{22}}{w_{12} + w_{22}}.$$

and combining the above with (4.9) we obtain

$$\frac{w_{22}}{w_{12}+w_{22}} = 2\frac{w_{22}}{w_{12}+w_{22}}.$$

Note that there exists no w_{22} that satisfies the above equation (except $w_{22} = 0$ which is not a possible setting for the weight best response). We therefore conclude that there exists no NE. □

4.8.6 Proof of Theorem 10

Let us first focus on the case with sequential updates. We shall denote time as slotted $\{0, 1, ..., t, ...\}$ and assume a single slice makes an update each time slot. Without loss of generality, we will index slices $\{1, 2, ..., |\mathcal{O}|\} = \mathcal{O}$ according to their updating order in a round. We let $\mathbf{w}(t) = (\mathbf{w}^o(t) \colon o \in \mathcal{O})$ be the weights of all slices at the end the time slot t update, where $\mathbf{w}^o(t) = (w_u(t) \colon u \in \mathcal{U}^o)$. Suppose that slices have arbitrary positive initial weight vectors at time zero denoted $\mathbf{w}(0) = (\mathbf{w}^1(0), \mathbf{w}^2(0), ..., \mathbf{w}^{|\mathcal{O}|}(0))$. Consequently, slice 1 will update its weights at time slots: $\{1, |\mathcal{O}| + 1, ..., r \cdot |\mathcal{O}| + 1\}$, corresponding to rounds $\{0, 1, ..., r, ...\}$.

We will further define $\mathbf{\Delta w}^o(t+1) = (\Delta w_u(t+1) \colon u \in \mathcal{U}^o)$, where $\mathbf{\Delta w}^o(t+1) = (\Delta w_u(t+1) \colon u \in \mathcal{U}^o)$ such that,

$$w_u(t+1) = w_u(t)(1 + \Delta w_u(t+1)), \quad \forall o \in \mathcal{O}, u \in \mathcal{U}^o$$

where $1 + \Delta w_u(t+1)$ captures the relative change in slice o's weight update at time slot $t+1$. Furthermore, to capture the overall changes in slices weights at the end of each round, we shall define $\boldsymbol{\omega}(0) = \mathbf{w}(0)$, $\boldsymbol{\omega}(r) = (\boldsymbol{\omega}^o(r) \colon o \in \mathcal{O})$ where

$\boldsymbol{\omega}^o(r) = \mathbf{w}^o(r \cdot |\mathcal{O}| + 1)$ and $\boldsymbol{\Delta\omega}^o(r)$ such that $\boldsymbol{\Delta\omega}(r) = (\Delta\omega_u^o(r) \colon u \in \mathcal{U}^o)$. For all $o \in \mathcal{O}$, we define

$$\overline{\Delta}\omega^o(r) := \max_{u \in \mathcal{U}^o} \Delta\omega_u^o(r), \quad \underline{\Delta}\omega^o(r) := \min_{u \in \mathcal{U}^o} \Delta\omega_u^o(r).$$

The slices responses at the end of each round are captured by the sequence $\boldsymbol{\omega}(r)$ for $r = 1, 2, ...$ and the dynamics are given by $\boldsymbol{\omega}(r+1) = \mathbf{R}(\boldsymbol{\omega}(r))$, where $\mathbf{R}(\boldsymbol{\omega}(r))$ is the result of applying sequentially the best response for each of the slices. Note that $\boldsymbol{\omega}(r)$ is in the set $\mathcal{S} = \mathcal{S}^1 \times \mathcal{S}^2 \times ... \times \mathcal{S}^{|\mathcal{O}|}$ where:

$$\mathcal{S}^o = \{\boldsymbol{\omega}^o \in \mathbf{R}^o \mid \boldsymbol{\omega}^o \geq \mathbf{0} \text{ and } \sum_{b \in \mathcal{B}} n_b^o \omega_b^o = s^o\}.$$

Let us define a function:

$$V(\boldsymbol{\omega}(r)) := \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta}\omega^o(r+1), \frac{1}{1 + \underline{\Delta}\omega^o(r+1)}\right) - 1$$

Notice that $V(\boldsymbol{\omega}(r))$ is a function of the relative weight changes at round $r+1$ given the weights at round r, $\boldsymbol{\omega}(r)$.

Note that, unless algorithm converged (i.e., $\boldsymbol{\omega}(r+1) = \boldsymbol{\omega}(r)$ and $V(\boldsymbol{\omega}(r)) = 0$), there should at least one slice with a user u for which $\Delta\omega_u^o(r+1) > 0$ and a user v for which $\Delta\omega_v^o(r+1) < 0$. Thus, in this case it must be that $V(\boldsymbol{\omega}(r)) > 0$ and

$$\max_{o \in \mathcal{O}} \left(1 + \overline{\Delta}\omega^o(r+1), \frac{1}{1 + \underline{\Delta}\omega^o(r+1)}\right) > 1$$

Note also that by Lemma 6 (provided as supplementary material) it follows that if $\boldsymbol{\omega}(r+1) \neq \boldsymbol{\omega}(r)$, then

$$\max_{o \in \mathcal{O}} \left(1 + \overline{\Delta}\omega^o(r+1), \frac{1}{1 + \underline{\Delta}\omega^o(r+1)}\right) < \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta}\omega^o(r), \frac{1}{1 + \underline{\Delta}\omega^o(r)}\right), \tag{4.10}$$

which is equivalent to $V(\boldsymbol{\omega}(r+1)))) < V(\boldsymbol{\omega}(r))$.

Note that, since $\underline{\Delta}\omega^o > -1$, $V(\boldsymbol{\omega}(r))$ is continuous. Furthermore, the best response functions governing Δ are also continuous, as shown in Theorem 3.

In summary, the above results show that $V(\boldsymbol{\omega}(r))$ is continuous, non-negative and decreasing each round. So, we might expect it to converge to 0 in which case the slices weights must have converged. We show this by contradiction. Suppose $V(\boldsymbol{\omega}(r))$ converges instead to $\epsilon > 0$, so $V(\boldsymbol{\omega}(0)) \geq V(\boldsymbol{\omega}(r)) \geq \epsilon$ for all r. Let us define $\mathcal{V} = \{\boldsymbol{\omega} | V(\boldsymbol{\omega}(0)) \geq V(\boldsymbol{\omega}) \geq \epsilon\}$ and let $\mathcal{C} = \{\boldsymbol{\omega} | \boldsymbol{\omega} \in \mathcal{S} \cap \mathcal{V}\}$. From this, $\mathcal{V}$ is closed, and since $\mathcal{S}$ is compact, so is $\mathcal{C}$. From the continuity of V we have that:

$$\delta = \max_{\boldsymbol{\omega} \in C} V(\mathbf{R}(\boldsymbol{\omega}))) - V(\boldsymbol{\omega}) < 0$$

Clearly it is not possible $V(\boldsymbol{\omega}(r)) \geq \epsilon$ since each round it decreases by at least δ. It follows that the weights must converge when $V(\boldsymbol{\omega}(r)) = 0$, which terminates the proof for the case with sequential updates.

Let us now focus on the case of simultaneous updates. The proof goes along the lines of the proof for the above case.

Similarly to the previous case, we shall denote time as slotted $\{0, 1, ..., t, ...\}$. With simultaneous updates, every slice makes an update each time slot. We let $\mathbf{w}(t) = (\mathbf{w}^o(t) \colon o \in \mathcal{O})$ be the weights of all slices at the end the time slot t update, where $\mathbf{w}^o(t) = (w_u(t) \colon u \in \mathcal{U}^o)$. As before, we define $\boldsymbol{\Delta}\mathbf{w}^o(t+1) = (\Delta w_u(t+1) \colon u \in \mathcal{U}^o)$, where $\boldsymbol{\Delta}\mathbf{w}^o(t+1) = (\Delta w_u(t+1) \colon u \in \mathcal{U}^o)$ and $\boldsymbol{\Delta}\mathbf{w}(r) = (\boldsymbol{\Delta}\mathbf{w}^o(r) \colon o \in \mathcal{O})$ such that,

$$w_u(t+1) = w_u(t)(1 + \Delta w_u(t+1)), \quad \forall o \in \mathcal{O}, u \in \mathcal{U}^o$$

where $1+\Delta w_u(t+1)$ captures the relative change in slice o's weight update at time slot $t+1$. For all $o \in \mathcal{O}$, we define

$$\overline{\Delta} w^o(t) := \max_{u \in \mathcal{U}^o} \Delta w_u^o(t), \quad \underline{\Delta} w^o(t) := \min_{u \in \mathcal{U}^o} \Delta w_u^o(t).$$

We first show that if the game has not converged to a Nash equilibrium, i.e. $\boldsymbol{\Delta}\mathbf{w}(r) \neq \mathbf{0}$ for $r > 1$, then:

$$\max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} w(t+1), \frac{1}{1+\underline{\Delta} w(t+1)}\right) < \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} w(t), \frac{1}{1+\underline{\Delta} w(t)}\right).$$

The above can be seen as follows. Defining $\boldsymbol{\Delta}\mathbf{a}^o(t) = (\Delta a_b^o(t) \colon b \in \mathcal{B})$ such that $a_b^o(t) = a_b^o(t-1)(1+\Delta a_b^o(t))$, from the proof of Lemma 6 we have:

$$\max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} w^o(t+1), \frac{1}{1+\underline{\Delta} w^o(t+1)}\right) < \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} a^o(t), \frac{1}{1+\underline{\Delta} a^o(t)}\right),$$

where $\overline{\Delta} a^o(t) = \max_{o \in \mathcal{O}} \Delta a^o(t)$ and $\underline{\Delta} a^o(t) = \min_{o \in \mathcal{O}} \Delta a^o(t)$. Also, it holds that

$$1 + \overline{\Delta} a^o(t) \leq \max_{o' \in \mathcal{O}, o' \neq o} 1 + \overline{\Delta} w^{o'}(t).$$

and similarly:

$$1 + \underline{\Delta} a^o(t) \geq \min_{o' \in \mathcal{O}, o' \neq o} 1 + \underline{\Delta} w^{o'}(t).$$

Therefore:

$$\begin{aligned}\max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} w(t+1), \frac{1}{1+\underline{\Delta} w(t+1)}\right) &< \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} a^o(t), \frac{1}{1+\underline{\Delta} a^o(t)}\right) \\ &\leq \max_{o \in \mathcal{O}} \left(1 + \overline{\Delta} w^o(t), \frac{1}{1+\underline{\Delta} w^o(t)}\right)\end{aligned}$$

Taking into account that in the case of simultaneous updates each time slot corresponds to a round (where all slices update their weights), the above result is equivalent to the result given by (4.10) for the case of sequential updates. The rest of the proof then follows exactly the one for sequential updates. □

4.8.7 Proof of Theorem 11

We first show that an optimal (not necessarily unique) solution to the centralized problem is given by $\mathbf{w}^*$ which assigns weights to all users of a given slice proportionally to their priorities, i.e., $w_u^* = \phi_u s_o,\ \forall u \in \mathcal{U}_o$. To prove this we only need to show that $U(\mathbf{w}^*) \geq U(\mathbf{w})$ for any other feasible weight vector $\mathbf{w}$. To that end, consider

$$U(\mathbf{w}^*) - U(\mathbf{w}) = \sum_{o\in\mathcal{O}} \sum_{u\in\mathcal{U}_o} \phi_u \left(\log\left(\frac{w_u^* c_u}{l_b(\mathbf{w}^*)}\right) - \log\left(\frac{w_u c_u}{l_b(\mathbf{w})}\right)\right)$$

Let us denote the distributions induced by $\mathbf{w}^*$ and $\mathbf{w}$ respectively as: $\mathbf{p}^b(\mathbf{w}) = (p_u^b(\mathbf{w}) = \frac{w_u}{l_b(\mathbf{w})} \colon u \in \mathcal{U}_b)$ and $\mathbf{p}^b(\mathbf{w}^*) = (p_u^b(\mathbf{w}^*) = \frac{w_u}{l_b(\mathbf{w}^*)} \colon u \in \mathcal{U}_b)$. Since $\boldsymbol{\phi} = \mathbf{w}^*$, we have

$$\begin{aligned} U(\mathbf{w}^*) - U(\mathbf{w}) &= \sum_{b\in\mathcal{B}} l_b(\mathbf{w}^*) \sum_{o\in\mathcal{O}} \sum_{u\in\mathcal{U}_o^b} p_u^b(w^*) \log\left(\frac{p_u^b(w^*)}{p_u^b(w)}\right) \\ &= \sum_{b\in\mathcal{B}} l_b(\mathbf{w}^*) D(\mathbf{p}^b(\mathbf{w}^*)||\mathbf{p}^b(\mathbf{w})) \end{aligned}$$

where $D(\mathbf{p}^b(\mathbf{w}^*)||\mathbf{p}^b(\mathbf{w}))$ is the Kullback-Leibler divergence, between the distributions induced by $\mathbf{w}^*$ and $\mathbf{w}$ respectively, i.e., $\mathbf{p}^b(\mathbf{w}^*)$ and $\mathbf{p}^b(\mathbf{w})$. It is known [66] that $D(\mathbf{p}^b(\mathbf{w}^*)||\mathbf{p}^b(\mathbf{w})) \geq 0$ and 0 only when $\mathbf{p}^b(\mathbf{w}) = \mathbf{p}^b(\mathbf{w}^*)$ Hence it follows that $\mathbf{w}^*$ is optimal.

We next show that $U(\mathbf{w}^*) - U(\mathbf{w}) \leq \log(e)$ holds when $\mathbf{w}$ is a Nash Equilibrium of the distributed resource allocation game and $\mathbf{w}^*$ an optimal solution. To show this, we proceed as follows. Since in the Nash Equilibrium each slice maximizes its utility given the allocation of the other slices,

$$\sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u}{l_{b(u)}(\mathbf{w})}\right) \geq \sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u^*}{d^o_{b(u)}(\mathbf{w}^*) + a^o_{b(u)}(\mathbf{w})}\right)$$

Given that $d^o_{b(u)}(\mathbf{w}^*) + a^o_{b(u)}(\mathbf{w}) \leq l_{b(u)}(\mathbf{w}) + l_{b(u)}(\mathbf{w}^*)$,

$$\sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u}{l_{b(u)}(\mathbf{w})}\right) \geq \sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u^*}{l_{b(u)}(\mathbf{w}) + l_{b(u)}(\mathbf{w}^*)}\right)$$

From the above it follows that

$$\begin{aligned}
&\sum_{u \in \mathcal{U}^o} \phi_u \log(r_u(\mathbf{w}^*)) - \sum_{u \in \mathcal{U}^o} \phi_u \log(r_u(\mathbf{w})) \\
&\quad \leq \sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u^* c_u}{l_{b(u)}(\mathbf{w}^*)}\right) - \sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{w_u^* c_u}{l_{b(u)}(\mathbf{w}) + l_{b(u)}(\mathbf{w}^*)}\right) \\
&\quad = -\sum_{u \in \mathcal{U}^o} \phi_u \log\left(\frac{l_{b(u)}(\mathbf{w}^*)}{l_{b(u)}(\mathbf{w}) + l_{b(u)}(\mathbf{w}^*)}\right)
\end{aligned}$$

Summing the above over all slices weighted by the corresponding shares yields

$$U(\mathbf{w}^*) - U(\mathbf{w}) \leq -\sum_{u \in \mathcal{U}} \phi_u s_o \log\left(\frac{l_{b(u)}(\mathbf{w}^*)}{l_{b(u)}(\mathbf{w}) + l_{b(u)}(\mathbf{w}^*)}\right)$$

Given $w_u^* = \phi_u s_o$, we have

$$\begin{aligned}
U(\mathbf{w}^*) - U(\mathbf{w}) &\leq -\sum_{b \in \mathcal{B}} \log\left(\frac{l_b(\mathbf{w}^*)}{l_b(\mathbf{w}) + l_b(\mathbf{w}^*)}\right)^{\sum_{u \in \mathcal{U}_b} w_u^*} \\
&= -\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} w_u \log\left(\frac{l_b(\mathbf{w}^*)}{l_b(\mathbf{w}) + l_b(\mathbf{w}^*)}\right)^{\frac{\sum_{v \in \mathcal{U}_b} w_v^*}{\sum_{v \in \mathcal{U}_b} w_v}} \\
&= -\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} w_u \log\left(\frac{l_b(\mathbf{w}^*)/l_b(\mathbf{w})}{1 + l_b(\mathbf{w}^*)/l_b(\mathbf{w})}\right)^{\frac{l_b(\mathbf{w}^*)}{l_b(\mathbf{w})}}
\end{aligned}$$

and, given that $(x/(1+x))^x > 1/e$ for $x \geq 0$, this yields

$$U(\mathbf{w}^*) - U(\mathbf{w}) \leq \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} w_u \log(e) = \log(e).$$

Finally, we show that there exists some scenario for which $U(\mathbf{w}^*) - U(\mathbf{w}) = \log(e)$. Let us consider a scenario with two slices with shares s_1 and s_2, respectively. There are two base stations. Slice 1 has $m+1$ users, m associated to base station 1 and one associated to base station 2. Slice 2 has one user associated to base station 2. All users have $c_{ub} = 1$. Under the optimal allocation:

$$U(\mathbf{w}^*) = \frac{s_1}{m+1} m \log\left(\frac{1}{m}\right) + \frac{s_1}{m+1} \log\left(\frac{\frac{s_1}{m+1}}{\frac{s_1}{(m+1)} + s_2}\right) + s_2 \log\left(\frac{s_2}{\frac{s_1}{(m+1)} + s_2}\right),$$

and under the Nash equilibrium

$$U(\mathbf{w}) = \frac{s_1}{m+1} m \log\left(\frac{1}{m}\right) + \frac{s_1}{m+1} \log\left(\frac{s_1}{s_1+s_2}\right) + s_2 \log\left(\frac{s_2}{s_1+s_2}\right).$$

For $m \to \infty$ this yields $U(\mathbf{w}^*) = s_1 \log\left(\frac{1}{m}\right) + s_2 \log(1)$ and $U(\mathbf{w}) = s_1 \log\left(\frac{1}{m}\right) + s_2 \log\left(\frac{s_2}{s_1+s_2}\right)$. From this,

$$U(\mathbf{w}) - U(\mathbf{w}^*) = s_2 \log\left(\frac{s_2}{s_1 + s2}\right)$$

which tends to $-\log(e)$ when $s_1 \to 1$ and $s_2 \to 0$. □

4.8.8 Proof of Theorem 12

Let us consider two slices, o and o', that have the same share s_o. Let the utility function of slice o be $U^o = \sum_{u \in \mathcal{U}^o} \phi_u \log(r_u)$. We first show that it holds

$$U^o(\tilde{\mathbf{w}}_\mathbf{o}) - U^o(\mathbf{w}_\mathbf{o}) \leq 0.060$$

In order to bound the envy $U^o(\tilde{\mathbf{w}}) - U^o(\mathbf{w})$ at the NE, we will construct a weight allocation $\mathbf{m}$ that satisfies $U^o(\mathbf{m}) \leq U^o(\mathbf{w})$ and $U^o(\tilde{\mathbf{m}}) \geq U^o(\tilde{\mathbf{w}})$ – where $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{m}}$ are the allocations resulting from exchanging the resources of slices o and o' in $\mathbf{w}$ and $\mathbf{m}$, respectively. It then follows that $U^o(\tilde{\mathbf{m}}) - U^o(\mathbf{m})$ is an upper bound on the envy.

Specifically, the weight allocation $\mathbf{m}$ will be chosen such that: (i) for all slices different from o, the weights remain the same as in the NE, i.e, $\mathbf{m}^{-o} = \mathbf{w}^{-o}$; and (ii) the weights of slice o are chosen so as to maximize $U^o(\mathbf{m})$ subject to $d_b^o(\mathbf{m}^o) = \sum_{u \in \mathcal{U}_b^o} m_u \leq a_b^o(\mathbf{m}^{-o})\ \forall b \in \mathcal{B}$ and slice o's share constraint. Note that with this weight allocation we have $a_{b(u)}^o(\mathbf{m}^{-o}) = a_{b(u)}^o(\mathbf{w}^{-o})$ – for readability purposes, we will use just $a_{b(u)}^o$. Note also that the weights that slice o would have with the resources of o' remain the same, i.e. $\tilde{\mathbf{m}}^o = \tilde{\mathbf{w}}^o$.

By following a similar argument to that of Lemma 2, it can be seen that the above leads to the weights m_u for $u \in \mathcal{U}^o$ solving the set of equations below, which have a feasible solution as long as $s_o < \sum_{u \in \mathcal{U}^o} a_{b(u)}^o(\mathbf{m}^{-o})$ (we deal with the case $\sum_{b \in \mathcal{B}_o} a_b^o < s_o$ later).

$$m_u = \begin{cases} a_{b(u)}^o \dfrac{\phi_u}{\sum_{v \in \mathcal{U}_{b(u)}^o} \phi_v}, & a_{b(u)}^o = d_{b(u)}^o(\mathbf{m}^o) \\ \dfrac{\phi_u \frac{a_{b(u)}^o}{a_{b(u)}^o + d_{b(u)}^o(\mathbf{m}^o)}}{\sum\limits_{v \in \hat{\mathcal{U}}^o} \phi_v \frac{a_{b(v)}^o}{a_{b(v)}^o + d_{b(v)}^o(\mathbf{m}^o)}} s_o', & a_{b(u)}^o > d_{b(u)}^o(\mathbf{m}^o) \end{cases}$$

where $\hat{\mathcal{U}}^o$ is the set of users of slice o for which $a_{b(u)}^o > d_{b(u)}^o(\mathbf{m}^o)$ and $s_o' = s_o - \sum_{u \in \mathcal{U}^o \setminus \hat{\mathcal{U}}^o} m_u$.

It is clear that with this weight allocation we have $U^o(\mathbf{m}) \leq U^o(\mathbf{w})$. Indeed, only the weights of slice o have changed and (as mentioned before) $\mathbf{w}^o$ is the best response of the slice o, hence any other weight setting for this slice will provide a lower utility.

To show $U^o(\tilde{\mathbf{m}}) \geq U^o(\tilde{\mathbf{w}})$ we proceed as follows. The base stations that initially had a load for operator o larger than a_b^o ($d^o_{b(u)}(\mathbf{m}^o) > a_b^o$) decrease their load with the new allocation, while the others increase it. Let us denote the first set of base stations as $\mathcal{B}_1$ and the other set as $\mathcal{B}_2$. Since the base stations of set $\mathcal{B}_1$ decrease their load in the new allocation and the base stations of set $\mathcal{B}_2$ increase it, we can move from the initial allocation to the new one by iteratively selecting one base station of set $\mathcal{B}_1$ and one of set $\mathcal{B}_2$ and moving load from the first one to the second until one of them reaches its target load. When decreasing the load of base station b and increasing that of base station b' by δ we have

$$\frac{dU^o(\tilde{\mathbf{w}})}{d\delta} = -\frac{\sum\limits_{u\in\mathcal{U}^o_{b'}} \phi_u}{l_{b'}(\tilde{\mathbf{w}})} + \frac{\sum\limits_{u\in\mathcal{U}^o_{b}} \phi_u}{l_{b}(\tilde{\mathbf{w}})}$$

If we can show at the beginning (before increasing/decreasing the load of any base station), for any $b \in \mathcal{B}_1$ and $b' \in \mathcal{B}_2$ it holds

$$\frac{\sum\limits_{u\in\mathcal{U}^o_{b}} \phi_u}{l_{b}(\tilde{\mathbf{w}})} \geq \frac{\sum\limits_{u\in\mathcal{U}^o_{b'}} \phi_u}{l_{b'}(\tilde{\mathbf{w}})} \tag{4.11}$$

we will have the value of $\sum_{u\in\mathcal{U}^o_b} \frac{\phi_u}{l_b}$ for any base station of set $\mathcal{B}_1$ will always be larger than for any base station of set $\mathcal{B}_2$, since it are larger at the beginning and it increases in the intermediate steps, while it decreases for a base station of $\mathcal{B}_2$. With

this, $dU^o(\tilde{\mathbf{m}})/d\delta$ is positive at the beginning and will continue to be positive in the intermediate steps, yielding to an increase in $dU^o(\tilde{\mathbf{m}})$.

To show (4.11), we proceed as follows. It holds that

$$\frac{d_b^o(\mathbf{m}^o)}{d_{b'}^o} = \frac{\sum_{u\in\mathcal{U}_b^o}\phi_u}{\sum_{u'\in\mathcal{U}_{b'}^o}\phi_{u'}}\frac{\frac{1}{1+d_b^o(\mathbf{m}^o)/a_b^o}}{\frac{1}{1+d_{b'}^o(\mathbf{m}^o)/a_{b'}^o}} = \frac{\sum_{u\in\mathcal{U}_b^o}\phi_u}{\sum_{u'\in\mathcal{U}_{b'}^o}\phi_{u'}}\left(\frac{1+\frac{d_{b'}^o(\mathbf{m}^o)}{a_{b'}^o}}{1+\frac{d_b^o(\mathbf{m}^o)}{a_b^o}}\right)$$

For $b\in\mathcal{B}_1$ and $b'\in\mathcal{B}_2$ (since $a_b^o < d_b^o(\mathbf{m}^o)$ and $a_{b'}^o > d_{b'}^o(\mathbf{m}^o)$)

$$\frac{d_b^o(\mathbf{m}^o)}{d_{b'}^o(\mathbf{m}^o)} < \frac{\sum_{u\in\mathcal{U}_b^o}\phi_u}{\sum_{u'\in\mathcal{U}_{b'}^o}\phi_{u'}}$$

and thus

$$\frac{l_b}{\sum_{u\in\mathcal{U}_b^o}\phi_u} = \frac{a_b^o + d_b^o(\mathbf{m}^o)}{\sum_{u\in\mathcal{U}_b^o}\phi_u} < \frac{2d_b^o(\mathbf{m}^o)}{\sum_{u\in\mathcal{U}_b^o}\phi_u} < \frac{2d_{b'}^o(\mathbf{m}^o)}{\sum_{u\in\mathcal{U}_{b'}^o}\phi_u} \le \frac{a_{b'}^o + d_{b'}^o(\mathbf{m}^o)}{\sum_{u\in\mathcal{U}_{b'}^o}\phi_u} = \frac{l_{b'}}{\sum_{u\in\mathcal{U}_{b'}^o}\phi_u}$$

which proves (4.11), and thus $U^o(\tilde{\mathbf{m}}) \ge U^o(\tilde{\mathbf{w}})$.

We now go back to the case $\sum_{b\in\mathcal{B}_o} a_b^o < s_o$. Following the above procedure, in this case we can find an allocation $\mathbf{m}^o$ that satisfies: (i) $U^o(\mathbf{m}) \le U^o(\mathbf{w})$, (ii) $U^o(\tilde{\mathbf{m}}) \ge U^o(\tilde{\mathbf{w}})$ and (iii) $d_b^o(\mathbf{m}^o) \ge a_b^o\ \forall b$. In this case we then have $U^o(\tilde{\mathbf{w}}) - U^o(\mathbf{w}) \le U^o(\tilde{\mathbf{m}}) - U^o(\mathbf{m}) \le 0$.

To find an upper bound on $U^o(\tilde{\mathbf{m}}) - U^o(\mathbf{m})$, recall that

$$U^o(\tilde{\mathbf{m}}) = \sum_{u\in\mathcal{U}^o}\phi_u \log\left(\frac{\tilde{m}_u c_u}{l_b(\tilde{\mathbf{m}})}\right),$$

and

$$U^o(\mathbf{m}) = \sum_{u\in\mathcal{U}^o}\phi_u \log\left(\frac{m_u c_u}{l_b(\mathbf{m})}\right).$$

Given that $l_b(\tilde{\mathbf{m}}) = l_b(\mathbf{m})$ and $\tilde{m}_u = m_u$ for $u \notin \hat{\mathfrak{U}}^o$, this yields

$$U^o(\tilde{\mathbf{m}}) - U^o(\mathbf{m}) = \sum_{u \in \hat{\mathfrak{U}}^o} \phi_u \log(\tilde{m}_u) - \sum_{u \in \hat{\mathfrak{U}}^o} \phi_u \log(m_u).$$

Since $\sum_{u \in \hat{\mathfrak{U}}^o} \log(\tilde{m}_u)$ subject to $\sum_{u \in \hat{\mathfrak{U}}^o} \tilde{m}_u = s'_o$ takes a maximum at $\tilde{m}_u = \hat{\phi}_u s'_o$ (where $\hat{\phi}_u = \phi_u / \sum_{v \in \hat{\mathfrak{U}}^o} \phi_v$),

$$\begin{aligned} U^o(\tilde{\mathbf{m}}) - U^o(\mathbf{m}) &\leq \sum_{u \in \hat{\mathfrak{U}}^o} \phi_u \log(\hat{\phi}_u s'_o) - \sum_{u \in \hat{\mathfrak{U}}^o} \phi_u \log(m_u) \\ &\leq \sum_{u \in \hat{\mathfrak{U}}^o} \hat{\phi}_u \log(\hat{\phi}_u s'_o) - \sum_{u \in \hat{\mathfrak{U}}^o} \hat{\phi}_u \log(m_u) \end{aligned} \tag{4.12}$$

In order to bound the term $\sum_{u \in \hat{\mathfrak{U}}^o} \hat{\phi}_u \log(m_u)$ above, we look for a bound on $\frac{m_u}{m_v}$. Given that $a_b^o \geq d_b^o(\mathbf{m}^o)$ holds for all b, we have for $u, v \in \hat{\mathfrak{U}}^o$:

$$\frac{m_u}{m_v} = \frac{\phi_u}{\phi_v} \frac{\frac{a^o_{b(u)}}{a^o_{b(u)} + d^o_{b(u)}(\mathbf{m}^o)}}{\frac{a^o_{b(v)}}{a^o_{b(v)} + d^o_{b(v)}(\mathbf{m}^o)}} > \frac{\phi_u}{\phi_v} \frac{\frac{a^o_{b(u)}}{a^o_{b(u)} + a^o_{b(u)}}}{\frac{a^o_{b(v)}}{a^o_{b(v)}}} = \frac{1}{2} \frac{\hat{\phi}_u}{\hat{\phi}_v}.$$

It can be seen that $\sum_{u \in \hat{\mathfrak{U}}^o} \hat{\phi}_u \log(m_u)$ subject to $\frac{m_u}{m_v} \geq \frac{1}{2} \frac{\hat{\phi}_u}{\hat{\phi}_v}$ and $\sum_{u \in \hat{\mathfrak{U}}^o} \hat{\phi}_u = 1$ is maximized when the $\frac{m_u}{\hat{\phi}_u}$ of all users but one is equal to the lower bound given by the constraint, which yields

$$\frac{m_u}{\hat{\phi}_u} = \frac{1}{2} \frac{m_v}{\hat{\phi}_v}, \ \forall u \neq v. \tag{4.13}$$

This is shown by contradiction. Let us imagine that in the weight allocation that maximizes (4.12) there exists some other user u for which $\frac{m_u}{\hat{\phi}_u} > \frac{m_v}{2\hat{\phi}_v}$, where v is the user with the largest $m_v / \hat{\phi}_v$ of that allocation. Then, if we increase m_v by δ

and decrease m_u by δ we have

$$\frac{d}{d\delta}\sum_{u\in\mathcal{U}^o}\hat{\phi}_u\log\left(\frac{\hat{\phi}_u s'_o}{m_u}\right) = -\frac{\hat{\phi}_v}{m_v}+\frac{\hat{\phi}_u}{m_u} > 0$$

and thus (4.12) increases, which contradicts our assumption that (4.12) was already maximum. From (4.13) we have

$$m_u = \frac{\hat{\phi}_u s_o}{\sum\limits_{u'\in\mathcal{U}^o\setminus\{v\}}\hat{\phi}_u+2\hat{\phi}_v}, \text{ and } m_v = \frac{2\hat{\phi}_v s_o}{\sum\limits_{u'\in\mathcal{U}^o\setminus\{v\}}\hat{\phi}_u+2\hat{\phi}_v}$$

Combining this with (4.12) we obtain

$$\begin{aligned} U^o(\tilde{\mathbf{w}}^*_o) - U^o(\mathbf{w}^*_o) &\leq \sum_{u\in\mathcal{U}^o\setminus\{v\}}\hat{\phi}_u\log\left(\sum_{u'\in\mathcal{U}^o\setminus\{v\}}\hat{\phi}_u+2\hat{\phi}_v\right) \\ &\quad + \hat{\phi}_v\log\left(\frac{1}{2}\sum_{u'\in\mathcal{U}^o\setminus\{v\}}\hat{\phi}_u+2\hat{\phi}_v\right) \\ &= \log(1+\hat{\phi}_v)+\hat{\phi}_v\log(1/2) \end{aligned}$$

If we now compute the $\hat{\phi}_v$ that maximizes this expression we obtain $\hat{\phi}_v = \frac{1}{\log 2}-1$, and substituting this value

$$U^o(\tilde{\mathbf{w}}^*_o) - U^o(\mathbf{w}^*_o) \leq -\log(\log 2) - \left(\frac{1}{\log 2}-1\right)\log 2$$

As mentioned at the beginning, the above bounds also applies to $U^o(\tilde{\mathbf{w}}_\mathbf{o}) - U^o(\mathbf{w}_\mathbf{o})$.

We next show that the worst case envy is lower bounded by 0.041, by finding a game instance for which $U^o(\tilde{\mathbf{w}}_\mathbf{o}) - U^o(\mathbf{w}_\mathbf{o}) = 0.041$. Let us consider a scenario with 2 base stations. Let slice o have a share of s_o and one user at each base station with priorities ϕ_1 and ϕ_2. Let the loads of the other slices in these two base stations

be $a_1 = 1 - s_o - x\phi_2 s_o$ and $a_2 = x\phi_2 s_o$. for a fixed $x > 0$. Let s_o be sufficiently small such that $a_1 > \phi_2 s_o$.

In this setting, the weights of slice o at each station are given by

$$d_1^1 = \frac{s_o\phi_1 \frac{a_1}{a_1+d_1^1}}{\phi_1 \frac{a_1}{a_1+d_1^1} + \phi_2 \frac{a_2}{a_2+d_2^1}}, \text{ and } d_2^1 = \frac{s_o\phi_2 \frac{a_2}{a_2+d_2^1}}{\phi_1 \frac{a_1}{a_1+d_1^1} + \phi_2 \frac{a_2}{a_2+d_2^1}}$$

We distinguish the cases (i) $x \geq 1$ and (ii) $x < 1$.

(i) For $x \geq 1$, we consider slice o' with share $s_{o'} = s_o$ with priorities $\tilde{\phi}_1$ and $\tilde{\phi}_2$, where

$$\frac{\tilde{\phi}_1}{\tilde{\phi}_2} = \frac{\phi_1}{\phi_2} \frac{\frac{a_2-\phi_2 s_o+d_2^1}{a_2+d_2^1}}{\frac{a_1-\phi_1 s_o+d_1^1}{a_1+d_1^1}}$$

We further consider a third slice with only one user in the first base station with $s_3 = a_1 - \phi_1 s_o$ and a fourth slice with a one user in the second base station with $s_4 = a_2 - \phi_2 s_o$. This leads to $d_1^2 = \phi_1 s_o$ and $d_2^2 = \phi_2 s_o$.

If we now let $s_o \to 0$,

$$d_2^1 = \frac{\phi_2 x \phi_2 s_o}{\phi_1(x\phi_2 s_o + d_2^1) + \phi_2 x \phi_2 s_o} s_o = \phi_2 \frac{x\phi_2 s_o}{x\phi_2 s_o + d_2^1} s_o$$

From the above, $d_2^1 = \hat{x}\phi_2 s_o$, where $\hat{x}$ is the unique solution to the equation $x = (x + \hat{x})\hat{x}$. Then, $d_1^1 = s_o - \hat{x}\phi_2 s_o$. From this, we have that in this case

$$\begin{aligned} U^o(\tilde{\mathbf{w}}) - U^o(\mathbf{w}) &= \phi_1 \log\left(\frac{\phi_1 s_o}{s_o - \hat{x}\phi_2 s_o}\right) + \phi_2 \log\left(\frac{\phi_2 s_o}{\hat{x}\phi_2 s_o}\right) \\ &= \phi_1 \log\left(\frac{\phi_1}{1 - \hat{x} + \hat{x}\phi_1}\right) - (1 - \phi_1)\log(\hat{x}) \end{aligned}$$

(ii) In case that $x < 1$, we consider slice o' has priorities $\tilde{\phi}_1$ and $\tilde{\phi}_2$, where

$$\frac{\tilde{\phi}_1}{\tilde{\phi}_2} = \frac{\phi_1}{\phi_2} \frac{\frac{a_2-x\phi_2 s_o+w_2}{a_2+w_2}}{\frac{a_1-s_o-x\phi_2 s_o+w_1}{a_1+w_1}}$$

which leads to $\tilde{w}_1 = (1 - x\phi_2)s_o$ and $\tilde{w}_2 = x\phi_2 s_o$. We further consider a third slice in the first base station with $s_3 = a_1 - (1 - x\phi_2)s_o$. If we now let $s_o \to 0$, we have the same expressions as above for w_1 and w_2, from which

$$\begin{aligned} U^o(\tilde{\mathbf{w}}_o) - U^o(\mathbf{w}_o) &= \phi_1 \log\left(\frac{s_o - x\phi_2 s_o}{s_o - \hat{x}\phi_2 s_o}\right) + \phi_2 \log\left(\frac{x\phi_2 s_o}{\hat{x}\phi_2 s_o}\right) \\ &= \phi_1 \log\left(\frac{1 - x + x\phi_1}{1 - \hat{x} + \hat{x}\phi_1}\right) - (1 - \phi_1)\log\left(\frac{x}{\hat{x}}\right) \end{aligned}$$

By putting together the cases $x \geq 1$ and $x < 1$, we can obtain a lower bound for the worst-case envy by finding the values of x and ϕ_1 over $x > 0$ and $\phi_1 \in [0, 1]$ that minimize the following expression

$$\begin{cases} \phi_1 \log\left(\frac{\phi_1}{1-\hat{x}+\hat{x}\phi_1}\right) - (1 - \phi_1)\log(\hat{x}), & \text{for } x \geq 1 \\ \phi_1 \log\left(\frac{1-x+x\phi_1}{1-\hat{x}+\hat{x}\phi_1}\right) - (1 - \phi_1)\log\left(\frac{x}{\hat{x}}\right), & \text{for } x < 1 \end{cases}$$

Performing the above search numerically, we find a scenario with the following envy level:

$$U^o(\tilde{\mathbf{w}}_o) - U^o(\mathbf{w}_o) = 0.041$$

which terminates the proof of the theorem. □

Lemma 5. *For any NE weight allocation of game $G^{(\varepsilon)}$, there exists a constant $\delta > 0$ such that $w_u^\varepsilon \geq \delta \; \forall u \in \mathcal{U}$ and $\varepsilon < \varepsilon_{max}$.*

Proof. According to Lemma 3, the best response of a user $u \in \mathcal{U}^o$ in the ε perturbed

game is given by

$$w_u = \frac{\frac{(\varepsilon + a^o_{b(u)})^{\frac{1}{\alpha_o}}}{(\varepsilon + a^o_{b(u)} + d^o_{b(u)})^{\frac{2}{\alpha_o}-1}}}{\sum\limits_{v \in \mathcal{U}^o} \left(\frac{\beta_v}{\beta_u}\right) \frac{(\varepsilon + a^o_{b(v)})^{\frac{1}{\alpha_o}}}{(\varepsilon + a^o_{b(v)} + d^o_{b(v)})^{\frac{2}{\alpha_o}-1}}} s_o. \tag{4.14}$$

In order to derive a bound for w_u, we proceed along the following steps. First, we obtain a bound for w_u as a function of l_b. Second, we derive a bound for l_b. Finally, by combining the results of the first and the second steps, we obtain a bound for w_u.

Bound for w_u as a function of l_b

We will first prove the existence of a bound for the case of $\alpha_o \geq 1$ and then for the case $\alpha_o < 1$. Let us start with the case $\alpha_o \geq 1$. From Eq. (4.14) it follows

$$w_u \geq \frac{\frac{(\varepsilon + a^o_{b(u)})^{\frac{1}{\alpha_o}}}{(\varepsilon + a^o_{b(u)} + d^o_{b(u)})^{\frac{2}{\alpha_o}-1}}}{\max\limits_{v \in \mathcal{U}^o} \left[\frac{\beta_v}{\beta_u}\right] \sum\limits_{v \in \mathcal{U}^o} \frac{(\varepsilon + a^o_{b(v)})^{\frac{1}{\alpha_o}}}{(\varepsilon + a^o_{b(v)} + d^o_{b(v)})^{\frac{2}{\alpha_o}-1}}} s_o$$

Let us define the constant $m_u^o = \max_{v \in \mathcal{U}^o} \left[\frac{\beta_v}{\beta_u} \right]$. Then,

$$
\begin{aligned}
w_u &\geq \frac{\frac{(\varepsilon + l_{b(u)} - d^o_{b(u)})^{\frac{1}{\alpha_o}}}{(\varepsilon + l_{b(u)})^{\frac{2}{\alpha_o}-1}}}{m_u^o \sum\limits_{v \in \mathcal{U}^o} \frac{(\varepsilon + l_{b(v)} - d^o_{b(v)})^{\frac{1}{\alpha_o}}}{(\varepsilon + l_{b(v)})^{\frac{2}{\alpha_o}-1}}} s_o \geq \frac{\frac{(l_{b(u)} - d^o_{b(u)})^{\frac{1}{\alpha_o}}}{(l_{b(u)})^{\frac{2}{\alpha_o}-1}}}{m_u^o \sum\limits_{v \in \mathcal{U}^o} (\varepsilon + l_{b(v)})^{-\frac{1}{\alpha_o}+1}} s_o \\
&\geq \frac{\frac{(l_{b(u)} - d^o_{b(u)})^{\frac{1}{\alpha_o}}}{(l_{b(u)})^{\frac{2}{\alpha_o}-1}}}{m_u^o \sum\limits_{v \in \mathcal{U}^o} \left((\varepsilon)^{-\frac{1}{\alpha_o}+1} + (l_{b(v)})^{-\frac{1}{\alpha_o}+1} \right)} s_o \geq \frac{\frac{(l_{b(u)} - d^o_{b(u)})^{\frac{1}{\alpha_o}}}{(l_{b(u)})^{\frac{2}{\alpha_o}-1}}}{m_u^o \sum\limits_{v \in \mathcal{U}^o} \left(1 + (\varepsilon)^{-\frac{1}{\alpha_o}+1} \right)} s_o \\
&\geq \frac{s_o}{2 m_u^o |\mathcal{U}^o|} \frac{(l_{b(u)} - d^o_{b(u)})^{\frac{1}{\alpha_o}}}{(l_{b(u)})^{\frac{2}{\alpha_o}-1}}
\end{aligned}
$$

We next focus on the case $\alpha_o < 1$. From (4.14) it follows that

$$
\frac{(w_u)^{\alpha_o}}{(w_v)^{\alpha_o}} = \frac{\frac{(\varepsilon + l_{b(u)} - d^o_{b(u)})}{(\varepsilon + l_{b(u)})^{2-\alpha_o}}}{\frac{(\varepsilon + l_{b(v)} - d^o_{b(v)})}{(\varepsilon + l_{b(v)})^{2-\alpha_o}}} \geq \frac{\frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}}}{\frac{(\varepsilon + l_{b(v)} - d^o_{b(v)})}{(\varepsilon + l_{b(v)})^{2-\alpha_o}}}
$$

From the above,

$$
\begin{aligned}
(w_u)^{\alpha_o} &\geq \frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}} \frac{(\varepsilon + l_{b(v)})^{2-\alpha_o}}{(\varepsilon + l_{b(v)} - d^o_{b(v)})} (w_v)^{\alpha_o} \\
&\geq \frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}} (\varepsilon + l_{b(v)})^{1-\alpha_o} (w_v)^{\alpha_o} \\
&\geq \frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}} w_v \left(\frac{w_v}{l_{b(u)}} \right)^{1-\alpha_o} \\
&\geq \frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}} w_v
\end{aligned}
$$

Given that the sum of the weights of all the users of slice o is equal to s_o, there needs to be at least one user v for which $\sum_{u' \in \mathcal{U}^o_{b(v)}} w_{u'} = d_v w_v \geq \frac{s_o}{|\mathcal{U}^o|}$. If we take such a user v, from the above we obtain

$$
(w_u)^{\alpha_o} \geq \frac{(l_{b(u)} - d^o_{b(u)})}{(l_{b(u)})^{2-\alpha_o}} \frac{s_o}{|\mathcal{U}^o|}, \ \forall u \in \mathcal{U}_o
$$

Isolating w_u from the above yields

$$w_u \geq \left(\frac{s_o}{|\mathcal{U}_o| m_u^o}\right)^{\frac{1}{\alpha_o}} \frac{\left(l_{b(u)} - d_{b(u)}^o\right)^{\frac{1}{\alpha_o}}}{\left(l_{b(u)}\right)^{\frac{2}{\alpha_o}-1}} \geq \frac{s_o}{2m_u^o|\mathcal{U}^o|} \frac{\left(l_{b(u)} - d_u w_u\right)^{\frac{1}{\alpha_o}}}{\left(l_{b(u)}\right)^{\frac{2}{\alpha_o}-1}}$$

Putting together the bounds obtained for $\alpha_o \geq 1$ and $\alpha_o < 1$ leads to the following expression which holds for any $\alpha_o > 0$:

$$w_u \geq \frac{s_o}{2m_u^o|\mathcal{U}^o|} \frac{\left(l_{b(u)} - d_u w_u\right)^{\frac{1}{\alpha_o}}}{\left(l_{b(u)}\right)^{\frac{2}{\alpha_o}-1}}, \text{ for } \alpha_o > 0 \tag{4.15}$$

Bound for l_b

We will now provide a bound for l_b. Let us define $\mathcal{U}_b^*$ as a subset of $\mathcal{U}$ that contains a representative user of each slice at base station b. Let us further define sets $\mathcal{U}_b^{(*,\geq 1)}$ and $\mathcal{U}_b^{(*,<1)}$ as the subsets of $\mathcal{U}_b^*$ corresponding to the slices with $\alpha_o \geq 1$ and $\alpha_o < 1$, respectively. Note that

$$l_{b(u)} = \sum_{u \in \mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u = \sum_{u \in \mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u + \sum_{u \in \mathcal{U}_{b(u)}^{(*,<1)}} d_u w_u.$$

Let us denote the largest α_o of the slices with $\alpha_o \geq 1$ by $\overline{\alpha_o}$ and and the smallest α_o of these slices by $\underline{\alpha_o}$. Similarly, let $\overline{\alpha_o'}$ denote the largest α_o of the slices with $\alpha_o < 1$ and $\underline{\alpha_o'}$ the smallest. Let us further define $\eta = \min_{o \in \mathcal{O}, u \in \mathcal{U}_{b(u)}^{(*,\geq 1)}} \frac{s_o d_u}{2m_u^o|\mathcal{U}|}$.

Then, using the bound for w_u in (4.15), we obtain

$$\begin{aligned}
\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u &\geq \eta \sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} \frac{(l_{b(u)} - d_u w_u)^{\frac{1}{\alpha_o(u)}}}{(l_{b(u)})^{\frac{2}{\alpha_o(u)}-1}} \\
&\geq \eta (l_{b(u)})^{1-\frac{1}{\overline{\alpha_o}}} \sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} (1 - \frac{d_u w_u}{l_{b(u)}})^{\frac{1}{\alpha_o}} \\
&\geq \eta (l_{b(u)})^{1-\frac{1}{\overline{\alpha_o}}} \left[\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} 1 - \sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} \frac{d_u w_u}{l_{b(u)}}\right]^{\frac{1}{\alpha_o}} \\
&\geq \eta (l_{b(u)})^{1-\frac{1}{\overline{\alpha_o}}} \left[\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} 1 - \sum_{u\in\mathcal{U}_{b(u)}^{*}} \frac{d_u w_u}{l_{b(u)}}\right]^{\frac{1}{\alpha_o}} \\
&\geq \eta (\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u)^{1-\frac{1}{\overline{\alpha_o}}} \left[(|\mathcal{U}_{b(u)}^{(*,\geq 1)}| - 1)\right]^{\frac{1}{\alpha_o}}
\end{aligned}$$

where the third inequality holds from concavity.

Isolating $\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u$ from the above yields

$$\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u \geq (\eta)^{\overline{\alpha_o}} (|\mathcal{U}_{b(u)}^{(*,\geq 1)}| - 1)^{\frac{\overline{\alpha_o}}{\alpha_o}} \tag{4.16}$$

Applying the same reasoning to set $\mathcal{U}_{b(u)}^{(*,<1)}$, we obtain

$$\sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u \geq (\eta)^{\overline{\alpha'_o}} (|\mathcal{U}_{b(u)}^{(*,\leq 1)}| - 1)^{\frac{\overline{\alpha'_o}}{\alpha'_o}} \tag{4.17}$$

Putting together (4.16) and (4.17) yields

$$\begin{aligned}
l_{b(u)} &= \sum_{u\in\mathcal{U}_{b(u)}^{(*,\geq 1)}} d_u w_u + \sum_{u\in\mathcal{U}_{b(u)}^{(*,<1)}} d_u w_u \\
&\geq (\eta)^{\overline{\alpha_o}} (|\mathcal{U}_{b(u)}^{(*,\geq 1)}| - 1)^{\frac{\overline{\alpha_o}}{\alpha_o}} + (\eta)^{\overline{\alpha'_o}} (|\mathcal{U}_{b(u)}^{(*,\leq 1)}| - 1)^{\frac{\overline{\alpha'_o}}{\alpha'_o}}
\end{aligned}$$

The above gives a lower bound for l_b as long there are at least two slices with $\alpha_o \geq 1$ or $\alpha_o < 1$ in base station b. We next deal with the case in which there are two slices in base station b, o and o', one with $\alpha_o < 1$ and the other with $\alpha_{o'} \geq 1$. It follows from the previous analysis that for this case we have

$$w_u \geq \frac{s_o}{2m_u^o|\mathcal{U}^o|} \frac{(d_v w_v)^{\frac{1}{\alpha_o}}}{(l_b)^{\frac{2}{\alpha_o}-1}} \tag{4.18}$$

and

$$w_v \geq \frac{s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|} \frac{(d_u w_u)^{\frac{1}{\alpha_{o'}}}}{(l_b)^{\frac{2}{\alpha_{o'}}-1}} \tag{4.19}$$

Combining (4.18) and (4.19) we obtain

$$w_u^{\alpha_o} \geq \left(\frac{s_o}{2m_u^o|\mathcal{U}^o|}\right)^{\alpha_o} \frac{d_v}{(l_b)^{2-\alpha_o}} \frac{s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|} \frac{(d_u w_u)^{\frac{1}{\alpha_{o'}}}}{(l_b)^{\frac{2}{\alpha_{o'}}-1}}$$

from which

$$l_b^{1-\alpha_o+2/\alpha_{o'}} (w_u)^{\alpha_o - \frac{1}{\alpha_{o'}}} \geq \left(\frac{s_o}{2m_u^o|\mathcal{U}^o|}\right)^{\alpha_o} \frac{d_v (d_u)^{\frac{1}{\alpha_{o'}}} s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|}$$

and thus

$$l_b^{1+1/\alpha_{o'}} \left(\frac{w_u}{l_b}\right)^{\alpha_o - 1/\alpha_{o'}} \geq \left(\frac{s_o}{2m_u^o|\mathcal{U}^o|}\right)^{\alpha_o} \frac{d_v (d_u)^{\frac{1}{\alpha_{o'}}} s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|}.$$

Multiplying each side by $d_u^{\alpha_o - 1/\alpha_{o'}}$ we obtain

$$l_b^{1+1/\alpha_{o'}} \left(\frac{d_u w_u}{l_b}\right)^{\alpha_o - 1/\alpha_{o'}} \geq \left(\frac{s_o}{2m_u^o|\mathcal{U}^o|}\right)^{\alpha_o} \frac{d_v (d_u)^{\alpha_o} s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|}$$

from which

$$l_b \left(\frac{d_{b(u)}^o}{l_b}\right)^{\alpha_o - 1/\alpha_{o'}} \geq \left(\frac{s_o}{2m_u^o|\mathcal{U}^o|}\right)^{\alpha_o} \frac{d_v (d_u)^{\alpha_o} s_{o'}}{2m_v^{o'}|\mathcal{U}^{o'}|} \doteq \mu_1 \tag{4.20}$$

By following the same reasoning as above but isolating w_v instead of w_u from (4.18) and (4.19), we obtain

$$l_b \left(\frac{d^o_{b(v)}}{l_b} \right)^{\alpha_{o'} - 1/\alpha_o} \geq \left(\frac{s_{o'}}{2m^{o'}_v |\mathcal{U}^{o'}|} \right)^{\alpha_{o'}} \frac{s_o d_u (d_v)^{\alpha_o}}{2m^o_u |\mathcal{U}^o|} \doteq \mu_2 \tag{4.21}$$

If $\alpha_o - 1/\alpha_{o'} > 0$, it holds $\left(\frac{d^o_{b(u)}}{l_b} \right)^{\alpha_o - 1/\alpha_{o'}} \leq 1$, and then from (4.20) we obtain

$$l_b \geq \left(\frac{s_o}{2m^o_u |\mathcal{U}^o|} \right)^{\alpha_o} \frac{s_{o'} d_v (d_u)^{\alpha_o}}{2m^{o'}_v |\mathcal{U}^{o'}|}$$

Otherwise (i.e., in the case $\alpha_o - 1/\alpha_{o'} < 0$), by taking the minimum of both sides of (4.20) and (4.21), we obtain the following inequality

$$l_b \min \left(\left(\frac{d^o_{b(u)}}{l_b} \right)^{\alpha_o - \frac{1}{\alpha_{o'}}}, \left(\frac{d^o_{b(v)}}{l_b} \right)^{\alpha_{o'} - \frac{1}{\alpha_o}} \right) \geq \min (\mu_1, \mu_2)$$

From the above,

$$l_b \min \left(\frac{1}{\left(\frac{d^o_{b(u)}}{l_b} \right)^{\hat{\alpha_o}}}, \frac{1}{\left(\frac{d^o_{b(v)}}{l_b} \right)^{\hat{\alpha_o}}} \right) \geq \min (\mu_1, \mu_2)$$

where $\hat{\alpha_o} = \max(-\alpha_o + \frac{1}{\alpha_{o'}}, -\alpha_{o'} + \frac{1}{\alpha_o})$. The above is equivalent to

$$l_b \frac{1}{\left(\max \left(\frac{d^o_{b(u)}}{l_b}, \frac{d^o_{b(v)}}{l_b} \right) \right)^{\hat{\alpha_o}}} \geq \min (\mu_1, \mu_2)$$

Noting that either $\frac{d^o_{b(u)}}{l_b} \geq \frac{1}{2}$ or $\frac{d^o_{b(v)}}{l_b} \geq \frac{1}{2}$,

$$l_b \geq \left(\frac{1}{2} \right)^{\hat{\alpha_o}} \min (\mu_1, \mu_2) \doteq \underline{l_b} > 0$$

The above proves that l_b is bounded in the perturbed game. Hereafter, we denote this bound by $\underline{l_b} > 0$.

Bound for w_u

Let us start with the case $\alpha_o \geq 1$. Combining (4.15) with the bound for l_b gives

$$w_u \geq \frac{s_o}{2m_u^o|\mathcal{U}^o|(\underline{l_b})^{\frac{2}{\alpha_o}-1}}(\underline{l_b} - d_u w_u)^{\frac{1}{\alpha_o}} \geq \frac{s_o}{2m_u^o|\mathcal{U}^o|(\underline{l_b})^{\frac{2}{\alpha_o}-1}}\left[(\underline{l_b})^{\frac{1}{\alpha_o}} - (d_u w_u)^{\frac{1}{\alpha_o}}\right]$$

Since $w_u^{\frac{1}{\alpha_o}} \geq w_u$, it follows that

$$w_u^{\frac{1}{\alpha_o}} + \frac{s_o(d_u w_u)^{\frac{1}{\alpha_o}}}{m_u^o|\mathcal{U}^o|(\underline{l_b})^{\frac{2}{\alpha_o}-1}} \geq \frac{s_o(\underline{l_b})^{\frac{1}{\alpha_o}}}{m_u^o|\mathcal{U}^o|(\underline{l_b})^{\frac{2}{\alpha_o}-1}}$$

from which

$$w_u \geq \left(\frac{s_o(\underline{l_{b(u)}})^{\frac{1}{\alpha_o}}}{m_u^o|\mathcal{U}^o|(\underline{l_{b(u)}})^{\frac{2}{\alpha_o}-1} + s_o(d_u)^{\frac{1}{\alpha_o}}}\right)^{\alpha_o}$$

which provides a lower bound on w_u for this case. We next look at the case $\alpha_o < 1$. Combining again (4.15) with the bound for $\underline{l_b}$ gives

$$(w_u)^{\alpha_o} \geq \left(\frac{s_o}{2|\mathcal{U}_o|m_u^o}\right)(\underline{l_b} - d_u w_u)$$

which yields

$$(w_u)^{\alpha_o} + (w_u)^{\alpha_o}\frac{s_o d_u}{2|\mathcal{U}_o|m_u^o} \geq (w_u)^{\alpha_o} + w_u\frac{s_o d_u}{2|\mathcal{U}_o|m_u^o} \geq \frac{s_o l_{b(u)}}{|\mathcal{U}_o|m_u^o} \geq \frac{s_o \underline{l_b}}{2|\mathcal{U}_o|m_u^o}$$

Isolating w_u from the above equation, we obtain the following lower bound for this case:

$$w_u \geq \left(\frac{s_o \underline{l_b}}{(2|\mathcal{U}_o|m_u^o + s_o d_u)}\right)^{\frac{1}{\alpha_o}} \doteq \delta$$

With the above, we have obtained a lower bound on w_u for all possible cases. Hereafter we denote this lower bound by δ. □

Lemma 6. *Let us consider the game with sequential updates. If the game has not converged to a Nash equilibrium, i.e.* $\mathbf{\Delta\omega}(r) \neq \mathbf{0}$ *for* $r > 1$*, then:*

$$\max_{o\in\mathcal{O}}\left(1+\overline{\Delta}\omega^o(r+1),\frac{1}{1+\underline{\Delta}\omega^o(r+1)}\right) < \max_{o\in\mathcal{O}}\left(1+\overline{\Delta}\omega^o(r),\frac{1}{1+\underline{\Delta}\omega^o(r)}\right). \tag{4.22}$$

Proof. We consider the following two cases: (i) $\mathbf{\Delta\omega}(r+1) = 0$ and (ii) $\mathbf{\Delta\omega}(r+1) \neq 0$.

Case 1: $\mathbf{\Delta\omega}(r+1) = \mathbf{0}$**.**

From $\mathbf{\Delta\omega}(r+1) = \mathbf{0}$, we have that the lhs of (4.22) is equal to 1. Furthermore, from $\mathbf{\Delta\omega}(r) \neq \mathbf{0}$ it follows that the rhs must be strictly greater than 1. The inequality for this case follows from these two results.

Case 2: $\mathbf{\Delta\omega}(r+1) \neq \mathbf{0}$**.**

Let us define $\mathbf{\Delta a}^o(t) = (\Delta a_b^o(t) \colon b \in \mathcal{B})$ such that

$$a_b^o(t) = a_b^o(t - |\mathcal{O}| + 1)(1 + \Delta a_b^o(t)).$$

Note that, if operator o updates its best response at time t, then $a_b^o(t)$ is the congestion that operator o sees upon making its update, and $\Delta a_b^o(t)$ corresponds to the congestion variation relative to its previous update slot.

In order to prove (4.22) we will begin by showing that for any slice o updat-

ing its weights in round $r+1$, say at time $t+1$, the following holds:

$$\max\left(1+\overline{\Delta}w^o(t+1), \frac{1}{1+\underline{\Delta}w^o(t+1)}\right) \tag{4.23}$$
$$< \max_{t'\in\{t-|\mathcal{O}|+1,\ldots,t\}}\left(1+\overline{\Delta}w^{o(t')}(t'), \frac{1}{1+\underline{\Delta}w^{o(t')}(t')}\right)$$

where $o(t')$ denotes the slice updating its weights at time t'. The above means that the "relative change" in weights for slice o is strictly smaller than that seen by the other slices in the previous $|\mathcal{O}|$ updates.

To prove (4.23), we will consider two additional subcases: (i) $\mathbf{\Delta a}^o(t)=0$, and (i) $\mathbf{\Delta a}^o(t)\neq 0$.

Subcase 2.1: $\mathbf{\Delta a}^o(t)=\mathbf{0}$.

To prove (4.23) for this case, we will show that the lhs of (4.23) is equal to 1 and the rhs is strictly greater than 1. To show that the lhs is equal to 1, we proceed as follows. Note that, as $\mathbf{\Delta a}^o(t)=\mathbf{0}$, the congestion seen by slice o at time $t+1$ is unchanged with respect to its previous update, and therefore $\mathbf{\Delta w}^o(t+1)=\mathbf{0}$. As a result of this, the lhs of (4.23) is equal to 1.

The proof that the rhs of (4.23) is strictly greater than 1 follows by contradiction. Suppose that the rhs is equal to 1. This implies that $\mathbf{\Delta w}^{o(t')}(t')=\mathbf{0}$ for $t'\in\{t-|\mathcal{O}|+1,...,t\}$, where $o(t')$ denotes the slice that changes its weights at time t'. This yields $\mathbf{\Delta a}^{o(t+2)}(t+1)=\mathbf{0}$, since there have not been any changes in the weights since its last update, which in turn leads to $\mathbf{\Delta w}^{o(t+2)}(t+2)=0$. Applying this argument recursively, it can be shown that $\mathbf{\Delta w}^{o(t')}(t')=0$ for any $t''>t+1$. This in turn implies that there is now weight change in round

$r+1$, i.e., $\mathbf{\Delta\omega}(r+1) = \mathbf{0}$, which contradicts the fact that we are looking at case $\mathbf{\Delta\omega}(r+1) \neq \mathbf{0}$.

Subcase 2.2: $\mathbf{\Delta a}^o(t) \neq \mathbf{0}$.

In order to show (4.23) for this case, we will start by proving the following intermediate result

$$\max\left(1+\overline{\Delta}w^o(t+1), \frac{1}{1+\underline{\Delta}w^o(t+1)}\right) < \max\left(1+\overline{\Delta}a^o(t), \frac{1}{1+\underline{\Delta}a^o(t)}\right), \tag{4.24}$$

where $\overline{\Delta}a^o(t) = \max_{b\in\mathcal{B}} \Delta a^o(t)$ and $\underline{\Delta}a^o(t) = \min_{b\in\mathcal{B}} \Delta a^o(t)$.[10]

If $\mathbf{\Delta w}^o(t+1) = 0$, (4.24) follows from the fact that the lhs of (4.24) is equal to 1 and the rhs is strictly greater than 1 (given that $\mathbf{\Delta a}^o(t) \neq \mathbf{0}$).

We next consider the case where $\mathbf{\Delta w}^o(t+1) \neq 0$. For this case, we prove (4.24) by showing that the each of two terms of the lhs of (4.24) is strictly lower than the rhs. First, we show this for the first term, i.e.,

$$\left(1+\overline{\Delta}w^o(t+1)\right) < \max\left(1+\overline{\Delta}a^o(t), \frac{1}{1+\underline{\Delta}a^o(t)}\right). \tag{4.25}$$

We prove (4.25) by contradiction: we suppose that

$$1+\overline{\Delta}w^o(t+1) \geq \max\left(1+\overline{\Delta}a^o(t), \frac{1}{1+\underline{\Delta}a^o(t)}\right). \tag{4.26}$$

and show that this yields a contradiction.

[10]Note that we are defining $\overline{\Delta}a^o(t)$ and $\underline{\Delta}a^o(t)$ across all base stations, and not only those where slice o has users; for the base stations where slice o does not have users, $a^o(t)$ will simply be the load of the base station.

Let $u, v \in \mathcal{U}^o$ be, respectively, the user for which $\Delta w_u(t+1)$ takes the largest value and the one for which it takes the smallest value. Then,

$$\frac{w_u(t+1)}{w_v(t+1)} = \frac{w_u(t)(1+\overline{\Delta}w^o(t+1))}{w_v(t)(1+\underline{\Delta}w^o(t+1))} \tag{4.27}$$

where $\overline{\Delta}w^o(t+1) = \Delta w_u(t+1)$ and $\underline{\Delta}w^o(t+1) = \Delta w_v(t+1)$

For readability purposes, let us denote the base station of user u by b and the base station of user v by b'. From (4.2), we have

$$\begin{aligned}\frac{w_u(t+1)}{w_v(t+1)} &= \frac{\frac{\beta_u\left(a_b^o(t-|\mathcal{O}|+1)(1+\Delta a_b^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)(1+\Delta a_b^o(t))+d_u w_u(t)(1+\Delta w_u(t+1)\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\Delta a_{b'}^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\Delta a_{b'}^o(t))+d_v w_v(t)(1+\Delta w_v(t+1)\right)^{\frac{2}{\alpha_o}-1}}} \\ &= \frac{\frac{\beta_u\left(a_b^o(t-|\mathcal{O}|+1)(1+\Delta a_b^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)(1+\Delta a_b^o(t))+d_u w_u(t)(1+\overline{\Delta}w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\Delta a_{b'}^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\Delta a_{b'}^o(t))+d_v w_v(t)(1+\underline{\Delta}w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}}\end{aligned}$$

By taking the largest and smallest values in $\mathbf{\Delta a}^o(t)$, we obtain the following inequality:

$$\begin{aligned}\frac{w_u(t+1)}{w_v(t+1)} &\leq \frac{\beta_u}{\beta_v}\frac{\frac{\left(a_b^o(t-|\mathcal{O}|+1)(1+\overline{\Delta}a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)(1+\overline{\Delta}a^o(t))+d_u w_u(t)(1+\overline{\Delta}w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta}a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta}a^o(t))+d_v w_v(t)(1+\underline{\Delta}w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}} \\ &= \frac{\beta_u}{\beta_v}\frac{\frac{\left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)\frac{(1+\overline{\Delta}a^o(t))}{(1+\overline{\Delta}a^o(t))^{\frac{1}{2-\alpha_o}}}+d_u w_u(t)\frac{(1+\overline{\Delta}w^o(t+1))}{(1+\overline{\Delta}a^o(t))^{\frac{1}{2-\alpha_o}}}\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta}a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta}a^o(t))+d_v w_v(t)(1+\underline{\Delta}w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}}\end{aligned}$$

From our assumption, we have that $1 + \overline{\Delta} w^o(t+1) \geq 1 + \overline{\Delta} a^o(t)$, from which it follows that:

$$
\begin{aligned}
\frac{w_u(t+1)}{w_v(t+1)} &\leq \frac{\frac{\beta_u \left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)+d_u w_u(t)\right)^{\frac{2}{\alpha_o}-1} \left(\frac{(1+\overline{\Delta} a^o(t))}{(1+\overline{\Delta} a^o(t))^{\frac{1}{2-\alpha_o}}}\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v \left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta} a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta} a^o(t))+d_v w_v(t)(1+\underline{\Delta} w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}} \\
&\leq \frac{\frac{\beta_u \left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}} (1+\overline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)+d_u w_u(t)\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v \left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta} a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta} a^o(t))+d_v w_v(t)(1+\underline{\Delta} w^o(t+1))\right)^{\frac{2}{\alpha_o}-1}}} \\
&< \frac{\frac{\beta_u \left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}} (1+\overline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)+d_u w_u(t)\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v \left(a_{b'}^o(t-|\mathcal{O}|+1)(1+\underline{\Delta} a^o(t))\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)+d_v w_v(t)\right)^{\frac{2}{\alpha_o}-1}}} \\
&= \frac{\beta_u \left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1)+d_u w_u(t)\right)^{\frac{2}{\alpha_o}-1}} \frac{(1+\overline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}{(1+\underline{\Delta} a^o(t))^{\frac{1}{\alpha_o}}} \Big/ \frac{\beta_v \left(a_{b'}^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1)+d_v w_v(t)\right)^{\frac{2}{\alpha_o}-1}} \\
&= \frac{w_u(t)}{w_v(t)} \frac{(1+\overline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}{(1+\underline{\Delta} a^o(t))^{\frac{1}{\alpha_o}}}.
\end{aligned}
$$

where the third step holds since $\underline{\Delta} a^o(t) < 0$ and $\underline{\Delta} w^o(t+1) < 0$, and the last one follows from the fact that $w_u(t) = w_u(t - |\mathcal{O}| + 1)$ (as slice o does not updated its weights in the time interval $\{t - |\mathcal{O}| + 2, ..., t\}$). Combining (4.27) with the above yields

$$
\frac{w_u(t)(1+\overline{\Delta} w^o(t+1))}{w_v(t)(1+\underline{\Delta} w^o(t+1))} < \frac{w_u(t)}{w_v(t)} \frac{(1+\overline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}{(1+\underline{\Delta} a^o(t))^{\frac{1}{\alpha_o}}}.
$$

From the fact that $x^a y^b \leq \max(x, y)$ for $x, y \geq 1$ and $a + b = 1$ we have

$$\frac{(1 + \overline{\Delta} a^o(t+1))^{1-\frac{1}{\alpha_o}}}{(1 + \underline{\Delta} a^o(t))^{\frac{1}{\alpha_o}}} \leq \max\left(1 + \overline{\Delta} a^o(t), \frac{1}{1 + \underline{\Delta} a^o(t)}\right).$$

From the above two equations we have

$$\frac{1 + \overline{\Delta} w^o(t+1)}{1 + \underline{\Delta} w^o(t+1)} < \max\left(1 + \overline{\Delta} a^o(t), \frac{1}{1 + \underline{\Delta} a^o(t)}\right).$$

and combining this with (4.26) yields

$$\frac{1}{1 + \underline{\Delta} w^o(t+1)} < 1$$

which contradicts $\frac{1}{1+\underline{\Delta} w^o(t+1)} > 1$, thus proving (4.25).

The next step to prove (4.24) is to show that the second term of the lhs of (4.24) is strictly lower than the rhs, i.e.,

$$\frac{1}{1 + \underline{\Delta} w^o(t+1)} < \max\left(1 + \overline{\Delta} a^o(t), \frac{1}{1 + \underline{\Delta} a^o(t)}\right). \tag{4.28}$$

The proof is analogous to the one for (4.25) and proceeds again by contradiction: we suppose that

$$\frac{1}{1 + \underline{\Delta} w^o(t+1)} \geq \max\left(1 + \overline{\Delta} a^o(t), \frac{1}{1 + \underline{\Delta} a^o(t)}\right), \tag{4.29}$$

and show that this yields a contradiction.

Employing a similar argument to the one above, we have:

$$\begin{aligned}\frac{w_u(t+1)}{w_v(t+1)} &< \frac{\frac{\beta_u \left(a_b^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_b^o(t-|\mathcal{O}|+1) + d_u w_u(t)\right)^{\frac{2}{\alpha_o}-1}}}{\frac{\beta_v \left(a_{b'}^o(t-|\mathcal{O}|+1)\right)^{\frac{1}{\alpha_o}}}{\left(a_{b'}^o(t-|\mathcal{O}|+1) + d_v w_v(t)\right)^{\frac{2}{\alpha_o}-1}}} \frac{(1 + \overline{\Delta} a^o(t+1))^{\frac{1}{\alpha_o}}}{(1 + \underline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}} \\ &= \frac{w_u(t)}{w_v(t)} \frac{(1 + \overline{\Delta} a^o(t))^{\frac{1}{\alpha_o}}}{(1 + \underline{\Delta} a^o(t))^{1-\frac{1}{\alpha_o}}}.\end{aligned}$$

Combining (4.27) with the above yields

$$\frac{w_u(t)(1+\overline{\Delta}w^o(t+1))}{w_v(t)(1+\underline{\Delta}w^o(t+1))} < \frac{w_u(t)}{w_v(t)}\frac{(1+\overline{\Delta}a^o(t))^{\frac{1}{\alpha_o}}}{(1+\underline{\Delta}a^o(t))^{1-\frac{1}{\alpha_o}}}.$$

Again from $x^a y^b \leq \max(x,y)$ for $x, y \geq 1$ and $a+b=1$ we have

$$\frac{(1+\overline{\Delta}a^o(t))^{\frac{1}{\alpha_o}}}{(1+\underline{\Delta}a^o(t))^{1-\frac{1}{\alpha_o}}} \leq \max\left(1+\overline{\Delta}a^o(t), \frac{1}{1+\underline{\Delta}a^o(t)}\right).$$

From the above two equations

$$\frac{1+\overline{\Delta}w^o(t+1)}{1+\underline{\Delta}w^o(t+1)} < \max\left(1+\overline{\Delta}a^o(t), \frac{1}{1+\underline{\Delta}a^o(t)}\right).$$

Combining the above with (4.29) leads to $(1+\overline{\Delta}w^o(t+1)) > 1$, which contradicts the fact that $\overline{\Delta}w^o(t+1)$ is necessarily strictly greater than 1.

The above proves our intermediate step (4.24). Next, building on this intermediate result, we shall show that (4.23) holds.

Recall that $o(t')$ denotes the slice that updates its weights at time slot t'. We have that

$$\begin{aligned} a_b^o(t) &= \sum_{u\in\mathcal{U}_b\setminus\mathcal{U}_b^o} w_u(t) = \sum_{u\in\mathcal{U}_b\setminus\mathcal{U}_b^o} w_u(t(u)) \\ &= \sum_{u\in\mathcal{U}_b\setminus\mathcal{U}_b^o} w_u(t(u)-1)(1+\Delta w_u(t(u))) \end{aligned}$$

where $t(u)$ represents the time when user u updates its weights within the interval $\{t-|\mathcal{O}|+2, ..., t\}$.

If we take the maximum $\Delta w_u(t(u))$ over all users, we obtain the following

inequality

$$a_b^o(t) \leq \max_{u \in \mathcal{U}_b \setminus \mathcal{U}_b^o} (1 + \Delta w_u(t(u))) \sum_{u \in \mathcal{U}_b \setminus \mathcal{U}_b^o} w_u(t(u) - 1)$$
$$= \max_{u \in \mathcal{U}_b \setminus \mathcal{U}_b^o} (1 + \Delta w_u(t(u))) a_b^o(t - |\mathcal{O}| + 1).$$

where the second inequality follows from the fact that we are taking the weight values prior to their update. Given that $a_b^o(t) = a_b^o(t - |\mathcal{O}| + 1)(1 + \Delta a_b^o(t))$, the above leads to

$$1 + \Delta a_b^o(t) \leq \max_{u \in \mathcal{U}_b \setminus \mathcal{U}_b^o} (1 + \Delta w_u(t(u)))$$

From the definition of $\overline{\Delta} w^o(t)$ we have that $\overline{\Delta} w^o(t) \geq \Delta w_u(t)$ for $u \in \mathcal{U}^o$, from which

$$1 + \Delta a_b^o(t) \leq \max_{t' \in \{t - |\mathcal{O}| + 2, \ldots, t\}} 1 + \overline{\Delta} w^{o(t')}(t')$$

Since the above inequality holds for all $b \in \mathcal{B}$, it also holds if we take the maximum value over all b in the rhs, which leads to

$$1 + \overline{\Delta} a^o(t) \leq \max_{t' \in \{t - |\mathcal{O}| + 2, \ldots, t\}} 1 + \overline{\Delta} w^{o(t')}(t').$$

Similarly, it can be shown that

$$1 + \underline{\Delta} a^o(t) \geq \min_{t' \in \{t - |\mathcal{O}| + 2, \ldots, t\}} 1 + \underline{\Delta} w^{o(t')}(t').$$

Combining the above with (4.24) yields

$$\max \left(1 + \overline{\Delta} w^o(t+1), \frac{1}{1 + \underline{\Delta} w^o(t+1)}\right)$$
$$< \max_{t' \in \{t - |\mathcal{O}| + 2, \ldots, t\}} \left(1 + \overline{\Delta} w^{o(t')}(t'), \frac{1}{1 + \underline{\Delta} w^{o(t')}(t')}\right)$$

If we add the term $t - |\mathcal{O}| + 1$ in the $\max$ operation of the rhs, the inequality still holds, thus

$$\max\left(1 + \overline{\Delta} w^o(t+1), \frac{1}{1 + \underline{\Delta} w^o(t+1)}\right) < \max_{t' \in \{t-|\mathcal{O}|+1,\ldots,t\}} \left(1 + \overline{\Delta} w^{o(t')}(t'), \frac{1}{1 + \underline{\Delta} w^{o(t')}(t')}\right)$$

With the above we have proven (4.23). In the next (and last) step of the proof, we now show (4.22) building on this result. For conciseness, we define:

$$f(t) = \max\left(1 + \overline{\Delta} w^{o(t)}(t), \frac{1}{1 + \underline{\Delta} w^{o(t)}(t)}\right). \tag{4.30}$$

With the above definition, (4.23) can be rewritten as

$$f(t+1) < \max_{t' \in \{t-|\mathcal{O}|+1,\ldots,t\}} f(t'). \tag{4.31}$$

Applying the same argument to the update at time slot $t+2$ yields

$$f(t+2) < \max_{t' \in \{t-|\mathcal{O}|+2,\ldots,t+1\}} f(t') = \max\left[\max_{t' \in \{t-|\mathcal{O}|+2,\ldots,t\}} f(t'), f(t+1)\right]$$

and combining the above with (4.31) we obtain

$$f(t+2) < \max_{t' \in \{t-|\mathcal{O}|+1,\ldots,t\}} f(t').$$

If we now apply the above argument recursively for $t+3, t+4, t+5, \ldots$, we obtain

$$f(t+i) \leq \max_{t' \in \{t-|\mathcal{O}|+1,\ldots,t\}} f(t'),\ i < 1. \tag{4.32}$$

Without loss of generality, let assume that $t+1$ coincides with the start of a round; then from the above it follows that

$$\max_{i \in \{1,\ldots,|\mathcal{O}|\}} f(t+i) < \max_{j \in \{1,\ldots,|\mathcal{O}|\}} f(t - |\mathcal{O}| + j) \tag{4.33}$$

where the $\max$ on the lhs includes the updates corresponding to round $r+1$, while those on the rhs correspond to round r.

Finally, expressing the above equation in terms of $\Delta\omega_u^o(r)$ leads to (4.22).

□

Chapter 5

Inelastic Network Slicing Games: Admission control policies

When employing dynamic sharing mechanisms, tenants may exhibit strategic behavior, optimizing their choices in response to those of other tenants. While this problem has been studied in Chapter 4 for the case of elastic users, this chapter analyzes dynamic sharing in network slicing when tenants support inelastic users with *minimum rate requirements*. When attempting to satisfy such user requirements, tenants' behavior may differ substantially from that in Chapter 4, affecting both network efficiency and stability. The focus of this chapter [1] is thus on the analysis of resource allocation for network slicing when tenants support inelastic users and to deal with this problem, we propose a NEtwork Slicing (NES) framework combining (i) admission control, (ii) resource allocation and (iii) user dropping. We model the network slicing system with admitted users as a *network slicing game*; this is a new class of game where the inelastic nature of the traffic may lead to dropping users whose requirements cannot be met. We show that, as long as admission control guarantees that slices can satisfy the rate requirements of all their users,

[1] Publications based on this chapter: Pablo Caballero, Albert Banchs, Gustavo de Veciana, Xavier Costa-Perez and Arturo Azcorra. Network Slicing Games for Guaranteed Rate Services. In IEEE Transactions on Wireless Communications, [Accepted, to appear]. All co-authors contributed equally.

this game possesses a Nash Equilibrium. Admission control policies (a conservative and an aggressive one) are considered, along with a resource allocation scheme and a user dropping algorithm, geared at maintaining the system in Nash Equilibria. We analyze our NES framework's performance in equilibrium, showing that it achieves the same or better utility than static resource partitioning, and bound the difference between NES and the socially optimal performance. Simulation results confirm the effectiveness of the proposed approach.

5.1 Related work

The resource allocation mechanism analyzed in this chapter, as in the one analyzed in Chapter 4 corresponds to a Fisher market, which is a standard framework in economics. In such markets, buyers (in our case slices) have fixed budgets (in our case corresponding to pre-agreed network shares) and bid for resources within their budget (according to their preferences), which are then allocated to buyers proportionally to their bids [110]. Within the Fisher Market framework, our model falls in the category of buyers that anticipate the impact of their bids [34]. The analysis of Fisher markets under such price-anticipating buyers has been limited, so far, to the case of buyers with *linear* [34] or *concave* [20, 94] utility functions.

A related resource allocation model often considered in the networking field is the so-called 'Kelly's mechanism', which allocates resources to players proportionally to their bids [61]. This model has also been analyzed for price-anticipating players [52]. However, in Kelly's mechanism players respond to their payoff (given

by the utility minus cost) whereas in our model tenants' behavior is only driven by their utilities (since they have a fixed budget, i.e., the network share). Moreover, Kelly's model has mainly been studied for concave utility functions.

The topic of network slicing is currently attracting substantial attention from the research community. One of the main issues investigated is the resource allocation across different slices, which is the focus of this chapter. A number of works have been devoted to the resource allocation among different operators or tenants sharing the same wireless infrastructure (see e.g. [21, 76, 111]), and in [72], the authors focus on resource allocation of processing resources in network slicing in the context of C-RAN; see [93] for a survey on resource slicing in virtual wireless networks. In contrast, all these works have focused on elastic traffic.

In the context of network slicing, there are some works which have considered inelastic traffic. The algorithm proposed in [67] attempts to satisfy the demands of all slices but does not account for the resources each slice is entitled to. Similarly, [42, 50, 83] propose algorithms to meet requests from all tenants, but do not account for elastic demands and do not consider budget constraints. In [12], the authors propose an algorithm to trade resources among tenants, but their approach involves complex negotiations and relies on heuristic considerations rather than a well-established analytical framework. In contrast to all these works, our approach supports both elastic and inelastic services and is based on fixed budgets, corresponding to the *network shares*; this is in line with one of the scenarios considered in 3GPP [5] and does not involve pricing individual requests, which may represent an advantage in practical deployments.

In this work, we build on the Fisher Market mechanism for resource allocation across slices and analyze the game resulting from the interaction of several non-cooperative slices aiming to maximize their own network utility given a fixed budget. This problem has been addressed in the context of concave utility functions: [94] ensures the existence of Nash Equilibria (NE) for this type of utility functions, [110] proves the existence of a NE for price-taking players, [20] shows the convergence of Best Response Dynamics for certain classes of concave functions and [34] shows they may not converge for linear utilities. Much less attention has been paid to non-concave utility functions; among the few works on this topic it is worth mentioning [81], which uses potential games to prove convergence of Best Response Dynamics to a region around the NE for finite strategy games [24].

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

On the 5G standardization front, network slicing is currently being specified by 3GPP [8]. In particular, 3GPP's SA5 is working on the definition of a management and orchestration framework to support network slicing [4, 9]. While these

efforts do not specifically address dynamic resource allocation, which is our focus here, the algorithms we propose are in line with this framework. One of the key features of our approach is the ability of tenants to customize their allocations; there is wide consensus in the standardization community that this is needed to efficiently satisfy their very diverse requirements (see, e.g., [10] for examples of possible vertical tenants).

5.2 Chapter organization

The remaining content of this chapter is organized as follows. In Section 5.3 we present our system model, and propose the Network Slicing (NES) framework to address resource allocation in such system. NES consists of three modules: admission control, weight allocation and user dropping. Section 5.4 focuses on the admission control module: it finds the requirements to ensure stability and proposes two policies, a conservative and an aggressive one, to perform admission control. Section 5.5 presents the other two modules: a resource allocation mechanism and a strategy to drop users when rate guarantees are infeasible, and analyzes the convergence of the resulting dynamics. We then study in Section 5.6 the performance of NES versus two benchmark allocations: static resource partitioning and the social optimal. Throughout the chapter, we present analytical results that support the design of NES, including (i) the existence of a Nash Equilibrium and the convergence of Best Response Dynamics, (ii) the effectiveness of admission control and protection from other slices, (iii) the user selection and weight allocation choices, and (iv) the gains over static slices and loss over social optimal. We further evaluate

the performance of NES via simulation in Section 5.7, confirming that it provides substantial gains in terms of utility, throughput performance and reduced blocking probability while incurring an acceptable complexity. The proof of the theoretical results for this chapter are provided in Section 5.9.

5.3 System model

We consider a wireless network consisting of a set of resources $\mathcal{B}$ (the base stations or sectors) shared by a set of network slices $\mathcal{O}$ (each operated by a different tenant). At a given point in time, the network supports a set of active users $\mathcal{U}$ (the customers or devices), which can be subdivided into subsets $\mathcal{U}_b^o$, $\mathcal{U}_b$ and $\mathcal{U}^o$, corresponding to the users of slice o at base station b, the users at base station b, and the users of slice o, respectively. We consider that the association of users with base stations is fixed (e.g., by a pre-specified user association policy) and let $b(u)$ denote the base station that user u is (currently) associated with.

5.3.1 Resource allocation model

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

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

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

We let c_u denote the achievable rate for user u, defined as the product of (i) the *average* rate per resource unit achieved by the user, and (ii) the total amount of resources available at the base station. Note that this depends on the modulation and coding scheme selected for the current radio conditions, which accounts for noise as well as the interference from the neighboring base stations. Following similar analyses in the literature [43, 76, 97], we shall assume that c_u is fixed for each user at a given time.

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

$$r_u(\mathbf{w}) = \frac{w_u}{\sum_{v \in \mathcal{U}_{b(u)}} w_v} c_u = \frac{w_u}{l_{b(u)}(\mathbf{w})} c_u. \tag{5.1}$$

When implementing the proposed resource allocation mechanism, a slice may assign a non-zero weight to some users while others may be dropped. To decide the setting of the users' weights, we assume that each slice o is aware of the aggregate weight of the other tenants at each base station, i.e., $a_b^o(\mathbf{w}^{-o})$. It is worth

noting that for the mechanism under study we have that (i) a slice only sees the aggregate weight of the other slices, and hence can learn very limited information about the other slices; in particular, the weights of each tenant are not disclosed, and (ii) the mechanism needs to store very limited data; indeed, it is sufficient to keep the total load of each base station, as a tenant can obtain $a_b^o(\mathbf{w}^{-o})$ by simply subtracting its weight from the base station's load. Such information is already considered within the network slicing management system defined by 3GPP [4], and hence should be readily available.

To avoid the indeterminate form resulting from having all the weights at a base station equal to 0 in (5.1), we will require weights to exceed a fixed lower bound (i.e., $w_u \geq \delta, \ \forall u$). This bound can be arbitrarily small; indeed, in practice it should be set as small as possible, to allow slices the highest possible flexibility while avoiding zero weights. Accordingly, in the rest of the chapter we assume that δ is so small that its effect can be neglected, except for Theorem 14, where this assumption is required to prove the existence of a Nash Equilibrium.

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

5.3.2 Slice utility

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

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

For inelastic traffic, we assume each user u requires a guaranteed rate γ_u, hereafter referred to as the user's minimum *rate requirement*. Following standard practice, we shall model inelastic traffic utility functions as[2]

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

where $f_u(\cdot)$ is a concave[3] utility function associated with the user, and ϕ_u is the relative priority of user u (where $\phi_u \geq 0$ and $\sum_{u \in \mathcal{U}^o} \phi_u = 1$). The relative priorities reflect the importance that users are given by the tenant of their slice; they drive, jointly with the load at the respective base stations, the weights assigned to the users, which in turn determine the rate allocation.

Note that the above utility function is only defined for rates above the minimal requirement, as performance degrades drastically if this guarantee is not met.

[2]Inelastic traffic utility functions are typically modeled as a discontinuous function [98] or a sigmoidal one [45]. In this chapter we adopt the former model, which aims at providing users with a guaranteed rate, and thus is aligned with the Guaranteed Bit Rate (GBR) class of 3GPP [2].

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

Note also that the above definition includes elastic traffic, which corresponds to the special case $\gamma_u = 0$; thus, the results of this chapter apply to mixes of elastic and inelastic traffic.

While most of our results hold for arbitrary $f_u(\cdot)$ functions, in some cases we will focus on the following widely accepted family of utility functions (see α-fairness, [80]):

$$f_u(r_u) = \begin{cases} \frac{(r_u)^{1-\alpha_o}}{(1-\alpha_o)}, & \alpha_o \neq 1 \\ \log(r_u), & \alpha_o = 1, \end{cases} \tag{5.2}$$

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

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

Note that the above framework is sufficiently flexible to accommodate different network slicing models, including those under study in 3GPP [4]. For instance, in the case where tenants are Mobile Virtual Network Operators (MVNOs), the users of a tenant may have different service demands (e.g., elastic and inelastic users). Alternatively, we can also support a model where different slices are deployed for specific services; in this case, we may have some slices with only elastic users and others with only inelastic users.

5.3.3 Baseline allocations

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

Socially Optimal Allocation (SO) If slices were to share their utility functions with a central authority, one could in principle consider a (share-constrained) allocation of weights (and resources) that optimizes the overall performance of the network, expressed in terms of the *network utility* $U(\mathbf{w})$ defined as the sum of the slices' utilities (see [76], Chapter 4):

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

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

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

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

Static Slicing Allocation (SS) By static slicing (also known as static splitting [30]) we refer to a complete partitioning of resources based on the network shares s_o, $o \in$

$\mathcal{O}$. In this setting, each slice o receives a fixed fraction s_o of each resource, which is shared among its users proportionally to their weights,

$$r_u^{ss}(\mathbf{w}^o) = \frac{w_u}{\sum_{v \in \mathcal{U}^o_{b(u)}} w_v} s_o c_u \quad \forall u \in \mathcal{U}^o, \forall o \in \mathcal{O}, \tag{5.3}$$

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

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

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

5.3.4 Network slicing framework

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

1. *Admission control*: the purpose of this module is to ensure that admitted users will see their rate requirements met during their lifetime with a sufficiently high probability, even after there are changes in the network.

2. *Weight allocation*: this module determines how to allocate weights to the users, with the goal of maximizing the slice's utility.

3. *User dropping*: while admission control aims at ensuring that all rate requirements are always met, when users re-associate or see a change in their radio conditions, or when other slices admit more users, it could happen that a slice can no longer keep all its users while meeting their requirements; in that case, this module decides which users to drop.

The design of the admission control module is presented in Section 5.4, while that of the weight allocation and user dropping modules is presented in Section 5.5.

To analyze the stability of the NES framework, we assume that slices are *competitive* (strategic and selfish), i.e., each attempts to unilaterally optimize its own utility, and model the behavior of the *weight allocation* and *user dropping* modules as a non-cooperative game. Note that this game only considers admitted users, i.e., admission control is not part of the game. It may be played at a point in time when admitted users may have re-associated or seen a change in their radio conditions, or new users may have been admitted; as a result, when playing the game we may not be able to meet all rate requirements. Thus, the game involves slices deciding (i) which set of users to serve when the rate requirements of all users cannot be met, and (ii) how to allocate weights amongst the slice's users, in response to other slices' decisions. Hereafter we refer to this game as the *network slicing game*; its formal definition is stated as follows:

Definition 11. *Consider a set of slices $o \in \mathcal{O}$, each with a set of admitted users $u \in \mathcal{U}^o$. In the network slicing game, each slice selects which subset of users to serve within the set $\mathcal{U}^o$ and their associated weight allocation $\mathbf{w}^o$ such that (i) as many users as possible are served (meeting their rate requirements), and (ii) the slice's utility U^o is maximized for the selected subset of users.*

5.4 Admission control for sliced networks

To meet user rate requirements, NES needs to apply admission control on new users, rejecting them when the slice cannot guarantee with a very high probability that it will be able to satisfy the rate requirements of all its users during their lifetime. Note that this only applies to new users; in case the user rate requirements can no longer be satisfied as a result of users moving, or other tenants changing their allocations, this is handled by the *user dropping* module described in Section 5.5.1.

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

5.4.1 Nash Equilibrium existence

A critical question is whether the *network slicing game* defined in Section 5.3.4 possesses a Nash Equilibrium (NE), i.e., there exists a choice of users and associated weight allocation $\mathbf{w}$ such that no slice can unilaterally modify its choice to improve its utility. In the following, we analyze the requirements on admission control policies to ensure that a NE exists *after* admission control is applied. Note that, if the game does not have a NE, strategic slice behavior may lead to system instability affecting the practicality of the proposed approach.

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

Theorem 13. *When slices cannot satisfy all of their users' rate requirements, the existence of a NE cannot be guaranteed for the network slicing game.*

The problem identified by the above theorem can be overcome by applying an admission control scheme that avoids such situations. According to the following theorem, a NE exists as long as admission control is able to guarantee that a

slice can satisfy the rate requirements of all its users under any feasible weight allocation of the other slices (including *future* allocations when possibly new users may have been admitted). Note that in this case the resulting game focuses on maximizing slice utilities while meeting the rate requirements of all users. This result implies that, as long as proper admission control is implemented and ensures that rate requirements can always be satisfied, the stability of the system can be guaranteed.

Theorem 14. *Suppose admission control ensures that, for any feasible weight allocation of the other slices, each slice o has a weight allocation $\mathbf{w}^o$ such that its users' rate requirements are met. Then, the network slicing game has a (not necessarily unique) NE.*

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

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

In the following, we propose two alternative admission control policies (one more aggressive and one more conservative) that aim at ensuring that the conditions given by Theorem 14 are met. Note that it is ultimately up to the tenant to choose and customize its admission control strategy, and hence each tenant may independently apply its *own* admission control policy.

5.4.2 Worst-case admission control (WAC)

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

$$\sum_{u \in \mathcal{U}_b^o} \frac{\gamma_u}{c_u} \leq s_o. \tag{5.4}$$

The WAC policy aims at ensuring that (5.4) is satisfied at all times. However, even if this condition holds when a new user is admitted, it may be subsequently violated upon changes in the slice, e.g., due to mobility of users or changes in their c_u. To provide robustness against such changes, we follow the approach in [91] for single-tenant networks. Specifically, we add a guard band to (5.4) aimed at ensuring that the condition will continue to hold with high probability after any changes. Thus, a slice admits a new user request as long as the following holds

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

where $\rho_w < 1$ parametrizes the guard band: the smaller this parameter, the larger the guard band. In practice, this parameter may be set to different values by different slices based on the slice specifics, such as the fluctuations of c_u or user association

(where larger fluctuations will require a larger guard band) or the desired level of assurance to its users (stricter guarantees will require a larger guard band). The reader is referred to [91] for a discussion on how to set this parameter.

In the following, we analyze the properties of WAC under the assumption that (5.4) is satisfied with this policy. The theorem below shows that, as long as this condition is satisfied, a slice will always be able to meet its users' rate guarantees independent of the setting of the other slices. Thus, a high degree of protection to the choices and changes in other slices is provided. The theorem also shows that if the slice deviates from the proposed policy, it is not protected from the other slices' choices, implying that this policy represents a necessary condition to provide protection.

Theorem 15. *Consider a slice o with users having rate requirements $\boldsymbol{\gamma}^o = (\gamma_u \colon u \in \mathcal{U}^o)$, then the following hold:*

1. *If* (5.4) *is satisfied, there exists at least one weight allocation $\mathbf{w}^o$ such that $\forall u \in \mathcal{U}^o \; r_u(\mathbf{w}) \geq \gamma_u$, for any feasible allocation of the other slices' aggregate weights $\mathbf{a}^o$.*

2. *If* (5.4) *is not satisfied, slice o is not protected, as there is a feasible $\mathbf{a}^o$ allocation such that slice o is not able to meet the rate requirements of its admitted users.*

Note that combining this result with Theorem 14, it follows that a NE exists when all slices run WAC. Indeed, the above theorem ensures that a slice can find

an allocation that meets the rate requirements of all its users for any feasible $\mathbf{a}^o$, which comprises all the possible allocations of the other slices $\mathbf{w}^{-o}$. Theorem 14 guarantees that when this holds, a NE exists. Thus, we have the following corollary:

Corollary 2. *If* (5.4) *is satisfied by all slices, then the network slicing game has a NE.*

Note that Corollary 2 imposes more conservative conditions than Theorem 14; for instance, if a slice never has users at a given base station, according to Theorem 14 such a slice cannot place any weight on this base station; in contrast, the arguments behind (5.4) account for, and protect the slice against, such possibility.

5.4.3 Load-driven admission control (LAC)

While the WAC policy protects a given slice from the others, it may be overly conservative in some cases where base stations are lightly loaded or where some slices are unlikely to use resources at certain base stations. In those cases, one may opt to be more aggressive in admitting users without running significant risks. To this end, we propose the Load-driven Admission Control (LAC) policy, where a slice measures the current load across base stations and performs admission control decisions based on the measured loads (assuming that they will not change significantly).[4]

[4]Note that many similar (load-driven) admission control algorithms have been proposed in the literature [23, 64] in the context of single-tenant networks. In this chapter, we apply this concept to a network slicing setting.

The following theorem provides a basis for the design of the LAC policy. It gives a necessary and sufficient condition that has to be satisfied to meet the rate requirements of the slice's users, given the current weight allocations of the other slices. This constraint is shown to be less restrictive than the one imposed by (5.4), implying that LAC (potentially) allows the admission of more users than WAC.

Theorem 16. *Consider a slice o comprising users with rate requirements $\boldsymbol{\gamma}^o = (\gamma_u : u \in \mathcal{U}^o)$, and suppose the aggregate weight of the other slices is given by $\boldsymbol{a}^o$. Then, a weight allocation $\mathbf{w}^o$ that meets slice o's rate requirements exists if and only if the following is satisfied:*

$$\sum_{b \in \mathcal{B}} \frac{\sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u}{1 - \sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u} a_b^o \leq s_o. \tag{5.5}$$

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

Moreover, if the rate requirements satisfy (5.4)*, then the above condition is satisfied.*

The central idea of the LAC policy is as follows. Upon receiving a request of a new user u with a rate requirement γ_u, slice o assesses the current $\boldsymbol{a}^o$ values in the network and checks whether (5.5) would be satisfied with the new user. According to the theorem, as long as (5.5) is satisfied, the rate requirements can be met *if* the $\boldsymbol{a}^o$ values do not change. However, in practice $\boldsymbol{a}^o$ may change due to the response of the other slices to slice o, or to changes in the other slices (e.g., the admission of new users). We shall address this uncertainty by following a similar approach to

WAC: when admitting a new user, we verify that (5.5) is satisfied with a sufficiently large guard band, i.e.,

$$\sum_{b\in\mathcal{B}} \frac{\sum_{u\in\mathcal{U}_b^o}\gamma_u/c_u}{1-\sum_{u\in\mathcal{U}_b^o}\gamma_u/c_u} a_b^o \leq \rho_l \cdot s_o, \tag{5.6}$$

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

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

Theorem 17. *There exists a ρ_l value sufficiently small such that the rate requirements of all the users of slice o can be met independent of how the other slices change their weights.*

The following corollary follows from the above result and Theorem 15. Indeed, as long as every slice satisfies either (5.4) and (5.6), Theorems 15 and 17 guarantee that all slices can choose a weight allocation that satisfies the rate requirements of all their users. Furthermore, Theorem 14 guarantees that when this holds there exists a NE. These implies that, as long as all slices run either WAC or LAC, the system can be expected to be stable.

Corollary 3. *If either* (5.4) *or* (5.6) *holds for every slice (the latter with a sufficiently small ρ_l), then there exists a NE.*

5.5 Weight allocation and user dropping for Network Slicing

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

5.5.1 User subset selection

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

Below we present the algorithms for two possible approaches for user selection: (i) *MaxSubsetSelection*, which maximizes the cardinality of the subset of served users (thus minimizing user dropping); and (ii) *PriorityUserSelection*,

which uses a priority ordering on a slice's users (enabling a slice to customize its users' service).

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

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

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

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

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

Building on the above notation, we present a greedy solution in Algorithm 3, which provides as output the set of selected users $\tilde{\mathcal{U}}^o$.

The following theorem confirms the effectiveness of this algorithm.

Algorithm 3: MaxSubset Algorithm.

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

Theorem 18. *The MaxSubsetSelection algorithm results in a subset of users that maximizes the number of users the slice can serve and still meet their minimal rate requirements.*

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

5.5.2 Weight allocation

Once a slice has selected a set of users whose requirements can be satisfied, it sets their weights as follows. Given the aggregate weights of the other slices,

$a_b^o(\mathbf{w}^{-o})$, a slice chooses $\mathbf{w}^o$ such that the its utility is maximized, i.e.,

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

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

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

5.5.3 Convergence of best response dynamics

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

The following theorem shows that the above dynamics may not converge. In particular, the proof of the theorem considers an instance satisfying the conditions of Theorem 14, i.e., a feasible instance under admission control, and shows that, even though a NE is guaranteed to exist under such conditions, Best Response Dynamics do not converge.

Theorem 19. *Consider a game instance such that, for each slice $o \in \mathcal{O}$ there exists an allocation satisfying the rate requirements of all its users for any possible allocation of the other slices. Even though a NE is guaranteed to exist under these conditions, Best Response Dynamics may not converge.*

While the above theorem shows that convergence cannot be ensured, our simulation results show that in practice Best Response Dynamics converge quickly to a region close to the NE, and hence we can simply force the system to halt after a number of best response rounds and use the weights obtained in the last round. Specifically, following the results provided in Section 5.7.4, in our simulations we halt the system after 7 rounds.

From the above, it can be seen that NES incurs an acceptable computational load, as its execution involves solving a sequence of convex optimization problems (each of which scales with the number of users of the slice and number of base stations) for a limited number of times (namely, the number of slices in the network multiplied by 7). Moreover, the above computations may be possibly performed at centralized controllers, as the resource allocation does not need to be implemented in the base stations before the sequence of optimizations converges or stops. Also, resources may be re-allocated only periodically to alleviate the overhead associated to the reconfiguration of base stations. Quantitative results on the computational load are provided in Section 5.7.5.

5.6 Analysis of the NES framework

In the following, we analyze the performance achieved by the NES approach proposed above as compared to the two baseline allocations given in Section 5.3.3: (i) the socially optimal allocation, and (ii) static slicing. Our analysis assumes that NES reaches a Nash equilibrium.

5.6.1 Gain over static slicing

The result below shows that NES outperforms static slicing.

Theorem 20. *For the same set of admitted users, the utility achieved by an operator under NES is never lower than the utility that this operator would obtain under static slicing.*

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

5.6.2 Loss over the socially optimal allocation

We now study the difference in the utility achieved under socially optimal resource allocation vs. that achieved under NES. We focus on the case where $f_u(\cdot)$ follows (5.2) for $\alpha_o = 1$ and $\alpha_o = 2$, which are two highly relevant settings in

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

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

Theorem 21. *Let user utilities $f_u(\cdot)$ follow* (5.2)*, $\underline{\gamma}_u$ be the minimum rate guarantee in the network, $\bar{c}_u$ be the largest possible achievable rate and $\varepsilon = \underline{\gamma}_u / \bar{c}_u$. Under a given set of admitted users, we have that:*

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

Note that, according to the above results, the bound on the LSO relaxes as we decrease the minimum rate requirement in the network, and becomes unbounded in the case where we have elastic traffic with no rate guarantees, i.e., $\gamma_u = 0$.

However, in a well provisioned network all users should experience a sufficiently large rate, and in this case the LSO should be low according to the above result. This is corroborated by our simulation results, which show that in practice NES performance is close to optimal and LSO is very small.

5.7 Performance evaluation

We next evaluate the performance of NES via simulation. Unless otherwise stated, the mobile network setup of our simulator follows the IMT-A evaluation guidelines for dense 'small cell' deployments [1], considering a network with 19 base stations disposed in a hexagonal grid layout with 3 sectors, i.e., $|\mathcal{B}| = 57$. User mobility follows the Random Waypoint (RWP) model. The users arrive to the network following a Poisson Process with intensity λ arrivals/sec, and their holding times are exponentially distributed. Users' SINR is computed based on physical layer network model specified in [1] (which includes path loss, shadowing, fast fading and antenna gain) and user association follows the strongest signal policy. The achievable rate for a user u, c_u, is determined based on the thresholds reported in [7]. Unless otherwise stated, the rate requirement of the inelastic users is set to $\gamma_u = 0.5$ Mbps, we have $\alpha_o = 1$ for all slices, there are 5 slices in the network with equal shares, the arrival rate is $\lambda = 5$ (equally split among slices) and the average holding time is 1 minute. In the simulations, we consider both slices with mixed traffic of different types (Sections 5.7.1 and 5.7.3) as well as slices dedicated to one specific traffic type (Section 5.7.2). All confidence intervals are below 1%.

5.7.1 Network utility

We first analyze the network utility achieved by NES as compared to the two benchmark solutions presented in Section 4.3.3 (namely, SS and SO). To ensure that the rate requirements of admitted users are always met, we adopt the WAC admission control policy with $\rho_w = 1$ and suppress user movements yielding changes in base station associations and/or c_u values. To analyze the impact of inelastic traffic, we vary the fraction of inelastic traffic arrivals, θ, yielding an arrival rate of $\theta\lambda$ for inelastic users and of $(1-\theta)\lambda$ for elastic ones. The results, depicted in Fig. 5.1, show that (i) NES outperforms very substantially SS, providing very high gains, and (ii) it performs almost optimally, very close to the SO. Moreover, this holds independently of the mix of elastic and inelastic users present in the network.

5.7.2 Throughput gains

To give a more intuitive measure of the gains achieved by NES, we define the throughput gain over SS, Δ, as follows: it is the value such that, if we increase the rate of all users in SS by Δ, we reach the same network utility as NES (e.g., $\Delta = 100\%$ means that SS achieves the same utility as NES when multiplying all user rates by 2). Fig. 5.2 illustrates the throughput gains for (i) $\alpha_o = 1$ and $\alpha_o = 2$, which are the two most relevant α_o values in practice, (ii) elastic and inelastic slices, where all users are either elastic and inelastic, and (iii) different arrival rates λ, yielding different network loads. We conclude from the results that (i) gains are very substantial, ranging from 100% to 20%, (ii) they decrease with the load, as already observed in Chapter 4, and (iii) they are fairly insensitive to the fraction of

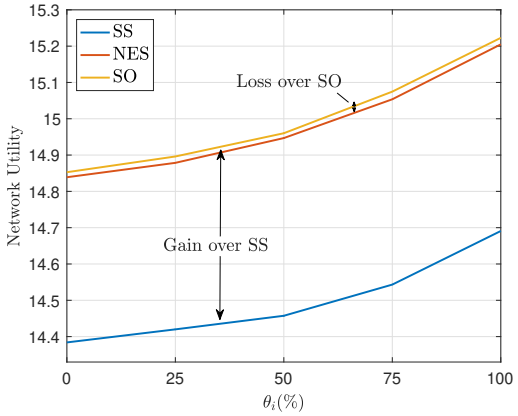

Figure 5.1: Performance of NES in terms of network utility as compared to the two benchmark allocations (SS and SO).

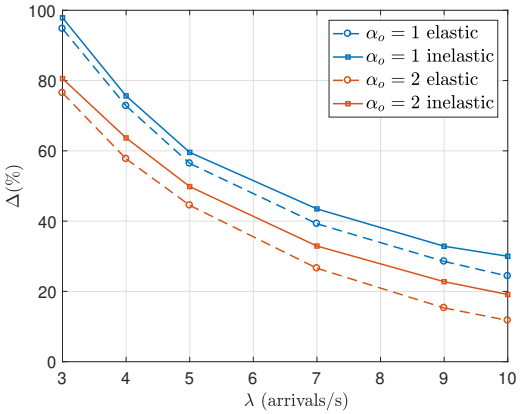

Figure 5.2: Throughput gains over SS for different traffic types (elastic, inelastic), utility functions (α_o) and network load (λ).

inelastic traffic and choice of utility function.

5.7.3 Blocking probability

In addition to improving the performance of admitted users, one of the key advantages of the dynamic resource allocation implemented by NES is that it allows admitting more users while meeting their rate requirements. To assess the achieved improvement, we evaluate the blocking probability (i.e., the probability that a new user cannot admitted) under NES versus SS. For NES, we consider the two admission policies proposed in Section 5.4 (WAC and LAC), while for SS we apply the policy given in [91]. For all settings, we drop users based on the *MaxSubsetSelection* algorithm, and adjust the guard bands to ensure that the probability of dropping an admitted user is no more than 1%. To increase the offered load sufficiently so that we can observe the behavior of the blocking probability, we set $\gamma_u = 1$ Mbps

and an average holding time of 2 minutes. The results are given in Fig. 5.3 as a function of the fraction of inelastic user arrivals (θ). They show very high gains over SS for both approaches (WAC and LAC), and confirm that, by behaving more aggressively, LAC is able to admit many more users than WAC.

5.7.4 Convergence to the NE

To better understand the dynamics of NES, we have evaluated a very large number of randomly generated scenarios (namely 10^4 scenarios) with the following settings: (i) a uniform number of slices between 2 and 10, i.e., $|\mathcal{O}| \sim U(2, 10)$, ($ii$) a number of users per slice of $|\mathcal{U}^o| \sim U(0, 350)$, ($iii$) inelasticity level $\theta \sim U(0, 100)(\%)$, ($iv$) minimum rate requirements $\gamma_u \sim U(0, 3)$ Mbps, and (v) the shares s_o proportional to the number of users. We have found that a vast majority (97.6%) of scenarios converge to the NE after 100 rounds. For such scenarios, Fig. 5.4 shows the difference between the weight allocation at a given round and the one at the NE in terms of mean squared error (RMSE), providing a box plot with the median (red), 95% percentile (box), 99% percentile (whisker) and outliers (red crosses). We observe that the RMSE decreases exponentially in the number of rounds. After 7 rounds we are already very close to the NE (the median is below 10^{-4}), which justifies our choice in Section 5.5.3. Additional results, not included for space reasons, show that user rates exhibit a very similar behavior to the weights.

5.7.5 Computational load

Next we evaluated the computational complexity of the NES algorithm when the system halts after 7 rounds (as given by the configuration chosen). Fig. 5.5 shows the computational times for a dual-core 2.9GHz i7 processor for elastic and inelastic traffic and different numbers of slices and users, when the number of base stations is scaled with the number of users and admission control is adjusted to ensure that dropping probabilities below 1%. Results confirm that NES can be applied to practical settings, as complexity is roughly linear with the size of the network and computational times remain low even for large size problems; for instance, for a network with 9000 users the time falls below 2.5 seconds. We further observe that inelastic traffic slightly increases complexity but does not challenge the practicality of the approach. Finally, we note that the computational time values provided here could be further improved by optimizing the code, parallelizing tasks and/or increasing the machine computational power.

5.7.6 Slice differentiation

We next analyze the ability of NES to deploy slices providing a customized service. To this end, we consider a scenario with 4 slices with different requirements: (i) slice 1 provides rate requirements of $\gamma_u = 1$ Mbps with WAC, (ii) slice 2 provides $\gamma_u = 0.5$ Mbps with WAC, (iii) slice 3 provides $\gamma_u = 0.5$ Mbps with LAC, and (iv) slice 4 provides no minimum rate requirements. All slices have the same share, the arrival rate is of $\lambda = 10$ equally split among the slices, and admission control is configured to provide dropping probabilities below 1%. Fig. 5.6

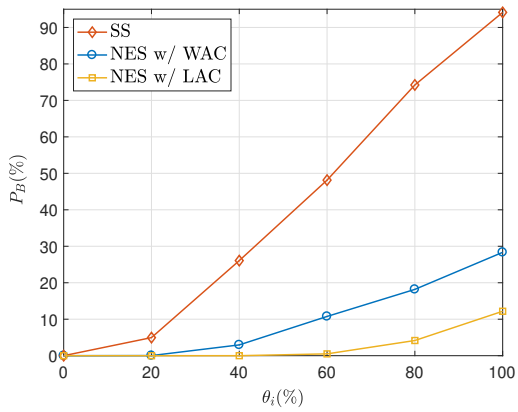

Figure 5.3: Blocking probability for new arrivals for the two policies proposed and the SS benchmark.

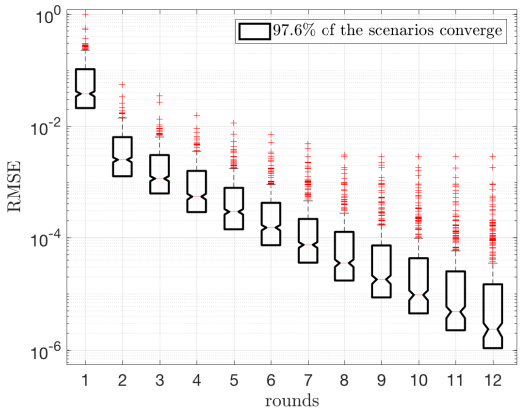

Figure 5.4: Box plot for the RMSE of the weight allocation at a given round with respect to the NE weight allocation.

shows the empirical CDF of the user rates for each slice as well as the blocking probabilities ($\approx 47.2\%, 16.7\%, 3.58\%$ and 0%, respectively). We observe that (i) the minimum rate requirements are satisfied for all slices; (ii) as the rate requirements increase, so does the blocking probability, yielding an overall improvement of the user rate distribution, and (iii) by employing LAC, we achieve a dramatic reduction of the blocking probability while paying a small price in terms of user rate distribution. We conclude that NES is effective in enabling slice differentiation.

5.8 Conclusions

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

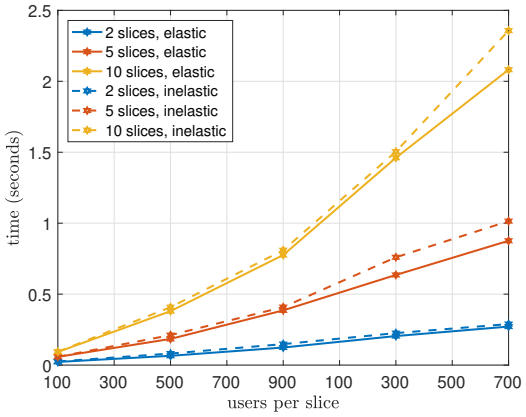

Figure 5.5: Computational times of the proposed approach as a function of the number of slices and users in the network.

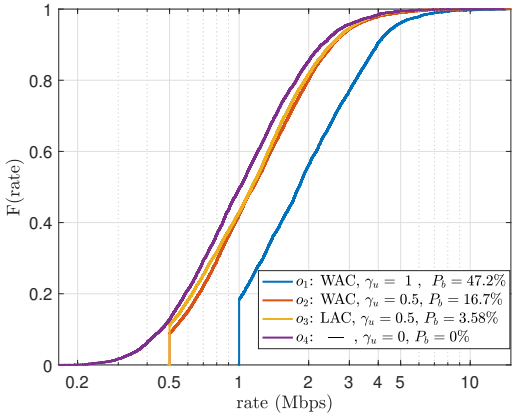

Figure 5.6: Blocking probability and empirical CDF of the user rates for a scenario of 4 slices with different requirements.

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

5.9 Proofs of chapter results

5.9.1 Proof of Theorem 13

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

5.9.2 Proof of Theorem 14

Let $\mathcal{W}$ be the convex and compact set of feasible weights $\mathbf{w}$ satisfying (i) $w_u \geq \delta \ \forall u$, and (ii) $\sum_{u \in \mathcal{U}_o} w_u = s_o \ \forall o$ and let us consider the mapping $\mathbf{w} \rightarrow \tilde{\mathbf{w}} = \Gamma(\mathbf{w})$, where $\tilde{\mathbf{w}}^o$ is the best response of slice o to $\mathbf{w}^{-o}$. We next show that this mapping satisfies the conditions of Kakutani's theorem: $i)$ $\Gamma(\mathbf{w})$ is non-empty, $ii)$ $\Gamma(\mathbf{w})$ is a convex-valued correspondence, and $iii)$ $\Gamma(\mathbf{w})$ has a closed graph. Conditions $i)$ and $ii)$ follow from the fact that the best response of a slice to $\mathbf{w}^{-o}$ is a unique allocation $\tilde{\mathbf{w}}^o$. This implies that that $\tilde{\mathbf{w}}$ exists and is a single point (and

hence a convex set). Condition iii) is shown by proving that $\tilde{\mathbf{w}}^o$ is a continuous function of $\mathbf{w}^{-o}$ for all slices. Consider the set of users for which $r_u > \gamma_u$ and the set for which $r_u = \gamma_u$. As long as these sets do not change, $\tilde{\mathbf{w}}^o$ can be expressed as a continuously differentiable function of $\{\tilde{\mathbf{w}}^o, \mathbf{w}^{-o}\}$, and it follows from the implicit function theorem that $\tilde{\mathbf{w}}^o$ is a continuous function of $\mathbf{w}^{-o}$. When some user moves from set $r_u > \gamma_u$ to $r_u = \gamma_u$ (or viceversa), such user satisfies both the equation for $r_u = \gamma_u$ and the one for $r_u > \gamma_u$, providing continuity over the transitions. Since all the conditions of Kakutani's theorem are satisfied, we have that the mapping Γ has at least one fixed point, which implies that at least one NE exists.

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

5.9.3 Proof of Theorem 15

The result of 1) follows directly from Lemma 2 in Chapter 4. If users are admitted at base stations such that under static slicing their rate guarantees are met, i.e. $r_u^{ss} \geq \gamma_u$, then it follows by the above mentioned lemma that there exists an

allocation satisfying $r_u \geq r_u^{ss} \geq \gamma_u$, which proves the first part of the theorem.

To prove 2), we proceed as follows. Suppose slice o admits users are such that their associated rate requirements violate (5.4) at some base station b, i.e., $\sum_{u \in \mathcal{U}_b^o} \gamma_u / c_u > s_o$. If all other slices place their entire share at that base station, we have

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

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

5.9.4 Proof of Theorem 16

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

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

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

$$\sum_{b \in \mathcal{B}} \frac{\sum_{u \in \mathcal{U}_b^o} r_u / c_u}{1 - \sum_{u \in \mathcal{U}_b^o} r_u / c_u} a_b^o = s_o. \tag{5.7}$$

We now prove that as long as (5.5) is satisfied, there exists a weight allocation $\mathbf{w}^o$ that meets the rate requirements of all users. Let us consider the weight

allocation satisfying[5]

$$w_u = \frac{(\gamma_u/c_u)l_{b(u)}}{\sum_{v\in\mathcal{U}^o}(\gamma_v/c_v)l_{b(v)}}s_o,\ \forall u \in \mathcal{U}^o. \tag{5.8}$$

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

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

which contradicts (5.5). Hence, it follows that $r_u \geq \gamma_u\ \forall u$.

We next prove that if (5.5) is not satisfied, then there exists no weight allocation meeting the rate requirements. The proof goes by contradiction. Assume (5.5) is not satisfied but $r_u \geq \gamma_u\ \forall u$. From the latter, it follows that $\sum_{u\in\mathcal{U}_b} r_u/c_u \geq \sum_{u\in\mathcal{U}_b}\gamma_u/c_u\ \forall b$. Combining this with (5.7) yields

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

which contradicts that assumption that (5.5) is not satisfied.

Finally, we show that if the rate requirements satisfy (5.4), then they surely satisfy (5.5). The lhs of (5.5) increases with $\sum_{u\in\mathcal{U}_b^o}\gamma_u/c_u$. As long as this value is no larger than s_o, we have that the following equation gives a sufficient condition for (5.5) to be satisfied: $\frac{s_o}{1-s_o}\sum_{b\in\mathcal{B}} a_b^o \leq s_o$.

[5]The existence of such an allocation follows from applying Brouwer fixed-point theorem to the function $\mathbf{f} : \mathcal{W} \to \mathcal{W}$, where $w_u = f_u(\mathbf{w})$ is given by (5.8) and $\mathcal{W}$ is the set of weights satisfying $\sum_{u\in\mathcal{U}^o} w_u = s_o$ and $w_u \geq (\gamma_u/c_u)a_b^o s_o/\sum_{v\in\mathcal{U}^o}(\gamma_v/c_v)$ (recall that $a_b^o \neq 0\ \forall b$, as weights cannot be zero).

The above is surely satisfied since $\sum_{b\in\mathcal{B}} a_b^o = 1 - s_o$. As (5.4) imposes $\sum_{u\in\mathcal{U}_b^o} \gamma_u/c_u \leq s_o$, it follows that as long as (5.4) is satisfied, (5.5) is also satisfied. □

5.9.5 Proof of Theorem 17

Let us take $\rho_l = \min_b \frac{a_b^o}{1-\sum_{u\in\mathcal{U}_b^o}\gamma_u/c_u}$. Then, from (5.6) it follows that $\sum_{b\in\mathcal{B}}\sum_{u\in\mathcal{U}_b^o}\gamma_u/c_u \leq s_o$. From this, we have that condition (5.4) is satisfied. According to Theorem 15, as long as this condition is satisfied, there exists a choice of $\mathbf{w}^o$ that satisfies the rate requirements of all users of slice o independent of the weight setting of the other slices, which completes the proof. □

5.9.6 Proof of Theorem 18

The proof goes by contradiction. Let $\tilde{\mathcal{U}}^o$ be the set of users selected by the *MaxSubsetSelection* algorithm, and let us assume that there exists an alternative feasible user selection $\hat{\mathcal{U}}^o$ such that $|\hat{\mathcal{U}}^o| > |\tilde{\mathcal{U}}^o|$. If we take the set $\hat{\mathcal{U}}^o$ and substitute each user by another one in the base station with smaller γ_u/c_u, the resulting set $\bar{\mathcal{U}}^o$ is feasible and has the same number of users as the original one. Note that set $\bar{\mathcal{U}}^o$ necessarily has some base station b with more users than set $\tilde{\mathcal{U}}^o$ – otherwise $|\hat{\mathcal{U}}^o| > |\tilde{\mathcal{U}}^o|$ would not hold. Let us assume that there exists some other base station b' with fewer users. In this case, let us remove user u from one of the base stations with more users, b, and add user u' in one of the base stations with fewer users, b'. The resulting set remains feasible, as $\Delta\omega_{b'}^o(\bar{\mathcal{U}}^o, u') \leq \Delta\omega_{b'}^o(\bar{\mathcal{U}}^o, u)$ – otherwise *MaxSubsetSelection* would have chosen a different subset of users. We can do this

until there are no base station with fewer users than in $\tilde{\mathcal{U}}^o$. The result of these operations is a feasible set where all base stations have as many users or more than $\tilde{\mathcal{U}}^o$, and overall it has more users. However, this yields a contradiction: if such set was feasible, the *MaxSubsetSelection* algorithm would have selected more users. □

5.9.7 Proof of Theorem 19

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

5.9.8 Proof of Theorem 20

The proof follows from Lemma 2 in Chapter 4, which shows that, given a slice o and a feasible weight allocation $\mathbf{w}^{-o}$ for the other slices, there exists a weight allocation $\mathbf{w}^o$ for slice o, possibly dependent on $\mathbf{w}^{-o}$, such that the resulting

weight allocation $\mathbf{w}$ satisfies $r_u(\mathbf{w}) \geq r_u^{ss}$ for all $u \in \mathcal{U}^o$. Therefore, there exists a weight allocation that provides the same utility as static slicing. Since the weight allocation chosen by NES is the one that maximizes the slice's utility, it surely provides a utility no smaller than that under static slicing. □

5.9.9 Proof of Theorem 21

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

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

Furthermore, from the minimum rate constraint it follows that

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

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

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

following weight allocation: $w_{11} = \left(\frac{1}{2} - \varepsilon\right)\left(\frac{1}{2} + \frac{\varepsilon}{2(1-\varepsilon)}\right)$, $w_{12} = 1/2 - w_{11}$, $w_{21} = \frac{\varepsilon}{2(1-\varepsilon)}$ and $w_{22} = 1/2 - w_{21}$, from which $U(\mathbf{w}^*) = \frac{1}{2}\log\left((1/2)\bar{c}_u\right) + \frac{1}{2}\log(\bar{c}_u)$. This yields $U(\mathbf{w}^*) - U(\hat{\mathbf{w}}) = -\frac{1}{2}\log\left(2\varepsilon\right)$, which terminates the proof for $\alpha_o = 1$.

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

$$U(\mathbf{w}^*) \geq -\sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u\frac{1}{\bar{c}_u} \qquad \text{and} \qquad U(\hat{\mathbf{w}}) =\leq -\sum_{o\in\mathcal{O}}\sum_{u\in\mathcal{U}_o} s_o\phi_u\frac{1}{\underline{\gamma}_u}.$$

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

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

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this thesis, we have demonstrated that substantial performance improvements (up to 50% capacity savings) are achievable by properly engineering multi-tenant RAN virtual pools. The cooperative dynamic resource allocation, proposed in Part 1 of this thesis, trades-off statistical multiplexing and slice performance differentiation. In this setting we showed that it is appropriate to perform virtual pooling of resources where slices have similar share and load profiles to exploit multiplexing while maintaining the ability to achieve a degree of tenant isolation and differentiation. The gains achieved by our cooperative scheme increases in the number of slices and the size of the network and as seen in Chapter 2 can be further improved by optimizing user association.

Given the possible requirement to allow tenants more freedom in allocating their resources, Part 2 of this thesis demonstrates that competitive dynamic resource allocation, under mild conditions, results in a network slicing game which has many desirable properties. The game is ensured to have a Nash Equilibrium allocation, which is reachable through Best Response updates and which has a low price of anarchy. This makes the framework also a good candidate to realize multi-tenant

RAN slicing. Note that if customers require minimum rate guarantees, admission control and user dropping mechanisms would be required. We propose a low complexity and reliable worst case admission control mechanism, as well as a more aggressive load-based admission control, allowing tenants to make opportunistic use of network resources under the network slicing game.

Overall, in this thesis we presented two different frameworks to realize RAN slicing. Our cooperative allocation framework stands out for its simplicity, implementability and for its ability to dynamically allocate resources across tenants while preserving slice isolation. However, the predefined allocation policy limits tenants' customization. The competitive framework enables a higher degree of customization at the expense of a small network performance degradation (price of anarchy) and a higher implementability cost. While the work in this thesis does not cover all possible settings, it provides a possible framework for RAN architects to consider along side an initial understanding of the tradeoffs among resource utilization, customization, isolation and implementability in multi-tenant scenarios and presents different solutions that help meeting various network design goals.

As a consequence of the research conducted in this thesis, we see further research directions that are open for future work. Two key main research directions that would complement this work are briefly described next.

1. Coupled resources network slicing. In this thesis, we have focused on RAN resource allocation. In particular, we focused on a setting where, resources are decoupled, i.e. a user is not using radio resources from multiple locations at the same time. A generalized version of this problem would involve joint resource allo-

cation of multiple possibly heterogeneous resources, such as RAN, computational and memory resources. Although some of our work can be used in this relevant generalization, the problem is different in nature, since resources are coupled, i.e. the same user will be simultaneously using radio, computational and memory resources. Although some joint schedulers have been proposed, see e.g., [51,69], this area is still understudied and current solutions do not share the same vision and objectives drawn in this thesis regarding network slicing resource allocation.

2. Elastic applications performance analysis. The expected performance/ utility of the proposed mechanisms were evaluated in the setting where users were rate-adaptive (with minimum rate requirements in Chapter 5.) and full-buffer applications. The key assumption is that the time of an active user in the system does not depend on the resources it was allocated, i.e., "a well-engineered" network for voice, video, etc. In the case of elastic applications, the performance and dynamics of the system is fundamentally different. It would be of great interest to extend our results on network slicing to capture the interactions (or lack thereof) among slices supporting elastic customers (some results on stability can be found in [38]) and its study is worth of a separate research effort.

Bibliography

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

[2] 3GPP. Technical Specification Group Services and System Aspects; Policy and charging control architecture. 3GPP TS 23.203, Jun. 2016.

[3] 3GPP. Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions. 3GPP TS 36.902 v9.3.0, March 2011.

[4] 3GPP. Management of 5G networks and network slicing; Concepts, use cases and requirements (Release 15). TS 28.530, v0.6.0, Apr. 2018.

[5] 3GPP. Study on Radio Access Network (RAN) Sharing Enhancements. 3GPP TR 22.852, v12.0.0, Jun. 2013.

[6] 3GPP. Network Sharing; Architecture and Functional Description. 3GPP TS 23.251, v12.1.0, Jun. 2014.

[7] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures. TS 36.213, v12.5.0, Rel. 12, Mar. 2015.

[8] 3GPP. Study on Architecture for Next Generation System. TR 23.799, v0.5.0, May 2016.

[9] 3GPP. Study on management and orchestration of network slicing for next generation network (Release 15). TR 28.801 V1.2.0, May 2017.

[10] 5GPPP White paper. 5G Empowering vertical industries. 2016.

[11] 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.

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

[13] 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.

[14] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. What Will 5G Be? *IEEE Journal on Selected Areas in Communications*, 32(6):1065–1082, June 2014.

[15] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang. RAT selection games in HetNets. In *Proc. of IEEE INFOCOM*, April 2013.

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

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

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

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

[20] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez. Network slicing games: Enabling customization in multi-tenant networks. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[21] 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.

[22] P. Caballero, X. Costa-Perez, K. Samdanis, and A. Banchs. RMSC: A Cell Slicing Controller for Virtualized Multi-Tenant Mobile Networks. In *Proc. of IEEE VTC*, May 2015.

[23] R.D. Callaway, M. Devetsikiotis, and C. Kan. Design and implementation of measurement-based resource allocation schemes within the realtime traffic flow measurement architecture. In *Proc. of IEEE ICC*, June 2004.

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

[25] 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.

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

[27] Coleago consulting. Mobile network sharing report, September 2015.

[28] Federal Communications Commision. The Next Step for LTE-U: Conducting Limited LTE-U Performance Tests, January 2016. Available at https://www.fcc.gov/news-events/blog/2016/01/29/next-step-lte-u-conducting-limited-lte-u-performance-tests.

[29] Federal Communications Commision. 3.5 GHz Band / Citizens Broadband Radio Service, July 2016. Available at https://www.fcc.gov/rulemaking/12-354.

[30] X. Costa-Perez et al. Radio access network virtualization for future mobile carrier networks. *IEEE Communications Magazine*, 51(7):27–35, July 2013.

[31] 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.

[32] H.S. Dhillon, R.K. Ganti, F. Baccelli, and J.G. Andrews. Modeling and Analysis of K-Tier Downlink Heterogeneous Cellular Networks. *IEEE Journal on Selected Areas in Communications*, 30(3):550–560, April 2012.

[33] L. Doyle, J. Kibida, T. K. Forde, and L. DaSilva. Spectrum without bounds, networks without borders. *Proceedings of the IEEE*, 102(3):351–365, March 2014.

[34] M. Feldman, K. Lai, and L. Zhang. The Proportional-Share Allocation Market for Computational Resources. *IEEE Transactions on Parallel and Distributed Systems*, 20(8):1075–1088, Aug. 2009.

[35] P. Di Francesco, F. Malandrino, T. K. Forde, and L. A. DaSilva. A Sharing- and Competition-Aware Framework for Cellular Network Evolution Planning. *IEEE Transactions on Cognitive Communications and Networking*, 1(2):230–243, June 2015.

[36] T. Frisanco, P. Tafertshofer, P. Lurin, and R. Ang. Infrastructure sharing and shared operations for mobile network operators From a deployment and operations view. In *Proc. of IEEE NOMS*, Salvador, Brazil, April 2008.

[37] M. Gairing, B. Monien, and K. Tiemann. Routing (un-) splittable flow in games with player-specific linear latency functions. *Lecture Notes in Computer Science*, 4051:501–512, 2006.

[38] A. Ganesh, S. Lilienthal, D. Manjunath, A. Proutiere, and F. Simatos. Load

balancing via random local search in closed and open systems. *SIGMETRICS Perform. Eval. Rev.*, 38(1):287–298, June 2010.

[39] 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.

[40] C. Georgiou, T. Pavlides, and A. Philippou. Network uncertainty in selfish routing. In *Proc. of IEEE IPDPS*, Apr. 2006.

[41] R. J. Gibbens and F. P. Kelly. Resource Pricing and the Evolution of Congestion Control. *Automatica*, 35(12):1969–1985, 1999.

[42] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen. A Service-oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory. *IEEE Access*, to appear.

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

[44] 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.

[45] P. Hande, S. Zhang, and M. Chiang. Distributed rate allocation for inelastic flows. *IEEE/ACM Transactions on Networking*, 15(6):1240–1253, December 2007.

[46] F. B. Hildebrand. Advanced calculus for applications. 1962.

[47] I. Hou and C.S. Chen. Self-organized resource allocation in LTE systems with weighted proportional fairness. In *Proc. of IEEE ICC*, May 2012.

[48] I. Hou and P. Gupta. Proportionally fair distributed resource allocation in multiband wireless systems. *IEEE/ACM Transactions on Networking*, 22(6):1819–1830, December 2014.

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

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

[51] M. Jiang, M. Condoluci, and T. Mahmoodi. Network slicing in 5g: An auction-based model. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[52] R. Johari and J.N. Tsitsiklis. Efficiency Loss in a Network Resource Allocation Game. *Mathematics of Operations Research*, 29(3):407–435, August 2004.

[53] R. Johari and J.N. Tsitsiklis. Efficiency of scalar-parameterized mechanisms. *Operations Research*, 57(4):823–839, August 2009.

[54] V. Joseph and G. de Veciana. Stochastic networks with multipath flow control: Impact of resource pools on flow-level performance and network congestion. *SIGMETRICS Perform. Eval. Rev.*, 39(1):61–72, June 2011.

[55] R. Jovanovic, A. Bousselham, and S. Voß. A heuristic method for solving the problem of partitioning graphs with supply and demand. *Annals of Operations Research*, 235(1):371–393, Dec 2015.

[56] 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.

[57] W. N. Kang, F. P. Kelly, N. H. Lee, and R.J. Williams. State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. *The Annals of Applied Probability*, 19(5):1719–1780, 2009.

[58] F. P. Kelly. Charging and rate control for elastic traffic. *European Transaction Telecommunications*, 8(1):33–37, Feb. 1997.

[59] F. P. Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[60] F. P. Kelly, L. Massoulié, and N.S. Walton. Resource pooling in congested networks: proportional fairness and product form. *Queueing Systems*, 63(1-4):165, 2009.

[61] 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(3):237–252, March 1998.

[62] 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, 08 2004.

[63] P. Key, L. Massoulie, and D. Towsley. Combining multipath routing and congestion control for robustness. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 345–350, March 2006.

[64] J. Kim and A. Jamalipour. Traffic management and QoS provisioning in future wireless IP networks. *IEEE Personal Communications*, 8(5):46–55, October 2001.

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

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

[67] J. Kwak, J. Moon, H. W. Lee, and L. B. Le. Dynamic network slicing and resource allocation for heterogeneous wireless services. In *Proc. of IEEE PIMRC*, October 2017.

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

[69] M. Leconte, G. Paschos, P. Mertikopoulos, and U. Kozat. A resource allocation framework for network slicing, 2017. submitted.

[70] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Joint resource allocation and base-station assignment for the downlink in CDMA networks. *IEEE/ACM Transactions on Networking*, 14(1), February 2006.

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

[72] 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*, to appear.

[73] B. Leng, P. Mansourifard, and B. Krishnamachari. Microeconomic Analysis of Base-station Sharing in Green Cellular Networks. In *Proc. of IEEE INFOCOM*, April 2014.

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

[75] B. Loeffler. Cloud computing: What is infrastructure as a service. Microsoft TechNet Magazine, October 2011. Available at https://technet.microsoft.com/en-us/library/hh509051.aspx.

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

[77] I. Malanchini, S. Valentin, and O. Aydin. Generalized resource sharing for multiple operators in cellular wireless networks. In *Proc. of IWCMC*, Aug. 2014.

[78] M. Mavronicolas, I. Milchtaich, B. Monien, and K. Tiemann. Congestion games with player-specific constants. In *Proc. of Mathematical Foundations of Computer Science*, August 2007.

[79] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

[80] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking (ToN)*, 8(5):556–567, October 2000.

[81] D. Monderer and L.S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

[82] M. Morgan and V. Grout. Finding optimal solutions to backbone minimisation problems using mixed integer programming. In *INC*, pages 53–63, 2008.

[83] O. Narmanlioglu, E. Zeydan, and S. S. Arslan. Service-Aware Multi-Resource Allocation in Software-Defined Next Generation Cellular Networks. *IEEE Access*, 6(1):1–15, February 2016.

[84] 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.

[85] NGMN Alliance. 5G White paper, February 2015.

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

[87] D. Nicoara, S. Kamali, K. Daudjee, and L. Chen. Hermes: Dynamic partitioning for distributed social network graph databases. In *EDBT*, pages 25–36, 2015.

[88] C. Ovando, Z. Frias, and JC. Bocarando. Connecting the unconnected: The case of mexicos wholesale shared network. *SSRN Electronic Journal*, 2017.

[89] J. S. Panchal, R. D. Yates, and M. M. Buddhikot. Mobile Network Resource Sharing Options: Performance Comparisons. *IEEE Transactions on Wireless Communications*, 12(9):4470–4482, September 2013.

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

[91] R. Ramjee, D. Towsley, and R. Nagarajan. On Optimal Call Admission Control in Cellular Networks. *Wireless Networks*, 3(1):29–41, March 1997.

[92] S. Rathinakumar and M. K. Marina. GAVEL: strategy-proof ascending bid auction for dynamic licensed shared access. In *Proc. of ACM MobiHoc*, July 2016.

[93] 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, Sept 2016.

[94] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, Jul. 1965.

[95] K. Samdanis and A. H. Aghvami. Load balancing through dynamic partitioning for hierarchical cellular networks. In *2008 International Conference on Telecommunications*, pages 1–6, June 2008.

[96] K. Samdanis, X. Costa-Perez, and V. Sciancalepore. From Network Sharing to Multi-tenancy: The 5G Network Slice Broker. *IEEE Communications Magazine*, 54(7):32–39, Jul. 2016.

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

[98] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal of Selected Areas in Communications*, 13(7):1176–1188, September 2006.

[99] J. Singh. A characterization of positive poisson distribution and its statistical application. *SIAM Journal on Applied Mathematics*, 34(3):545–548, 1978.

[100] K. Son, S. Chong, and G. D. Veciana. Dynamic association for load balancing and interference avoidance in multi-cell networks. *IEEE Transactions on Wireless Communications*, 8(7):3566–3576, July 2009.

[101] M. Song, C. Xin, Y. Zhao, and X. Cheng. Dynamic spectrum access: from cognitive radio to network radio. *IEEE Wireless Communications*, 19(1):23–29, February 2012.

[102] S. Valentin, W. Jamil, and O. Aydin. Extending generalized processor sharing for multi-operator scheduling in cellular networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 485–490. IEEE, 2013.

[103] 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.

[104] J. Westbrook. Load Balancing for Response Time. In *Proc. of the Third Annual European Symposium on Algorithms*, Sep. 1995.

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

[106] 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.

[107] S. Yang and B. Hajek. VCG-Kelly Mechanisms for Allocation of Divisible Goods: Adapting VCG Mechanisms to One-Dimensional Signals. *IEEE Journal on Selected Areas in Communications*, 25(6):1237–1243, August 2007.

[108] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews. User association for load balancing in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications*, 12(6):2706–2716, June 2013.

[109] D. Yuhuan and G. de Veciana. Wireless networks without edges: Dynamic radio resource clustering and user scheduling. In *Proc. of IEEE INFOCOM*, Apr. 2014.

[110] L. Zhang. Proportional response dynamics in the Fisher market. *Theoretical Computer Science*, 412(24):2691–2698, May 2011.

[111] 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.

[112] T. Zhou, Y. Huang, W. Huang, S. Li, Y. Sun, and L. Yang. Qos-aware user association for load balancing in heterogeneous cellular networks. In *Proc. of IEEE VTC Fall*, September 2014.

[113] X. Zhou, R. Li, T. Chen, and H. Zhang. Network slicing as a service: enabling enterprises' own software-defined cellular networks. *IEEE Communications Magazine*, 54(7):146–153, July 2016.

Vita

Pablo Caballero Garcs received his B.S. in Telecommunications Engineering and M.S. in Telematics Engineering from the University Carlos III of Madrid in 2013 and 2015 respectively. In 2015, he joined the Wireless Networking and Communications Group (WNCG) at the University of Texas at Austin to pursue his PhD degree under the supervision of Prof. Gustavo de Veciana and Prof. Albert Banchs while serving as an external PhD student at IMDEA Networks Institute. Prior to that, Pablo served as research assistant at IMDEA Network Institute from 2013 to 2015 and as a Research Intern at NEC Laboratories Europe during 2013. His main research interests lie on the design and performance evaluation of communication networks with a special focus on game theory and algorithm analysis.

Permanent address: pablo.caballero@utexas.edu

This dissertation was typeset by the author.