DIPARTIMENTO DI INGEGNERIA ELETTRICA, ELETTRONICA E INFORMATICA

Dottorato di Ricerca in Ingegneria Informatica e delle Telecomunicazioni
XXVIII Ciclo

Tesi di Dottorato

# USER-CENTRIC RESOURCE ALLOCATION THROUGH A POWER-EFFICIENT JAMMING-PROOF RAN ON TOP OF A MULTI-TENANT BACKHAUL

Ing. Salvatore D'Oro

| Coordinatore | Tutor |
| --- | --- |
| Chiar.ma Prof.ssa V. Carchiolo | Chiar.mo Prof. S. Palazzo |

*to my family, my grandmother and Elisa*

# ABSTRACT

The exponential growth in the number of communicating devices and the increasing demand for better and high-performance communications have made allocation of network resources an issue of extreme importance. The problem is further exacerbated if we consider that, while the number of deployed devices is massive, the amount of network resources is still limited. In this thesis, we aim at providing a holistic approach for resource allocation in modern telecommunication networks. Specifically, we consider a network consisting of a *backhaul* interconnecting a *Radio Access Network* (RAN) that provides mobile users with wireless access to a *Core Network* (CN). The CN allows access to the Internet and enables end-to-end communications by routing users' data and calls. We take both energy and security aspects into account, by proposing a power-efficient and jamming-proof resource allocation scheme for the RAN. Furthermore, we consider the relevant and emerging case where the backhaul is shared among different *tenants*. Accordingly, we exploit Software Defined Networks (SDNs) and Network Function Virtualization (NFV) paradigms to provide dynamic and flexible network and service management in the multi-tenant backhaul and CN, respectively. We formulate the resource allocation problem through both centralized and distributed approaches, we discuss the existence and uniqueness of efficient resource allocation solutions and we provide distributed privacy-preserving algorithms that provably converge allow to the optimal resource allocation policy by exploiting only local or shared information.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER

# ONE

# INTRODUCTION

The last decades have been characterized by the development of new technologies and materials which have paved the way for significant evolutionary and revolutionary advances in the field of telecommunications. Nowadays, sharing computing resources among several network users is made possible by distributed and cloud computing [1–3]. Device-to-Device (D2D) communications are now feasible and can be exploited to improve network efficiency and performance in modern communication networks such as 4G and 5G networks [4–7]. Also, new materials such as graphene allow the development of communication devices, i.e., the so called nanomachines, whose size is in the nanometer range [8–10].

The technological evolution, together with the reduction of manufacturing costs of technological devices such as smart-phones, tablets and laptops, brought to the exponential diffusion of those devices. Accordingly, concepts such as the Internet of Things (IoT), where even everyday objects access the Internet and communicate with other devices, are feasible [11]. Furthermore, even though modern devices are small in size, they are able to perform complex computations, thus

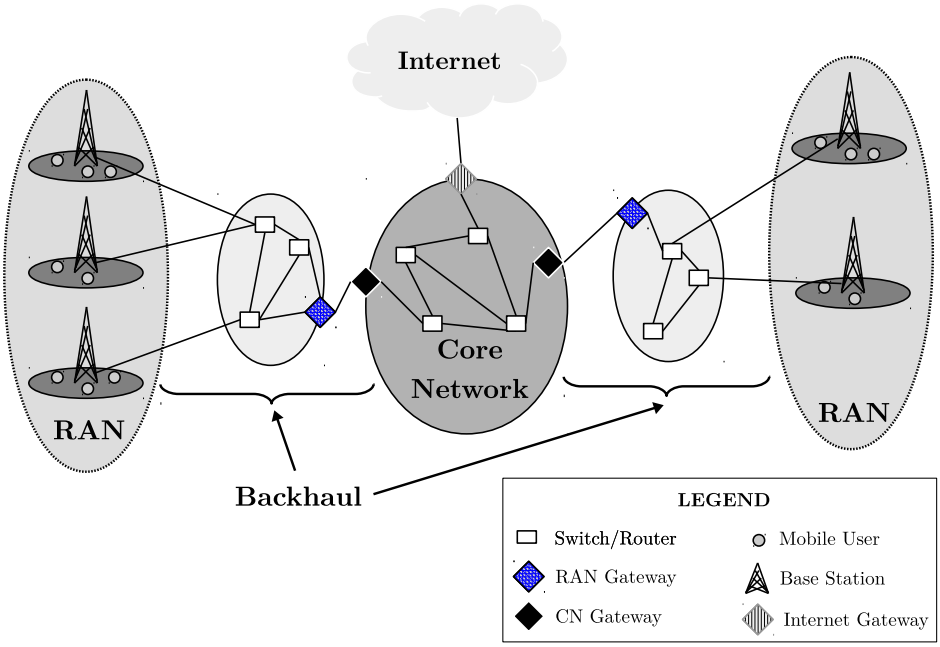



***Figure 1.1:*** *The considered network scenario.*

making possible to perform pervasive and ubiquitous computing.

For the above reasons, the deployment of such devices is already massive and it is expected to grow in the near future. On the one hand, the number of devices exponentially increases, on the other hand the amount of network resources (e.g., transmission power, spectrum, bandwidth in IP networks) to support device communications is limited and fixed [12–14].

Accordingly, the scarcity of network resource calls for efficient allocation and management of those resources. Even though a variety of efficient resource allocation schemes have already been designed in several wired and wireless scenarios [15–20], the development of new communication paradigms still leaves the resource allocation problem a challenging and open issue.

Modern communication networks generally consist of three ele-

ments: the *Radio Access Network (RAN)*, the *Backhaul* and the *Core Network (CN)* [5, 6, 21]. The RAN allows users to access the network through wireless connections provided by Access Points (APs) such as Base Stations (BSs). The CN allows end-to-end communications by routing users data and calls. Also, it provides network users with a variety of services and access to the Internet network and the cloud. Instead, the backhaul is the element in the network infrastructure that interconnects the RAN and the CN. A schematic computer network diagram is provided in Fig. 1.1. Also, Fig. 1.1 shows interconnections among the various network segments established by exploiting proper gateway nodes.

The core network is, in general, composed of high performance elements such as switches and routers that exploit optical technologies which are able to guarantee and support high-rate communications generated by BSs and network users. On the contrary, the backhaul is composed of hardware and links whose performance have been shown to be the bottle-neck of modern telecommunication networks [22, 23]. Furthermore, achievable performances are, in many scenarios, also affected by the fact that the backhaul is often shared among several *tenants*. This is the case of *multi-tenant networks* where the physical underlying network which composes the backhaul is divided in several logical slices that are assigned to one or more tenants that share the same physical resources such as routers, switches and links [24, 25].

In such a heterogeneous scenario where multiple tenants, wired and wireless technologies coexist, efficient resource allocation is of extreme importance. High-performance optical communications in the CN make the problem of resource allocation relevant but not critical. Instead, efficient allocation of network resources in both the RAN and the backhaul is fundamental. For example, due to the broadcast nature of wireless communications, the RAN is vulnerable to interferences, whether they are generated by legitimate users or by malicious

users, i.e. *jammers*. Accordingly, transmission policies that reduce the impact of interferences on network performance should be designed.

Instead, tenants are not expected to cooperate. Instead, they compete with each other to obtain as many users and network resources as possible to maximize their revenues and network performance. Recently, *Software Defined Networks (SDNs)* and *Network Function Virtualization (NFV)* have been exploited to provide flexible and dynamic network management and control. Also, these two technologies have enabled the softwarization of network devices and services, thus making possible to efficiently manage multi-tenant networks. However, how to optimally allocate network resources to provide efficient network management and control in softwarized networks is still an open issue.

To deal with resource allocation problems, two approaches are possible: *centralized* and *distributed*, respectively. The first approach requires a centralized entity that is charged to take decisions and to optimally allocate network resources. In general, such an approach implies that the centralized entity has full (or partial) access to the actual state of the network, e.g., topology of the network, number of connected users and their positions and channels conditions. However, to properly take into account all the above parameters, the centralized entity has to handle a large amount of variables, which implies that the decisional space is large. As a consequence, most centralized solutions result in combinatorial problems and require to test all the possible allocation combinations. This is the reason why most of the resource allocation problems that are addressed through centralized approaches are NP-Hard and are computationally inefficient. To overcome such an issue, greedy algorithms and heuristics approaches are often proposed. However, although the two latter approaches considerably reduce both the computational time and the complexity of the solution, they also introduce approximations which lead to sub-optimal solutions in most

cases.

On the contrary, distributed solutions allow each individual user to take decisions by using local information which, in most cases, is either already available, e.g., neighbors and user positions, or can be locally measured by each user in the network, e.g., signal-to-interference-and-noise ratio (SINR) and pilot signals. Accordingly, the decisional space of each user, i.e., the entity that takes decisions, is small and the number of variables to be handled is small. It follows that the computational complexity of such approaches is low and can often be computed in polynomial time.

To provide distributed solutions to resource allocation problems several approaches can be considered depending on the considered scenario. If users are allowed to cooperate with each other and exchange information, it is possible to allocate resources by exploiting well-known optimization algorithms such as message-passing [26, 27] and belief-propagation [28, 29]. However, such an assumption is either unrealistic or unfeasible in several scenarios where network users selfishly aim at maximizing their own performance without disclosing any private information to other users in the network. For example, spectrum auctions [30–32] and power allocation problems in Cognitive Radio Networks (CRNs) [12, 13, 33] are typical examples of network scenarios where users requests privacy-preserving solutions. Accordingly, a promising and rigorous approach to deal with such distributed non-cooperative scenarios, is Game Theory (GT). In game-theoretic approaches, a set of *players*, i.e., network users[1], take decisions from a *strategy set* to maximize a given *utility function*. Game-theoretic solutions can be derived for many resource allocation problems in communication networks. The desirable outcome of a game is a strategy profile such that each player maximizes its utility function and no

[1]In the following of this thesis,the terms network users, users and players are used interchangeably.

player has incentive in unilaterally deviating from.

In this thesis, we focus on the scenario where a set of mobile users access a multi-tenant backhaul through the RAN. We assume that mobile users are power constrained, i.e., their transmission power cannot exceed a given threshold. Also, we consider a malicious user that aim to disrupt wireless communications between mobile users and the RAN. Accordingly, the scope of this thesis is to design power-efficient and jamming-proof wireless communications in a non-cooperative multi-tenant backhaul where network and services management is performed by exploiting SDNs and NFV, respectively.

Accordingly, the main contributions of this thesis are as follows:

- A comprehensive analysis of the problem of resource allocation for power-efficient and jamming-proof communications in modern multi-tenant communication networks;

- Analysis and design of both centralized [34] and distributed [12, 13, 33, 35] resource allocation solutions for power-efficient and jamming-proof communications in the RAN. Also, we consider the challenging case of imperfect information w.r.t. system parameters such as jammer's position [34], channel state information (CSI) and SINR[12, 13, 33, 35];

- Investigation and design of distributed and privacy-preserving resource allocation solutions for network and service management of multi-tenant networks [36, 37];

- Discussion on the implementation aspects and issues of the proposed resource allocation solutions for the user-centric resource allocation in RANs with multi-tenant backhaul.

## 1.1 Structure of this Dissertation

This dissertation is organized in five chapters (including this introductory section) as follows:

Chapter 2 is devoted to a review of the existing literature which is relevant to our topic.

Chapter 3 proposes centralized and distributed resource allocation solutions for power-efficient and jamming-proof RAN. A resource allocation scheme that jointly provides power-efficient and jamming-proof communications is proposed. Also, the two cases where whether one of the two requirements is relaxed are discussed and efficient resource allocation policies are proposed in both cases.

Chapter 4, instead, presents game-theoretic SDN and NFV-based resource allocation schemes for managing network control in the multi-tenant backhaul and service provisioning in the CN. Existence and uniqueness of equilibrium points are discussed. Furthermore, distributed learning procedures which provably convergence to those equilibrium points are presented

Chapter 5 analyzes and discusses implementation aspects of the proposed resource allocation schemes for power-efficient and jamming-proof RAN over multi-tenant backhaul. Specifically, the computational complexity of proposed solutions is investigated and algorithmic implementations are proposed.

Finally, Chapter 6 summarizes the conclusions and proposes further works related to the presented subject.

## 1.2 Acknowledgements

A special thank goes to Prof. Sergio Palazzo, friend, teammate and advisor. Thanks for believing in me, for your suggestions and critics.

I would also like to thank my whole family that continuously pushes

me towards new objectives. A special thank goes to my grandma, rock-solid mother and woman.

A huge thank goes to Elisa, my better half, for loving and supporting me even though I was stressed and distant. Thanks for being part of my journey in this life.

I also would like to thank Profs L. Galluccio, G. Morabito and G. Schembra from my research group at the University of Catania. It was, and it is still, a great opportunity working with all of them.

Thanks to Prof. Eylem Ekici from the Ohio State University for his advisorship and his suggestions.

Finally, I would also like to thank my co-authors and Profs. P. Mertikopoulos, A. Moustakas, F. Martignon, L. Chen for sharing their knowledge with me.

Part of the results described in this dissertation has been supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM# (Grant agreement no. 318306).

CHAPTER

# TWO

# BACKGROUND AND RELATED WORK

Resource allocation is a challenging and timeless issue in both wired and wireless networks and several solutions to resource allocation problems have been proposed in the literature. However, quite often solutions to these problems result in scenario-specific approaches that make hard to develop a unifying framework to optimally allocate network resources in RANs with multi-tenant backhaul.

To achieve power-efficient and jamming-proof RAN in multi-tenant networks, we identify the following four issues that must be properly addressed.

- *Power Control/Allocation:* the power-efficiency requirement and the maximum transmission power constraint call for efficient and robust transmission policy design. Accordingly, to optimally schedule users' transmissions, and in line with a vast body of the literature, we identify power control and allocation as the most well-suited tool to achieve power-efficiency while satisfying power constraints;

- *Anti-jamming Mechanisms:* to achieve jamming-proof communications, anti-jamming mechanisms that either avoid the jam-

mer or counteract to its attacks must be designed. Also, imperfecteness or lack of information w.r.t. the jammer's system parameters (e.g., transmission power and position) should be considered;

- *Flexible and Dynamic Backhaul Management:* modern communication paradigms require flexibility of network management to adapt network behavior to dynamics of traffic which flows through the network. Recently, to achieve high flexibility, dynamism and fast deployment of routing policies, softwarization of network (e.g., such as in SDNs) has been proposed. It is expected that next-generation networks such as those devised in 5G technologies will be software-defined;

- *Service Management and Placement:* in the CN, services are provided to both tenants (e.g., Deep Packet Inspection (DPI) and Firewalls) and users (e.g., video encoding and streaming, caching and file sharing). Recently, a promising and successful approach consists in providing such services as Virtualized Network Functions (VNFs) hosted in virtual machine which run in data centers. Such an approach, known as NFV, allows to provide efficient and dynamic management of services. However, where to place those services is still a challenging and open problem.

In the literature, the four above problems have been thoroughly investigated. However, to the best of our knowledge, a holistic approach for a power-efficient and jamming-proof RAN in multi-tenant networks is still missing. Accordingly, this thesis is the first to devise and propose solutions for the considered problem. Since each one of the above issues deserves a separate analysis, in the following four subsections we separately provide fundamentals and analyze related work relevant to each considered topic.

Specifically, in Sections 2.1 and 2.2 we survey previous work relevant to power control/allocation and anti-jamming in wireless networks, respectively. Instead, in Sections 2.3 and 2.4 we consider related work in the field of SDNs and NFVs, respectively.

## 2.1 Power Control and Power Allocation

Ever since the early development stages of legacy wireless networks, power control has been an essential component of network design and operation, especially in decentralized environments where only local information is available at each mobile terminal [38]. As such, the introduction of fast and distributed power control algorithms (both closed- and open-loop) was one of the main improvements that were brought about in third generation CDMA-based cellular networks, in both single- and multi-carrier settings.

Controlling the transmitted power has two important purposes. The first is to minimize the interference of a given node to neighboring receivers in the RAN, an issue of critical importance in future and emerging wireless network paradigms where cells are deployed at a massive scale – for instance, as in the case of femto-cell networks [39]. Due to their close proximity, neighboring users may create significant interference to one another, so care must be taken to choose a power allocation profile that maximizes the users' transmission rate while limiting their overall transmit power – otherwise, the situation could rapidly degenerate to a cascade of power increases.

Second, power control reduces the users' overall transmitted power. Mobile terminals are generally energy-constrained (e.g. due to the limitations of their power source or because of the cost of power consumption), so inefficient power allocation can bring about unnecessary losses in performance. As a result, the problem that arises is to derive distributed power allocation policies that maximize the users' trans-

mission rate in energy-aware scenarios where transmission power also carries a commensurate cost. This objective is made more complicated by the fact that wireless users typically have conflicting interests and cannot be assumed to cooperate with each other for their collective benefit (in decentralized environments at least).

In view of the above, non-cooperative game theory has become an important tool to analyze the interactions between mobile users in wireless network scenarios where energy-efficient rate maximization is an issue – see e.g. [40–43] for applications to power control and [44–46] for power allocation problems.

Optimizing power allocation in such a way has been investigated in both single- [41, 42, 47] and multi-carrier scenarios [43, 48, 49]. In particular, the authors of [40, 41, 47] investigated the role of pricing as an effective mechanism to measure the cost of power consumption, thus leading to an energy-efficient formulation where users seek to maximize their transmission rate while keeping their transmit power in check (see also the very recent paper [49] where the authors consider the problem of maximizing the users' transmission rate per unit of transmitted power subject to minimum rate requirements).

Studies by the US Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA) have shown that only 15% to 85% of the licensed radio spectrum is used on average, leaving ample spectral voids that could be exploited via efficient spectrum management techniques [50, 51]. Accordingly, in this often unregulated context, the emerging paradigm of cognitive radio (CR) has attracted considerable interest as a promising way out of the spectrum gridlock [52–55].

At its most basic level, cognitive radio comprises a two-level hierarchy between wireless users induced by spectrum licencing: the network's licensed, Primary Users (PUCs) have purchased spectrum rights from the network operator (often in the form of contractual qual-

ity of service (QoS) guarantees), but they allow unlicenced Secondary Users (SUs) to access the spectrum provided that the induced co-channel interference (CCI) remains below a certain threshold [52, 54]. In such opportunistic and non-cooperative environments, power control and allocation has been identified as a fundamental tool to provide flexible transmit policies with minimal information exchange between mobile users and BSs. For example, the authors of [12, 40, 41, 56] investigated the role of pricing as an effective mechanism to desing efficient transmission policies and they provided an energy-efficient formulation of the problem where users seek to maximize their transmission rate while keeping their transmit power in check. To reach a stable equilibrium state in this setting, several distributed approaches have been proposed [57–62], based chiefly on reaction functions [41], Gauss-Seidel and Jacobi update algorithms [40], or learning [12, 13, 48].

That being said, the above works focus almost exclusively on RAN with static channel conditions where the benefits of power and interference control mechanisms are relatively easy to evaluate; by contrast, very little is known in the case where the channels vary with time (e.g., due to user mobility) or when the users' measurements are not accurate. In the presence of (fast) fading, channel gains are typically assumed to follow a stationary ergodic process, so the users' throughput depend crucially on the channel statistics. In this stochastic framework, the authors of [48] studied the problem of ergodic rate maximization in multi-carrier (MC) systems and derived an efficient power allocation algorithm that allows users to attain the system's capacity. More recently, [63] provided an efficient online learning algorithm for unilateral rate optimization in dynamic multi-carrier multiple-input and multiple-output (MIMO) cognitive radio systems. However, the above works do not provide any power-efficient and jamming-proof RAN as they fails when a jammer attacks the network.

## 2.2 Anti-jamming mechanisms

Wireless networks are especially prone to several attacks due to the shared and broadcast nature of the wireless medium. One of the most critical attacks is *jamming* [64, 65]. Jamming attacks can partially or totally disrupt ongoing communications at the RAN level of the network, and proper solutions have been proposed in various application scenarios [64, 66, 67]. *Continuous* jamming attacks can be really expensive for the jammer in terms of energy consumption as the transmission of jamming signals needs a significant, and constant, amount of power. To reduce energy consumption while achieving a high jamming effectiveness, *reactive* jamming is frequently used [68–71].

Reactive jamming attacks reach a high jamming efficiency and can even improve the energy-efficiency of the jammer in several application scenarios [72, 73]. Also, they can easily and efficiently be implemented on COTS hardware such as USRP radios [74–76]. But, more importantly, reactive jamming attacks are harder to detect due to the attack model, which allows jamming signal to be hidden behind transmission activities performed by legitimate users [69, 74, 77].

It is clear that under such hostile conditions, network performance can be significantly reduced. Accordingly, how to allocate network resources to provide proper anti-jamming techniques is an interesting topic and several solutions have been proposed in the literature. For example, spread-spectrum techniques are commonly used to avoid the jammer and its attacks [78–81]. However, to be effective, such techniques need to either share or establish a secret among network users. For this reason, such techniques cannot be applied in all wireless scenarios [82, 83].

In [82, 84, 85] a *trigger-identification* approach is presented. First, nodes whose transmission trigger the reactive jammer, i.e., the triggering nodes, are identified. Then, optimal routing paths are established,

which exclude triggering nodes from the routing process. However, such solutions are designed for large sensor networks where multi-hop communication is feasible. Such solutions fail in scenarios where the whole network is under attack and any node can potentially be a trigger node.

To detect transmission activities, the jammer has to first sense ongoing communications. Then, when an activity is detected the jammer starts its attack. Thus, there is a delay, i.e., activation time, between the detection and attack phases. All bits transmitted during the activation time escape jamming and can be exploited to establish communications under reactive jamming attacks [83].

*Timing channels* have been frequently exploited to support covert low rate [86], energy efficient [10, 87, 88] and undetectable communications [89]. Also, they have been proposed as anti-jamming solutions [35, 71, 90, 91]. As the reactive jammer does not attack when no transmission activities are performed by users, it is possible to encode the information to be transmitted in silences between consecutive packets. Accordingly, although transmitted packets can be completely damaged, the radio silence between consecutive packets can be modulated to convey data by mapping bit sequences and silence period durations. However, the above solutions neither aim to maximize network performance nor provide any minimum QoS service level to network users.

In [71] an analysis of energy consumption and effectiveness of a reactive jammer attack against timing channels is presented. Moreover, it is shown how a trade-off between energy consumption and jamming effectiveness can be sought. It is also demonstrated that continuous jamming can be very costly in terms of energy consumption.

Instead, in [90] Xu et al. propose an anti-jamming timing channel that exploits inter-arrival times between jammed packets to encode information to be transmitted, showing how timing channels are suit-

able to guarantee low rate communications even though a reactive jammer is disrupting transmitted packets. However, in [90] two constraining assumptions are made, that is, i) to perform an attack, the jammer first has to recognize the preamble of a packet, and ii) the jamming signal is transmitted as long as the jammer senses activity on the channel. In our proposed solution, the two above assumptions are relaxed.

As compared to the solutions proposed so far in the literature, one of the anti-jamming solution we propose in this thesis is comparable to that proposed in [92] by Anand et al. However, the main differences between our work and [92] can be summarized as follows:

- in [92] the target node focuses on deploying camouflaging resources (e.g., the number of auxiliary communications assisting the covert communication) to hide the underlying timing channel. In our work, instead, the target node establishes a timing channel that exploits the silence period between the end of an attack and the beginning of a subsequent packet transmission to counteract an ongoing jamming attack;

- in [92], only the Nash equilibrium (NE) is studied, whereas in our work we study both the NE and SE. Furthermore, we compare the achievable performance of each player, and find that the SE dominates the NE (i.e., both players improve their achieved utilities), thus allowing each player to improve its own utility;

- in our work, the target node is able to transmit covert information even if the jammer has successfully disrupted all the bits contained in a packet. On the other hand, the authors in [92] assume that the jamming attack is successful if the Signal-to-Interference ratio (SIR) of the attacked node measured at the receiver side is higher than the one of the target node. In our approach, instead, we do not make any assumption on the SIR

as, by exploiting our proposed timing channel implementation, it is possible to transmit some information even when the jammer has successfully corrupted each packet.

Power control has been recently proposed to overcome possible reactive jamming attacks [93, 94]. The intuition is that by controlling the transmission power of users, it is possible to let the received power at the jammer remain under the triggering threshold [94]. In [93], authors propose a joint *frequency hopping* (FH) and power control scheme to avoid reactive jamming attacks. The proposed solution consists in selecting users' transmission power such that the senders' power at the receiver side is higher than that of the jammer. However, the latter approach fails when dealing with systems such as the one we consider in this thesis where FH is not possible. In this thesis, we also propose a novel centralized anti-jamming solution which exploits power control to avoid jamming attacks while guaranteeing a minimum QoS service level over a finite time window. Also, our proposed solution differs from the already existing literature as we tackle the worst case scenario where no information w.r.t. the jammer position and attack strategy is available.

## 2.3 Network Management through Software Defined Networks (SDNs)

The proliferation of new services and applications in the Internet with different requirements in terms of availability, service quality and resilience, is making management of the backhaul a key challenge. In this evolving scenario, Telco Operators (TOs) which own the CN show increasing interest in "softwarizing" their networks, so making deployment, configuration, management and updating of network functions faster and easier, and thus achieving numerous advantages in terms

of both Capital Expenditure (CAPEX) and Operational Expenditure (OPEX).

In SDNs, control and data planes are decoupled. Network control and management are implemented in software, while the data/forwarding plane consists of an underlying physical network composed by several SDN-compliant switches and links. Two key enabler of this evolution which is gaining the interest from both the industry and the academic communities for the simplification in management processes, is Software Defined Networking [95–97].

Even though there are several ways to implement SDNs, OpenFlow [98] is the most popular implementation of SDNs as its specifications are easy to be implemented and they provide procedures to support dynamic resource allocation SDNs networks.

An important concept in SDNs is multitenancy. Multitenancy refers to the possibility for several users, i.e., the tenants, to share certain resources such as physical network elements and links and use them as they were the sole users of those resources. In the last years, increased attention has been paid to the application of the multitenancy concept in the networking domain. Even if most efforts have been focused on the application of multitenancy concepts to the datacenter domain [99, 100], in many other scenarios, such as the one we consider in this thesis, such concept can be exploited beneficially. Two relevant scenario, for example, are that of *virtual network operators* in which several companies sell network access services using the network infrastructure owned by a third party [101], and that of NFV [102].

In multi-tenant scenarios, the owner of the network infrastructure has two major needs: i) to maximize the quality of service experienced by its customers, that is, render its customers *satisfied*; ii) to maximize its revenue. In order to meet both of them, efficient resource management mechanisms should be considered [103–110].

To properly manage interactions among different tenants, Open-

Flow provides a FlowVisor [111], which is a high-level controller that is designed to act as a proxy between the physical network and multiple customers. By exploiting FlowVisor protocols, OpenFlow fully supports the multitenancy principle. In fact, FlowVisor and OpenFlow together allow the network owner to divide the network resources into *slices* and give full control of each slice to one customer that, to this purpose, runs a software program referred to as *Controller*. OpenFlow and FlowVisor ensure isolation between slices and therefore, each Controller can use its share of the network resources as if it was the sole controller doing it. In the following, we will identify the network owner with the FlowVisor and its customers with the corresponding Controllers.

So far, in OpenFlow implementations, construction of the slices has been static [112]. This is problematic if the interest of the Controllers for a certain network resource changes over time, for example due to dynamics in the generated traffic and/or to the occurrence of unexpected situations generating abnormal traffic load.

To provide optimal resource allocation schemes, several centralized approaches have been proposed in the literature [113–115]. However, such solutions often lead to NP-hard problems that can be implemented in real systems only if sub-optimal solutions and performance losses are tolerable [114, 115]. Also, centralized solutions require perfect information and cooperation among the centralized entity and the other agents in the network, i.e., the Controllers. Unfortunately, Controllers are likely not to share their private information, act selfishly and individually seek to maximize their own profits. Therefore, most of the above approaches cannot be used to solve resource allocation problems where privacy-preserving solutions which do not require cooperation among users are required.

Accordingly, and in line with a large body of literature on the design of efficient and distributed resource management techniques

spanning most networking domains [116–119], we consider *auctions* as the allocation instrument. In [120, 121], auctions have already been identified as promising resource allocation mechanisms in SDNs. However, [120, 121] do not provide any rigorous formulation and analysis of the problem. Instead, we are the first to theoretically formulate the resource allocation problem and solve it. In this perspective, we formulate a game-theoretic auction-based resource allocation mechanism where the FlowVisor acts as the *auctioneer* while the Controllers act as the *bidders*. Periodically the FlowVisor starts a new auction to sell a certain amount of resources of the backhaul and each Controller makes a bid. Controllers determine their bids based on their interest in the resources, i.e. the object of the auction. The FlowVisor then assigns each Controller a portion/share of the resources which is proportional to the submitted bid.

## 2.4 Services Management through Network Function Virtualization (NFV)

NFV, on the other hand, brings virtualization concepts from cloud computing to the network in order to let software-based network functions, also called virtualized network functions (VNFs) [122, 123], run on commodity hardware infrastructures in the CN or in the Internet network. However, as compared to purpose-built networking hardware or middle boxes devices, deterrents to this approach are the achievable performance and the scalability. Key elements for the design of these systems are resource allocation, and VNF orchestration.

Although similar design problems can be met in cloud computing scenarios [124–127], there are important differences stemming from the fact that servers in data centers are connected to each other through high-capacity and high-speed networks, so making the specifics of the

underlying network less important. On the contrary, in NFV deployments network constraints, such as bandwidth and latency, are of crucial importance [128]. Therefore, there is the need for considering NFV-specific aspects to allocate resources and optimize the position of VNFs, also taking into account that TOs prefer that VNFs run at the edge of the network, rather than in its core, so as to increase reliability and flexibility during the transient period of the first deployment [129, 130].

The choice of where running VNFs has to be made by accounting not only for the increased load in the nodes hosting the VNFs, but also for the latency experienced to reach these nodes, which can be different for each flow [131, 132]. Therefore, if for some flows it is possible to let VNFs run on remote public cloud providers, for those flows that have very stringent requirements the TO needs to create data centers close to the users, so increasing investments. Finally, another aspect that has to be considered is that, according to the NFV specifications, the above complex tasks of management, orchestration and resource allocation are in charge of only one entity, the *Orchestrator*, which therefore requires sophisticated algorithms that are able to simultaneously account for all the above needs [133–135].

In contrast with the above literature on NFV, we propose to distribute orchestration and resource allocation tasks, while limiting the work of the Orchestrator to coordinate and facilitate them. More specifically, some customers of the same TO, in the following referred to as *VNF Servers*, can host and execute VNFs to process network flows, so participating in the *VNF Market* as sellers. To this purpose, they decide the price to be applied to the users, and the bandwidth to request to the TO network to provide the service. Users, on the other hand, according to the price specified by each server, and the corresponding expected performance in terms of both experienced latency and received bandwidth, choose one server for each VNF. In this way

the task of associating each flow to a VNF Server is not decided by the Orchestrator, but autonomously and in a distributed way, as a consequence of the interaction between users and VNF Servers.

CHAPTER

# THREE

# POWER-EFFICIENT AND JAMMING-PROOF RADIO ACCESS NETWORKS

In this section, we focus our attention on how to provide a RAN which meets two important requirements. Specifically, the RAN has to be 1) power-efficient; and 2) jamming-proof.

We consider a set $\mathcal{N} = \{1, \ldots, N\}$ of users that are equipped with single-antenna transceivers and are power-constrained, i.e., their instantaneous transmission power is limited to a maximum power level $P$. We assume that users in $\mathcal{N}$ access the RAN through a set $\mathcal{K} = \{1, \ldots, K\}$ of non-interfering subcarriers[1]. In our model, we assume *Additive White Gaussian Noise* (AWGN) channels with channel gains defined as i.i.d. random variables. Also, we assume that a malicious user, i.e., the jammer, aims to disrupt ongoing communications between legitimate users and the RAN. More specifically, in our study we consider a reactive jamming attacker where the malicious user continuously monitors all communication channels searching for transmission activities and jams only those channels where the re-

[1]In the following we interchangeably use the terms channel, subcarrier and frequency to refer to a wireless channel.

ceived signal power is higher than a given threshold $P_{th}$. In our study, we assume that when a transmission activity is detected in a given slot, the jammer emits a jamming signal whose transmission power is $P_J$. For the sake of illustration, in the following we focus on the case where a set of users accesses the RAN through a single BS. However, note that our formulation can be extended to the more general case where multiple BSs have to be managed.

It worth noting that if no jammers are attacking the network, anti-jamming mechanisms are not required. Thus, providing a power-efficient resource allocation scheme is the only objective of the network provider. On the contrary, if jamming attacks are ongoing but no power constraints has to be considered, e.g., users are allowed to transmit with any transmission power level, then the resource allocation problem can be focused on only providing jamming-proof communications. Therefore, there could be some scenarios where either one of the two requirements can be relaxed.

Accordingly, in Section 3.1 we propose a resource allocation solution which jointly meets the above two requirements. Instead, for the sake of completeness in Sections 3.2 and 3.3 we provide resource allocation solutions when requirements 2) and 1) are relaxed, respectively.

## 3.1 A joint approach for a power-efficient and jamming-proof RAN

In this section, we consider a centralized resource allocation scheme to provide power-efficient and jamming-proof RAN. Specifically, we consider the case where the network operator has to schedule network users to maximize achievable network performance while guaranteeing a minimum performance level to each user. We also consider the worst case scenario where a reactive jammer is deployed within the RAN

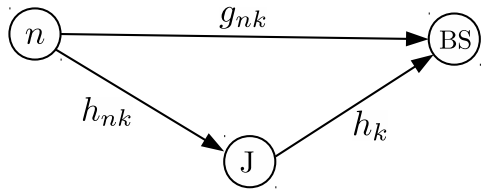



***Figure 3.1:*** *Considered channel model.*

coverage range. Each node is affected by its jamming activity and the transmission power of each user cannot exceed a given maximum threshold.

We show that the above problem can be modeled as a finite-horizon joint power control and user scheduling problem. Also, we prove that finding an optimal solution is NP-hard. We formulate the problem by exploiting techniques from Dynamic Programming (DP). The DP formulation allows us to show that the joint power control and user scheduling is a decomposable problem. That is, at each optimization step we can sequentially solve the power control and the user scheduling problems. We show that, under some conditions, it is possible to identify the optimal power control policy, i.e., conservative, exploratory or aggressive. To avoid the curse of dimensionality of the DP approach, we exploit state aggregation techniques. Finally, we study the complexity of the proposed solution.

### 3.1.1 System Model and Problem Formulation

We consider a multi-carrier slotted RAN where a set of users access the network and communicates with the BS through several non-interfering frequencies. We assume that users can transmit on at most one channel at a time and that a given slot on a given frequency can be assigned to only one user. In this study, we focus on the uplink scheduling problem. Let $h_{nk}$ be the channel gain coefficients between user $n$ and the jammer on channel $k$, while $h_k$ indicates the channel

gain coefficient between the jammer and the BS on channel $k$. Finally, $g_{nk}$ indicates the channel gain between user $n$ and the BS on channel $k$. For the sake of clarity, the considered channel model is depicted in Fig. 3.1.

In our study, we assume block fading, that is, channel gain coefficients remain constant for a fixed number of slots[2] before they change. Let $H$ be the number of slots where channel gain coefficients remain constant. Therefore, the optimal scheduling problem has to be periodically performed every $H$ slots. We refer to such time period as the *scheduling cycle* whose duration is $H$. Accordingly, $H$ is the finite-horizon of the optimization problem[3].

#### 3.1.1.1 Attack Model

We assume that the jammer's attack strategy is independent of channels and users; that is, the values of $P_{th}$ and $P_J$ are constant and equal for all $k \in \mathcal{K}$ and $n \in \mathcal{N}$.

To model the triggering mechanism that regulates the jammer, we define the *triggering function* $\alpha_{nk}(p) : \mathbb{R} \to \{0, 1\}$ for user $n$ transmitting on channel $k$. More specifically,

$$\alpha_{nk}(p) = \begin{cases} 1 & \text{if } ph_{nk} \geq P_{th} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $p$ is the generic transmission power for $n$ on channel $k$. Intuitively, according to (3.1), an attack is performed only when the received power at the jammer side ($ph_{nk}$) is greater than or equal to

[2]In this analysis, we will use the terms stage and slot interchangeably.

[3]Our model also applies to the scenario where a mobile jammer attacks the network. Since the jammer wants to be undetectable and unpredictable, it moves and changes its position every $H$ slots. It follows that channel gain coefficients vary in time and the network operator has to periodically find the optimal resource allocation policy every $H$ slots.

the triggering threshold $P_{th}$. Clearly, $\alpha_{nk}(0) = 0$. We consider the worst case scenario where $\alpha_{nk}(P) = 1$, i.e., all nodes are jammed by the jammer when they transmit with full power. However, the more general case where $\alpha_{nk}(P) = 0$ can be similarly treated by exploiting the same techniques presented in this investigation.

Let us consider the generic channel $k \in \mathcal{K}$ and let $p^m$ and $p^M$ be two feasible transmission power levels for a given user $n \in \mathcal{N}$ such that $0 \leq p^m < p^M \leq P$. We further assume that $\alpha_{nk}(p^m) = 0$ and $\alpha_{nk}(p^M) = 1$, that is, the system knows that when user $n$ transmits with power $p^m$ on channel $k$ it does not trigger the jammer, while transmitting with power $p^M$ on the same channel activates the jammer and consequently causes a decrease in the SINR[4]. Therefore, the probability of triggering the jammer when transmitting with power $p$ given the values of both $p^m$ and $p^M$ can be written as

$$\mathrm{F}_{nk}(p) = \Pr\{p h_{nk} \geq P_{th} | \pi_{nk}\} = \Pr\{P_{th}/h_{nk} \leq p | \pi_{nk}\} \tag{3.2}$$

where $\pi_{nk} = \left(p^m, p^M\right)$ is a tuple that represents the *history* (or knowledge) of the system. As shown in (3.2), the probability of triggering the jammer depends on the ratio $P_{th}/h_{nk}$ between the triggering threshold of the jammer and the channel gain coefficient between the jammer and the transmitter. Although in reality the position and the triggering threshold of the jammer are unknown and, thus, the exact value of the ratio $P_{th}/h_{nk}$ is unknown to the network operator, the information contained in the history $\pi_{nk} = \left(p^m, p^M\right)$ is still available. Note that given $\pi_{nk}$, the exact value of the ratio $P_{th}/h_{nk}$ can be any value in the range $(p^m, p^M]$. Therefore, to model such uncertainty on the knowledge of such parameters we assume that the ratio $P_{th}/h_{nk}$ is modeled as a *uniformly distributed* random variable over the interval

[4]Note that the existence of both $p^m$ and $p^M$ is always guaranteed by our assumptions and in Section 3.1.2 we provide proper mechanisms to identify the values of both parameters.

$(p^m, p^M]$. Accordingly, we rewrite (3.2) as follows:

$$\mathrm{F}_{nk}(p) = \begin{cases} 0 & \text{if } p \leq p^m \\ \frac{p-p^m}{p^M-p^m} & \text{if } p^m < p < p^M \\ 1 & \text{otherwise} \end{cases} \tag{3.3}$$

#### 3.1.1.2 Problem Formulation

As a consequence of the AWGN assumption, for any given user $n \in \mathcal{N}$ scheduled on channel $k \in \mathcal{K}$, the SINR received at the BS side is $\mathrm{SNR}_{nk}(p) = \frac{g_{nk}p}{\sigma^2+\alpha_{nk}(p)h_kP_J}$, where $p$ is the transmission power and $\sigma^2$ is the variance of the AWGN which we assume to be equal for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$. We assume that the channel gain coefficients $g_{nk}$ can be accurately estimated. Since the transmission power of each user is chosen by the centralized entity and all $g_{nk}$ are known, the product $h_kP_J$ can be accurately obtained by the BS by comparing the expected received signal with the actual received signal.

Let $\mathcal{H} = \{1, 2, \ldots, H\}$ be the set of slots in a scheduling cycle. Accordingly, we define the achievable rate $R_{nk}(p)$ at slot $j \in \mathcal{H}$ as follows:

$$R_{nk}(p(j)) = \log\left(1 + \frac{g_{nk}p(j)}{\sigma^2 + \alpha_{nk}(p(j))h_kP_J}\right) \tag{3.4}$$

Let $\theta_{nk}(j)$ and $p_{nk}(j)$ be the *allocation indicator* and *power control variable*, respectively. If user $n$ is allocated to channel $k$ at slot $j$, the allocation indicator is set to one, i.e., $\theta_{nk}(j) = 1$, otherwise it is set to zero, i.e., $\theta_{nk}(j) = 0$. Similarly, $p_{nk}(j)$ denotes the transmission power which must take values in the range $[0, P]$ due to the power constraint. Let $\boldsymbol{\theta}(j) = (\theta_{nk}(j))_{n,k}$ and $\boldsymbol{p}(j) = (p_{nk}(j))_{n,k}$ be the *scheduling policy* and *power control policy* at slot $j$, respectively. Clearly, if $\theta_{nk}(j) = 0$, then we set $p_{nk}(j) = 0$. Also, for any $n \in \mathcal{N}$ and $k \in \mathcal{K}$, let $\pi_{nk}(j) =$

$\left(p_{nk}^m(j), p_{nk}^M(j)\right)$ be the history up to slot $j$, where

$$\begin{aligned} p_{nk}^m(j) &= \max\{p_{nk}(l) \in \mathbf{p}(j) : \alpha_{nk}\left(p_{nk}(l)\right) = 0,\ l < j\} \\ p_{nk}^M(j) &= \min\{p_{nk}(l) \in \mathbf{p}(j) : \alpha_{nk}\left(p_{nk}(l)\right) = 1,\ l < j\} \end{aligned} \tag{3.5}$$

and $\mathbf{p}(j) = \left(p_{nk}(l)\right)_{n,k,l<j}$ is the set of all the power control decisions taken up to slot $j$. By assumption, we have $\pi_{nk}(1) = (0, P)$ for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$. Intuitively, at each slot the system keeps track of the reaction of the jammer to different policies chosen in the past.

To evaluate (3.4), we need to know the triggering function exactly. Unfortunately, the reaction of the jammer, i.e., the outcome of the triggering function $\alpha_{nk}(p(j))$, is known only at the end of each slot. Accordingly, from (3.1) (3.3) and (3.4), the expected achievable rate for user $n$ on channel $k$ is

$$\begin{aligned} E_\alpha\{R_{nk}(p)|\pi_{nk}(j)\} = &\frac{p - p_{nk}^m(j)}{p_{nk}^M(j) - p_{nk}^m(j)} \log\left(1 + \frac{g_{nk}p}{\sigma^2 + h_k P_J}\right) \\ &+ \left(1 - \frac{p - p_{nk}^m(j)}{p_{nk}^M(j) - p_{nk}^m(j)}\right) \log\left(1 + \frac{g_{nk}p}{\sigma^2}\right) \end{aligned} \tag{3.6}$$

where the expectation is taken w.r.t. the output of the triggering function given that the history of the system at slot $j$ is $\pi_{nk}(j)$.

We define the following finite-horizon joint power control and scheduling problem with minimum performance guarantee under jam-

ming attacks (Problem **A**).

$$(\mathbf{A}) : \max_{\boldsymbol{\theta},\mathbf{p}} \ \mathrm{E}_{\boldsymbol{\alpha}} \left\{ \sum_{j\in\mathcal{H}} \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} \theta_{nk}(j) R_{nk}(p_{nk}(j)) \right\}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} \theta_{nk}(j) \leq 1, \qquad \forall n \in \mathcal{N},\ j \in \mathcal{H} \tag{3.7}$$

$$\sum_{n\in\mathcal{N}} \theta_{nk}(j) \leq 1, \qquad \forall k \in \mathcal{K},\ j \in \mathcal{H} \tag{3.8}$$

$$\mathrm{E}_{\boldsymbol{\alpha}} \left\{ \sum_{j\in\mathcal{H}} \sum_{k\in\mathcal{K}} R_{nk}(p_{nk}(j)) \right\} \geq R_n^*, \qquad \forall n \in \mathcal{N} \tag{3.9}$$

$$\theta_{nk}(j) \in \{0,1\}, \qquad \forall n \in n,\ \forall k \in \mathcal{K},\ j \in \mathcal{H} \tag{3.10}$$

$$p_{nk}(j) \in [0,P], \qquad \forall n \in n,\ \forall k \in \mathcal{K},\ j \in \mathcal{H} \tag{3.11}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}(1), \boldsymbol{\theta}(2), \ldots, \boldsymbol{\theta}(H))$; $\boldsymbol{p} = (\boldsymbol{p}(1), \boldsymbol{p}(2), \ldots, \boldsymbol{p}(H))$ are the decision variables; $\boldsymbol{\alpha} = (\alpha_{nk}(p_{nk}(j)))_{n,k,j}$ is the set of all outcomes of the triggering function according to the actual power control policy $\boldsymbol{p}(j)$ at slot $j$; and $R_n^*$ is the minimum rate requirement of user $n$. In Problem (**A**), constraint (3.7) guarantees that at any given time a user can be allocated to only one slot. On the other hand, constraint (3.8) ensures that only one user can be allocated on a given slot, thus avoiding possible collisions and/or interferences among users. The minimum performance constraint is imposed by the non-linear constraint (3.9) which ensures that the expectation of the rate achieved by any user at the end of the optimization horizon is higher than or equal to the performance requirement $R_n^*$. Finally, constraints (3.10) and (3.11) guarantee the feasibility of the decision variables.

#### 3.1.1.3 Hardness of the Problem

(**A**) is a non-linear (concave) combinatorial optimization problem with both discrete (i.e., $\theta_{nk}(j)$) and continuous (i.e., $p_{nk}(j)$) decision vari-

ables. In this section, we prove that (**A**) is NP-hard by showing that the *Multiprocessor Scheduling* is polynomially reducible to a subproblem of (**A**). The multiprocessor scheduling is known to be NP-complete [136] and it is stated as follows: given $m$ processors, a deadline $D$ and a set $\mathcal{X}$ of jobs where each job $x_n \in \mathcal{X}$ has length $l_n$, is there a $m$-processor scheduling that schedules all jobs and meets the overall deadline $D$?

**Theorem 1** (NP-hardness)**.** *Problem* (***A***) *is NP-hard.*

*Proof.* Let $D = H$, $\mathcal{X} = \mathcal{N}$, $x_n = n$, $m = |\mathcal{K}|$. For each $n \in \mathcal{N}$ and $k \in \mathcal{K}$ we assume $g_n = g_{nk}$, i.e., users' channel gain coefficients are channel independent. Also, let us assume that the optimal power control policy for $n$, which we denote as $p_n$, is constant for all $k \in \mathcal{K}$ and $j \in \mathcal{H}$. Thus, the achievable rate of $n$ on each slot is constant and equal to $R_n(p_n)$. For any given minimum performance requirement $R_n^*$ we define the job's length $l_n = \frac{R_n^*}{R_n(p_n)}T$, where $T = 1$ is the duration of a single slot. We have built a reduction of the multiprocessor scheduling to an instance of a subproblem of (**A**). Thus, the above instance is NP-complete by reduction [136]. Also, since this reduction can be done in polynomial time, it follows that (**A**) is NP-hard and, unless P=NP, it cannot be solved in polynomial-time. □

### 3.1.2 Optimal Solution

In Theorem 1, we have shown that (**A**) is NP-hard. However, how to find an optimal solution still remains unsolved. In this section, we show that (**A**) is a dynamic problem. Moreover, we show that at each slot the joint power control and user scheduling problem is decomposable. That is, at each slot it is possible to separately solve the power control and the user scheduling problems.

#### 3.1.2.1 Problem Dynamism and Decomposability

In previous sections, we have shown that to maximize the achievable performance of the system, i.e., the overall transmission rate, the scheduler has to evaluate the expected achievable rate for all users at each slot. (3.6) shows that the expected achievable rate for a given user on a given channel depends on the history parameter $\pi_{nk}(j)$. In turn, (3.5) shows that the history $\pi_{nk}(j)$ depends on actions taken in the past. Therefore, the knowledge and the state of the system dynamically evolve at each slot. Intuitively, a DP approach is well-suited to model and solve the considered problem.

Another important issue is whether or not the problem is decomposable. To maximize the overall achievable rate of the RAN, the scheduling problem requires us to first estimate the achievable performance of each user. On the contrary, as shown in (3.6), to maximize the single-slot expected rate, the power control problem only needs to know the history parameter $\pi_{nk}(j)$ for all $n$ and $k$. Recall that $\pi_{nk}(j)$ does not depend on the scheduling policy at the actual slot, but only depends on the scheduling decisions taken in the past. Hence, at each slot, the power control problem can be solved independently of the actual scheduling policy. However, the user scheduling problem needs the output of the power control problem. Therefore, we first solve the power control problem and find the optimal transmission power level for each user on each channel. Then, we solve the scheduling problem.

#### 3.1.2.2 Optimal Power Control

To solve the single-slot power control problem, we must find the optimal transmission power level for all users on each channel. Constraints (3.7) and (3.8) imply that no collision may occur and we can separately solve the power control problem for any individual user.

For each slot $j$ and channel $k \in \mathcal{K}$, we define the single-user power

control problem (Problem **B**) as follows:

$$(\mathbf{B}) : \max \left\{ \max_{p \in \Pi_{nk}(j)} \mathrm{E}_{\alpha} \left\{ R_{nk}(p) | \pi_{nk}(j) \right\}, R_{nk}(P) \right\}$$

where $\Pi_{nk}(j) = [p_{nk}^{m}(j), p_{nk}^{M}(j)]$.

The challenges of (**B**) are twofold as: i) the problem evolves according to past choices; ii) the reaction of the jammer is observed only at the end of the slot. Therefore, we must consider all possible realizations of the triggering function $\alpha_{nk}(p)$. Let $p_{nk}^{*}(j)$ be defined as follows:

$$p_{nk}^{*}(j) = \arg\max_{p \in \Pi_{nk}(j)} E_{\alpha} \left\{ R_{nk}(p) | \pi_{nk}(j) \right\} \tag{3.12}$$

From (3.6), (3.12) can be rewritten as follows:

$$\begin{aligned} p_{nk}^{*}(j) = \arg\max_{p \in \Pi_{nk}(j)} & \frac{p - p_{nk}^{m}(j)}{\Delta_{nk}(j)} \log \left( 1 + \frac{g_{nk} p}{\sigma^2 + h_k P_J} \right) \\ & + \left( 1 - \frac{p - p_{nk}^{m}(j)}{\Delta_{nk}(j)} \right) \log \left( 1 + \frac{g_{nk} p}{\sigma^2} \right) \end{aligned}$$

where $\Delta_{nk}(j) = p_{nk}^{M}(j) - p_{nk}^{m}(j)$ is the *Lebesgue measure* of $\Pi_{nk}(j)$. In Proposition 1, we show that (**B**) admits a unique optimal solution.

**Proposition 1.** *Problem* (***B***) *always admits a unique solution* $p_{nk}^{\mathrm{OPT}}(j)$ *defined as*

$$p_{nk}^{\mathrm{OPT}}(j) = \begin{cases} p_{nk}^{*}(j) & \textit{if } \mathrm{E}_{\alpha}\{R_{nk}(p_{nk}^{*}(j))\} \geq R_{nk}(P) \\ P & \textit{otherwise} \end{cases} \tag{3.13}$$

To prove the uniqueness, it suffices to prove the strictly concavity of $\mathrm{E}_{\alpha} \left\{ R_{nk}(p) | \pi_{nk}(j) \right\}$. The second part can be proved by inspection.

Proposition 1 suggests that there are some scenarios where transmitting with the highest power, i.e., $P$, and triggering the jammer is the optimal power control solution. For example, if the jammer is in

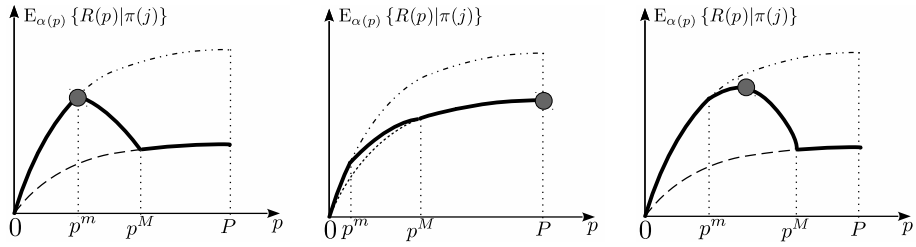



***Figure 3.2:*** *Comparison between different realizations of the expected achievable rate (solid lines) as a function of the transmission power p. Gray dots represent optimal transmission power policies: a) conservatory; b) aggressive; and c) exploratory. Dashed-dotted lines show the achievable rate when no attacks are performed ($\alpha(p) = 0$). Dashed lines show the achievable rate when the user is under attack ($\alpha(p) = 1$).*

proximity of a user but far away from the BS, it is reasonable to assume that even low transmission power levels can trigger the jammer. Therefore, to transmit with a low power level to avoid the jammer can be inefficient. It follows that transmitting with the highest power $P$ and triggering the jammer can be the only optimal policy. Clearly, $p_{nk}^{\mathrm{OPT}}(j)$ depends on the values of several parameters such as $P_J$, channel gain coefficients and $\pi_{nk}(j)$. Therefore, it is hard to know a priori the optimal policy chosen by the scheduler. However, to better understand the dynamics that regulate the power control problem, we define three different policies:

- *conservative*: is a policy where $p_{nk}^{\mathrm{OPT}}(j) = p_{nk}^{m}(j)$. Recall that $\alpha_{nk}(p_{nk}^{m}(j)) = 0$. Therefore, under such a policy (Fig. 3.2(a)), the scheduler chooses to avoid the jammer by choosing a safe strategy which ensures that the jammer will not be triggered;

- *aggressive*: is a policy where the optimal power control policy consists in transmitting with full power, i.e., $p_{nk}^{\mathrm{OPT}}(j) = P$.

In general, such a policy is optimal when jamming activities does not affect the performance of the system significantly (Fig. 3.2(b));

- *exploratory*: in this case, a transmitting power $p \in (p_{nk}^m(j), p_{nk}^M(j))$ is optimal (Fig. 3.2(c)). The scheduler decides to take the risk by exploring new transmitting power levels to which jammer's reaction is unknown.

The above policies are in line with the vast body of literature on the *exploration-exploitation* trade-off [137], where the decision maker has to choose between gathering new information by exploring new actions, or exploit the already explored actions whose system reactions are already known. When an exploratory policy is chosen, the scheduler takes the risk and explore new power control policies, even though such decision could trigger the jammer. When conservative and aggressive policies are chosen, the reaction of the jammer, together with the achievable performance under such policies, can be exactly predicted. Therefore, conservative and aggressive policies are exploitation decisions. In the rest of this study, we will refer to the exploration of new power control policies as the learning dynamics (or learning process) of the system.

Since the system is able to detect the presence or the absence of a jamming attack only when a slot ends, the achievable rate and the jammer's reaction to a given policy are known only at the end of each slot. Therefore, the history of the system is updated at the beginning of each slot according to the reaction of the jammer to decisions taken in the previous slot. Note that the history of the system is updated only when a user is scheduled on a given channel. That is, if $\sum_{k\in\mathcal{K}} \theta_{nk}(j) = 0$ for a given $j \in \mathcal{H}$ and $n \in \mathcal{N}$, we have that $\pi_{nk}(j+1) = \pi_{nk}(j)$. Instead, when a user is scheduled on a given channel and $j > 1$, the history of the system is updated according to

the history update dynamics in (3.14).

$$\pi_{nk}(j+1) = \begin{cases} \left(p_{nk}^{m}(j), p_{nk}^{OPT}(j)\right) & \text{if } \alpha_{nk}(p_{nk}^{OPT}(j)) = 1 \cup p_{nk}^{OPT}(j) \neq P \\ \left(p_{nk}^{m}(j), p_{nk}^{M}(j)\right) & \text{if } \alpha_{nk}(p_{nk}^{OPT}(j)) = 1 \cup p_{nk}^{OPT}(j) = P \\ \left(p_{nk}^{OPT}(j), p_{nk}^{M}(j)\right) & \text{otherwise} \end{cases} \tag{3.14}$$

where we recall that $\pi_{nk}(1) = (0, P)$.

In Proposition 2, we illustrate how sistem's learning dynamics impact the choice of the optimal power control policy.

**Proposition 2.** *Let $n \in \mathcal{N}$, $k \in \mathcal{K}$ and $j, l \in \mathcal{H}$. There exist $\delta_{nk}^{C}(j), \delta_{nk}^{A}(j) \in \mathbb{R}$ such that 1) if $\Delta_{nk}(j) \leq \delta_{nk}^{C}(j)$ a conservative policy is optimal, 2) if $\Delta_{nk}(j) \geq \delta_{nk}^{A}(j)$ an aggressive policy is optimal, and 3) if either conservative or aggressive policies are optimal at slot $j$, these policies will be optimal for any $l > j$.*

*Proof.* To prove the first part, let $r_{nk}(p)$ be the first order derivative of $\mathrm{E}_{\alpha}\{R_{nk}(p)\}$. A conservative policy is optimal if i) $r_{nk}(p_{nk}^{m}(j)) \leq 0$; and ii) $R_{nk}(p_{nk}^{m}(j)) \geq R_{nk}(P)$. i) holds if $\Delta_{nk}(\pi_{nk}(j)) \leq \tilde{\delta}_{nk}^{C}(j)$, and ii) holds if $\Delta_{nk}(\pi_{nk}(j)) \leq \hat{\delta}_{nk}^{C}(j)$ for some $\tilde{\delta}_{nk}^{C}(j), \hat{\delta}_{nk}^{C}(j) \in \mathbb{R}$. Therefore, a conservative policy is optimal at slot $j$ if $\Delta_{nk}(\pi_{nk}(j)) \leq \min\left\{\tilde{\delta}_{nk}^{C}(j), \hat{\delta}_{nk}^{C}(j)\right\}$. Similarly, we can prove that statement 2) holds as well. The last part of the proof is a direct consequence of the history update mechanism in (3.14). When $p_{nk}^{\mathrm{OPT}}(j)$ is conservative or aggressive, we have $\pi_{nk}(j+1) = \pi_{nk}(j)$. Thus, the same policy will still be optimal for all $l > j$, i.e., $\pi_{nk}(j) = \pi_{nk}(l) = \cdots = \pi_{nk}(H)$. □

Proposition 2 gives us an important insight on the learning dynamics of the system. We have shown that $\pi_{nk}(j) = \pi_{nk}(j')$ for all $j > j'$ if either conservative or aggressive policies are chosen at slot $j'$. That is, anytime that either conservative or aggressive policies are optimal for a given user on a given channel, the learning process for that user on that considered channel is stopped.

Although we proved that the optimal power control policy is always unique and we have shown the dynamics regulating the choice of the optimal policy, how to obtain the value of $p_{nk}^{\mathrm{OPT}}(j)$ still remains unsolved. A closed-form for $p_{nk}^{\mathrm{OPT}}(j)$, can be derived only by solving (3.12), which is not possible for our problem. To find the solution of (3.12), we exploit techniques from *stochastic approximation theory* and *exponential mappings*. For the sake of simplicity, in the following of this section we omit the subscripts $n$, $k$ and the slot index $j$. Let us define $\tilde{R}(p) = \mathrm{E}_{\alpha(p)}\{R(p)\}$, and let $r(p)$ denote the first derivative of $\tilde{R}(p)$ w.r.t. $p$. We also assume $p^m = 0$ and $p^M = P$. However, the more general case where $0 < p^m < p^M < P$ can be treated in a similar way[5].

The measure of the feasible power level set is $\Delta = p^M - p^m = P$. Finally, in (3.15) we consider the discrete-time stochastic approximation algorithm with exponential mappings

$$\begin{cases} z(i+1) = z(i) + \gamma_i r(p) \\ p(i+1) = \Delta \frac{e^{z(i+1)}}{1+e^{z(i+1)}} \end{cases} \tag{3.15}$$

where $i$ is the iteration index and $\gamma_i$ is a variable step-size [138]. It can be shown that the discrete-time stochastic approximation algorithm in (3.15) converges to the optimal solution of (3.12) for any possible feasible initial condition if $\sum_i \gamma_i^2 < \sum_i \gamma_i = +\infty$ (e.g., $\gamma = 1/i^\beta$, $\beta \in (0.5, 1]$). $p(i)$ is always bounded in $[0, \Delta]$. That is, the proposed algorithm in (3.15) always generates feasible transmission power updates[6].

[5]Note that we can define an auxiliary variable $p' = p - p^m$, $p' \in [0, p^M - p^m]$. Our results still hold as $r(p) = \frac{d\tilde{R}(p)}{dp} = \frac{d\tilde{R}(p'+p^m)}{dp'} = r(p')$.

[6]The same result also holds for the general case where $p(i) \in [p^m, p^M]$.

#### 3.1.2.3 Optimal User Scheduling

In this section, we define the finite-horizon DP-based algorithm to solve (**A**). To properly define the DP framework, we must take into account the uncertainty introduced by the jammer's behavior and its impact on the dynamics of the problem. Thus, in the language of DP, we define:

- *System state*: we define the system state at slot $j$ as the tuple $(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j))$, where $\boldsymbol{\pi}(j) = (\pi_{nk}(j))_{n,k}$ denotes the *history vector* at slot $j$. At each stage, $\pi_{nk}(j)$ is updated according to (3.14). On the other hand, $\boldsymbol{\rho}(j) = (\rho_n(j))_n$ denotes the *residual performance vector*. Each $\rho_n(j)$ specifies the remaining amount of performance that has to be allocated to user $n$ from slot $j$ to the horizon $H$ to satisfy its minimum performance request $R_n^*$. At each stage, $\rho_n(j)$ is updated as follows:

$$\rho_n(j+1) = \begin{cases} R_n^* & \text{if } j = 1 \\ \rho_n(j) - \sum_{k \in \mathcal{K}} \theta_{nk}(j) \mathrm{E}_\alpha \left\{ R_{nk}(p_{nk}^{\mathrm{OPT}}(j)) \right\} & \text{otherwise} \end{cases} \tag{3.16}$$

- *Action*: at each stage $j$, the actions of the scheduler are the optimal scheduling and power control policies, i.e., $\boldsymbol{\theta}(j)$ and $\mathbf{p}(j)$, respectively. For any scheduling policy $\theta_{nk}(j) \in \boldsymbol{\theta}(j)$, it must hold that $\theta_{nk}(j) = \{0, 1\}$. Instead, $\mathbf{p}(j)$ contains the optimal power control policies chosen when the history of the system is $\boldsymbol{\pi}(j)$. The generic element $p_{nk}(j) \in \mathbf{p}(j)$ is trivially defined as $p_{nk}(j) = 0$ if $\theta_{nk}(j) = 0$ and $p_{nk}(j) = p_{nk}^{\mathrm{OPT}}(j)$ if $\theta_{nk}(j) = 1$, where each $p_{nk}^{\mathrm{OPT}}(j)$ is given in (3.13);

- *Single Stage Reward*: we define the function $\Phi(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j), \boldsymbol{\theta}(j), \mathbf{p}(j), j)$ as the reward, i.e., the total transmission rate, that the system achieves at slot

$j$ when policy $(\boldsymbol{\theta}(j), \mathbf{p}(j))$ is chosen and the state is $(\boldsymbol{\rho}(j), \boldsymbol{\pi}(j))$. Therefore, $\Phi(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j), \boldsymbol{\theta}(j), \mathbf{p}(j), j) = \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} \theta_{nk}(j)\Gamma_{nk}(\boldsymbol{\pi}(j), \mathbf{p}(j), j)$,

where $\Gamma_{nk}(\boldsymbol{\pi}(j), \mathbf{p}(j), j) = \mathrm{E}_\alpha\left\{R_{nk}(p_{nk}^{OPT}(j))\right\}$ and $p_{nk}^{OPT}(j) \in \mathbf{p}(j)$ and it is calculated in (3.13). Thus, the dependence of $\Gamma_{nk}(\boldsymbol{\pi}(j), \mathbf{p}(j), j)$ from $\boldsymbol{\pi}(j)$ is implicit in the definition of $p_{nk}^{OPT}(j)$.

Now, we are ready to write the Bellman's equation:

$$J\left(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j), j\right) = \max_{\boldsymbol{\theta}(j), \mathbf{p}(j)} \Phi(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j), \boldsymbol{\theta}(j), \mathbf{p}(j), j)$$
$$+ \mathrm{E}\left\{J\left(\boldsymbol{\pi}(j+1), \boldsymbol{\rho}(j+1), j+1\right) | \boldsymbol{\pi}(j), \boldsymbol{\rho}(j)\right\} \quad (3.17)$$
$$\text{s.t.} \sum_{i=j}^{H} \sum_{k\in\mathcal{K}} \theta_{nk}(j)\mathrm{E}_\alpha\left\{R_{nk}\left(p_{nk}^{OPT}(i)\right)\right\} \geq \rho_n(j) \quad (3.18)$$
$$\sum_{k\in\mathcal{K}} \theta_{nk}(j) \leq 1, \qquad \forall n \in \mathcal{N} \quad (3.19)$$
$$\sum_{n\in\mathcal{N}} \theta_{nk}(j) \leq 1, \qquad \forall k \in \mathcal{K} \quad (3.20)$$
$$\theta_{nk}(j) \in \{0, 1\},\ p_{nk}(j) \in [0, P],\ \forall n \in n,\ \forall k \in \mathcal{K} \quad (3.21)$$

where we set $J(\boldsymbol{\pi}(j), \boldsymbol{\rho}(j), H+1) = 0$ for all $\boldsymbol{\pi}(j)$ and $\boldsymbol{\rho}(j)$.

Each stage of the Bellman's equation consists in a (binary) integer linear programming (ILP) problem. ILP problems are known to be NP-Complete and their exact solution can be computed through standard *Branch-and-Bound* methods. Finally, by using backwards induction we solve the Bellman's equation and find the optimal solution to (**B**).

### 3.1.3 Discretization of the Problem

In the previous sections, we have shown that DP can solve our considered problem. However, it is well known that DP suffers from the "curse of dimensionality" [139]. That is, when the state space and the number of variables increase, the number of possible combinations that we need to solve considerably increases. As an example, the power control variables $\boldsymbol{p}(j)$ are defined over the continuous set $[0, P]$. It follows that the number of possible combinations of both the system state and the feasible actions is infinite.

DP can theoretically still provide an optimal solution to such continuous space problem. However, from a practical point of view, it is unrealistic to implement the Bellman's equation on discrete-time systems. Therefore, we now propose a discretization of the continuous problem by exploiting *state aggregation* techniques [139].

State aggregation allows to aggregate one or more spaces of the original problem to create several lower dimension abstract spaces. For example, we can aggregate spaces by quantizing the power control action space to create a discretized version of it. Furthermore, both the history $\boldsymbol{\pi}(j)$ and the residual performance vector $\boldsymbol{\rho}(j)$ depend on $\boldsymbol{p}(j)$, which contributes to further increase the dimension of the problem.

In the following, we show that by discretizing both the power control variable and the residual performance vector, it is possible to significantly reduce the complexity of the problem while guaranteeing users's QoS requirement.

#### 3.1.3.1 State Aggregation Approach

Let $\xi_p$ be the power quantization step. Without losing in generality, we assume that the power quantization step is chosen such that $\xi_p$ is a divisor for $P$.

Let $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ be the ceiling and floor operators, respectively. For any $p \in [0, P]$, let $\overline{p} = \left\lceil \frac{p}{\xi_p} \right\rceil \xi_p$ and $\underline{p} = \left\lfloor \frac{p}{\xi_p} \right\rfloor \xi_p$ be the higher and lower quantized power level values, respectively.

In the discretized version of the problem, we use our quantization-based state aggregation to discretize $p^*_{nk}(j)$. That is, we calculate its quantized equivalents $\overline{p^*_{nk}}$ and $\underline{p^*_{nk}}$. Let $\hat{p}^*_{nk}(j)$ be defined as follows:

$$\hat{p}^*_{nk}(j) = \begin{cases} p^*_{nk}(j) & \text{if } \overline{p^*_{nk}} = \underline{p^*_{nk}} \\ \arg\max_{p=\{\overline{p^*_{nk}}, \underline{p^*_{nk}}\}} \mathrm{E}_{\alpha(p)} \left\{ R_{nk}(p) \right\} & \text{otherwise} \end{cases}$$

Thus, $\hat{p}^{\mathrm{OPT}}_{nk}(j) = \arg\max_{p=\{\hat{p}^*_{nk}(j), P\}} \mathrm{E}_{\alpha(p)} \left\{ R_{nk}(p) \right\}$ is the solution to the discretized version of (3.13) [7].

Let $\mathbf{p}(j)$ and $\boldsymbol{\theta}(j)$ be the optimal power control and scheduling policy at stage $j$, respectively. Similarly to the continuous space problem, for any $p_{nk} \in \mathbf{p}(j)$ we have $p_{nk} = \hat{p}^{\mathrm{OPT}}_{nk}(j)$ iff $\theta_{nk}(j) = 1$. Otherwise, $\hat{p}^{\mathrm{OPT}}_{nk}(j) = 0$. At each stage, the history of the system $\pi_{nk}(j)$ is updated according to (3.14).

So far, we have discretized the state of power control variable. However, from (3.4) and (3.16), it is clear that both the achievable rate and the residual performance vector have a continuous state space.

Let $\xi_r$ be the performance quantization step. We assume that the network operator forces each user to submit a minimum performance requirement, $R^*_n$, such that the latter is an integer multiple of $\xi_r$. To overcome the high-dimensionality caused by the definition of the residual performance vector, we modify the update dynamic of $\boldsymbol{\rho}(j)$ as follows:

$$\rho_n(j+1) = \begin{cases} R^*_n & \text{if } j = 1 \\ \left\lfloor \frac{\rho_n(j)}{\xi_r} - \frac{\sum_{k \in \mathcal{K}} \theta_{nk} \mathrm{E}_{\boldsymbol{\alpha}} \{ R_{nk}(p_{nk}) \}}{\xi_r} \right\rfloor \xi_r & \text{otherwise} \end{cases} \tag{3.22}$$

[7] Note that even though $\hat{p}^*_{nk}(j)$ is optimal for the discretized version of (3.12), it is sub-optimal for the continuous space problem, unless that $p^*_{nk}(j) = \hat{p}^*_{nk}(j)$.

***Table 3.1:*** *Simulation Setting used in Section 3.1.4*

| Parameter | Value |
|---|---|
| Carrier frequency | $f_c = 2.4\,\text{GHz}$ |
| Channel bandwidth | $B = 10.93\,\text{KHz}$ |
| Noise spectral density | $\sigma^2 = 584\,\mu\text{W}$ |
| Maximum transmission power of users | $P = 0.6\,\text{W}$ |
| Triggering threshold | $P_{th} = 0.5\,\mu\text{W}$ |
| Edge of the simulated square area | $L = 200\,\text{m}$ |
| Horizon duration | $H = 10$ |

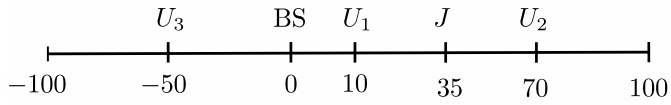



***Figure 3.3:*** *Topology of the simulated* 1*-dimensional scenario.*

where $p_{nk} \in \mathbf{p}(j)$.

Let $N_p$ and $N_r$ denote the maximum number of power and transmission rate quantized levels, respectively. Trivially, $N_p = (P/\xi_p + 1)$ and $N_r = (R^{max}/\xi_r + 1)$, where $R^{max} = \max\left\{R_1^*, R_2^*, \ldots, R_{|\mathcal{N}|}^*\right\}$.

Now, we apply the Bellman's equation to the discretized problem and find its optimal solution.

### 3.1.4 Numerical Analysis

In this section, we evaluate the achievable performance of the proposed solution through simulation results. We consider $N = 3$ legitimate users that access the RAN. Unless otherwise stated, we consider $K = 2$ channels whose gain coefficients $g_{nk}$, $h_{nk}$ and $h_k$ are generated according to the path-loss model. As depicted in Fig. 3.3, we consider a 1-dimensional scenario where users (i.e., $U_1$, $U_2$, $U_3$), the BS and the

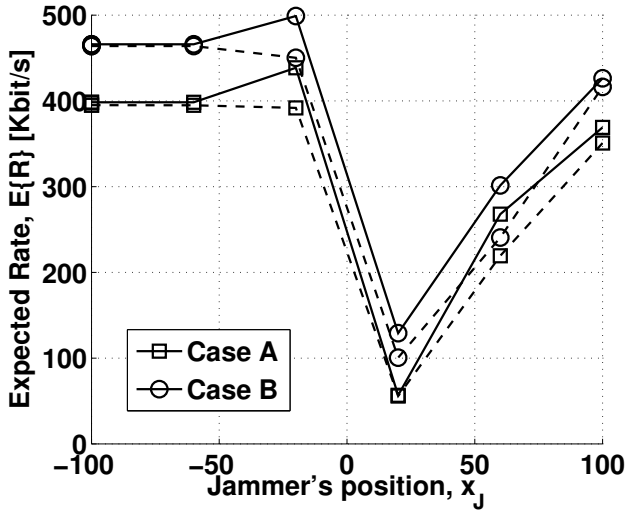



***Figure 3.4:*** *Expected rate of the system as a function of the position of the jammer $x_J$ for different QoS requirement settings and values of $P_J$ (Solid lines: $P_J = 0.5\,W$; Dashed lines: $P_J = 1\,W$).*

jammer (i.e., $J$) are located along the same axis[8]. Other relevant simulation parameters are reported in Table 3.1. Finally, unless otherwise stated, we assume $P_J = 0.6\,W$.

To investigate the impact of the position $x_J$ of the jammer on the achievable performance of the network at the RAN level, in Fig. 3.4 we evaluate the expected transmission rate as a function of $x_J$ under two different minimum QoS requirements. Specifically, we consider the case where all users request a same minimum QoS level $R_n^* = 10.93\,\text{Kbit/s}$ (Case A), and the case where no requirements are submitted by users (Case B). Fig. 3.4 shows that when the jammer is at the border of the considered scenario, its attacks have limited effect on RAN performance. Instead, when the jammer approaches the BS and the users, the achievable rate of the RAN considerably decreases.

[8]This is just an illustrative example. However, our approach is independent of the actual RAN topology.

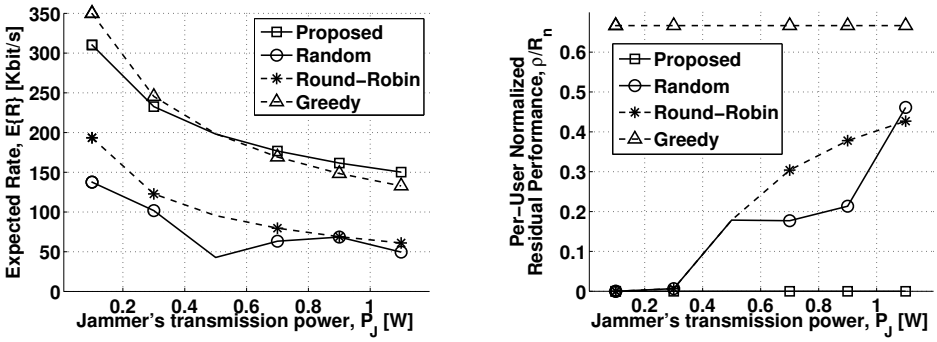
Also, when no QoS constraints are considered and rate maximization is the only objective of the network operator, system performance are higher than those achieved in Case A.

Finally, in Figs. 3.5(a) and 3.5(b) we compare the performance of the system when different scheduling policies are considered. Specifically, we compare our proposed solution to random, round-robin and greedy scheduling policies.

Fig. 3.5(a) shows that our proposed solution reaches high transmission rates and it outperforms random and round robin policies. The achievable performance of the RAN under the proposed solution and a greedy policy are comparable. However, in Fig. 3.5(b) we plot the per-user normalized residual performance variable $\bar{\rho}$ at the horizon $H$ defined as $\bar{\rho} = \frac{1}{N}\sum_{n\in\mathcal{N}} \frac{\rho_n(H)}{R_n^*}$. $\bar{\rho}$ represents the QoS-gap, i.e., the per-user amount of performance that has not been provided to users at the end of the scheduling cycle. Thus, the desirable value is $\bar{\rho} = 0$, while $\bar{\rho} > 0$ indicates unfeasible solutions. Our solution is the only one that guarantees $\bar{\rho} = 0$, i.e., all minimum QoS requirements are satisfied. Therefore, even though greedy policies allow to achieve high performance, they do not guarantee minimum QoS levels. Fig. 3.5(b) also shows that $\bar{\rho}$ for greedy policies is constant and high. On the contrary, the value of $\bar{\rho}$ under random and round robin policies is low for small values of the transmission power of the jammer, but it increases when the value of $P_J$ increases as well.

## 3.2 Power-efficient RAN access

In this section, we consider the problem of efficient power allocation in the RAN. To this purpose, we consider the case where users unilaterally maximize their transmission rate under power and pricing constraints. We provide a game-theoretic formulation for the above problem and we show that the resulting game admits a unique equi-




(a) (b)

***Figure 3.5:*** *a) Expected rate of the system under different scheduling policies as a function of $P_J$; b) Average per-user normalized residual performance at the end of the joint power control and user scheduling cycle as a function of $P_J$.*

librium for almost every realization of the system's channels. We then propose a distributed procedure which converges to equilibrium very rapidly using only local CSI and SINR measurements. Importantly, we are able to show that the algorithm retains its convergence properties even in the presence of imperfect measurements.

### 3.2.1 System Model and Problem Formulation

As we have already done in Section 3.1, in we focus our analysis on the uplink case. Also, we consider that an OFDM scheme is employed and users can simultaneously transmit on the available orthogonal frequencies. Under the AWGN assumption, the aggregate received signal $y_k$ over the $k$-th subcarrier is given by the familiar signal model:

$$y_k = \sum\nolimits_{n\in\mathcal{N}} h_{nk}x_{nk} + z_k \tag{3.23}$$

where $x_{nk} \in \mathbb{C}$ denotes the transmitted signal of user $n \in \mathcal{N}$ over subcarrier $k \in \mathcal{K}$, $h_{nk} \in \mathbb{C}$ is the corresponding channel coefficient

(assumed fixed for the duration of the transmission) and $z_k \in \mathbb{C}$ is the noise in the channel, including thermal, atmospheric and other ambient effects – and modeled as a zero-mean Gaussian vector $z_k \sim \mathcal{CN}(0, \sigma_k^2)$ with non-singular covariance.

In this context, the average transmit power of user $n$ on subcarrier $k$ is

$$p_{nk} = \mathbb{E}\left[|x_{nk}|^2\right] \tag{3.24}$$

and we will be assuming that each user's *total* transmit power $p_n = \mathbb{E}[\mathbf{x}_n^\dagger \mathbf{x}_n] = \sum_k p_{nk}$ satisfies the constraint:

$$p_n = \sum\nolimits_{k\in\mathcal{K}} p_{nk} \leq P_n \tag{3.25}$$

where $P_n$ denotes the maximum transmit power of user $n \in \mathcal{N}$. Accordingly, the set of admissible power allocation vectors for user $n$ will be

$$\mathcal{X}_n = \left\{\mathbf{p}_n \in \mathbb{R}^{\mathcal{K}} : p_{nk} \geq 0 \text{ and } \textstyle\sum_{k\in\mathcal{K}} p_{nk} \leq P_n\right\} \tag{3.26}$$

and the system's *state space*,i.e., the space of all admissible power allocation profiles $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_N)$, will be denoted by $\mathcal{X} = \prod_n \mathcal{X}_n$.

On that account, each user's achievable transmission rate will depend on his individual SINR

$$\mathrm{SINR}_{nk}(\mathbf{p}) = \frac{g_{nk}p_{nk}}{\sigma_k^2 + \sum_{\ell\neq n} g_{\ell k}p_{\ell k}} \tag{3.27}$$

where $g_{nk} = |h_{nk}|^2$ denotes the channel gain coefficient for user $n$ over the $k$-th subcarrier. Thus, in the single user decoding (SUD) regime – where interference by other users is treated as (possibly colored) noise – the maximum information transmission rate (achievable with random Gaussian codes) will be:

$$R_n(\mathbf{p}) = \sum\nolimits_{k\in\mathcal{K}} \log\left(1 + \mathrm{SINR}_{nk}(\mathbf{p})\right) \tag{3.28}$$

Given the form of this objective, each user will saturate the power constraint (3.25) and transmit with maximum possible power in order to maximize his throughput. In practical scenarios however, power consumption carries a commensurate cost, so we will instead consider the energy-aware utility model

$$\mathcal{U}_n(\mathbf{p}) = R_n(\mathbf{p}) - \Gamma_n(p_n) \tag{3.29}$$

where $p_n = \mathbb{E}[\mathbf{x}_n^\dagger \mathbf{x}_n] = \sum_{k\in\mathcal{K}} p_{nk}$ denotes the user's total transmit power and $\Gamma_n \colon [0, P_n] \to \mathbb{R}_+$ is a user-specific cost function measuring the impact of power consumption.

The utility/cost model above admits several interpretations, depending on one's point of view. Perhaps the most straightforward one is that of $c_n$ representing the monetary cost to be paid to the network operator to access the RAN, or the effective cost of power consumption (whether the cost is paid up front or postponed to the moment where the battery of the wireless device will need to be recharged). Alternatively, from the viewpoint of energy efficiency, the cost function $c_n$ could represent the user's adversity to transmit with higher power when not absolutely necessary. To keep things as general as possible, we will only consider $c_n$ as a generic "price" function and assume that it is convex and increasing in $p_n$ (in tune with standard economic assumptions).

With all this in mind, unilateral utility maximization leads to a non-cooperative game $\mathcal{G} = \mathcal{G}(\mathcal{N}, \mathcal{K}, \mathcal{U})$ for cost-efficient power allocation defined as follows:

1. The set of *players* of $\mathcal{G}$ comprises the set of wireless transmitters $\mathcal{N} = \{1, \ldots, N\}$.

2. Each player's set of *strategies* consists of the corresponding feasible power allocation profiles $\mathbf{p}_n \in \mathcal{X}_n = \{\mathbf{p}_n \in \mathbb{R}^{\mathcal{K}} : p_{nk} \geq 0 \text{ and } \sum_{k\in\mathcal{K}} p_{nk} \leq P_n\}$.

3. Each player's *utility* $\mathcal{U}_n \colon \mathcal{X} \to \mathbb{R}$ is given by (3.29).

We will thus say that a power allocation profile $\mathbf{p} \in \mathcal{X}$ is a NE of $\mathcal{G}$ when

$$\mathcal{U}_n(\mathbf{p}_n; \mathbf{p}_{-n}) \geq \mathcal{U}_n(\mathbf{p}'_n; \mathbf{p}_{-n}) \tag{3.30}$$

for all $\mathbf{p}'_n \in \mathcal{X}_n$ and for all $\mathbf{p}_{-n} \in \mathcal{X}_{-n} \equiv \prod_{\ell \neq n} \mathcal{X}_\ell$.

As the next lemma shows, an important property of the game $\mathcal{G}$ is that the players' objectives are aligned along a (concave) *potential function* (in the sense of [140]):

**Lemma 1.** *The (concave) function*

$$V(\mathbf{p}) = \sum\nolimits_{k \in \mathcal{K}} \log\left(1 + \sum\nolimits_n g_{nk} p_{nk} / \sigma_k^2\right) - \sum\nolimits_{n \in \mathcal{N}} \Gamma_n(p_n) \tag{3.31}$$

*is a potential function for $\mathcal{G}$; more precisely:*

$$\mathcal{U}_n(\mathbf{p}_n; \mathbf{p}_{-n}) - \mathcal{U}_n(\mathbf{p}'_n; \mathbf{p}_{-n}) = V(\mathbf{p}_n; \mathbf{p}_{-n}) - V(\mathbf{p}'_n; \mathbf{p}_{-n}) \tag{3.32}$$

*for all $\mathbf{p}_n \in \mathcal{X}_n$, $\mathbf{p}_{-n} \in \mathcal{X}_{-n}$, and for all $n \in \mathcal{N}$.*

*Sketch of proof.* The claim follows by carrying out the calculation at the right-hand side of (3.32). □

By exploiting the game's potential property, it is easy to see that the game's set of equilibria coincides with the set of maximizers of $V$ [141]; as such, we are led to the (nonlinear) concave maximization problem:

$$\begin{aligned} &\text{maximize} && V(\mathbf{p}_1, \ldots, \mathbf{p}_n) \\ &\text{subject to} && p_{nk} \geq 0 \text{ and } \sum\nolimits_{k \in \mathcal{K}} p_{nk} \leq P_n \end{aligned} \tag{3.33}$$

**Remark 1.** *The first term of the potential function $V$ is simply the system's sum rate under successive interference cancellation (SIC). As a result, maximizing $V$ over the set of feasible power allocation profiles $\mathbf{p} \in \mathcal{X}$ is equivalent to maximizing the users' aggregate utility (sum*

*rate minus aggregate cost) in a* centralized *environment where one can apply more sophisticated successive interference cancellation (SIC) techniques.*

## 3.2.2 Nash Equilibrium Analysis

### 3.2.2.1 Existence and Uniqueness of the Nash Equilibrium

With Lemma 1 at hand, establishing the existence of NE for $\mathcal{G}$ is trivial; in fact, since the game's potential is concave, it follows that the set of NE of the game is a convex subset of $\mathcal{X}$ [141, 142]. As it turns out, this convex set is almost surely a singleton:

**Proposition 3.** *The cost-efficient power allocation game $\mathcal{G}$ admits a unique NE for almost every realization of the channel coefficients $h_{nk}$.*

*Proof.* First, note that $V$ is not *strictly* concave: $V(\mathbf{p}) = V(\mathbf{p}')$ whenever $\sum_n g_{nk}p_{nk}/\sigma_k^2 = \sum_n g_{nk}p'_{nk}/\sigma_k^2$ and $p_n = p'_n$. These liner relations define a convex subset of maximizers of $V$ which lie at the intersection of $\mathcal{X}$ with an affine space of "degenerate" directions along which $V$ is constant. By using a graph-theoretic method introduced in [143], it can be shown that the quilibrium point lies in the interior of a face of $\mathcal{X}$ with dimension at most $K-1$. Since the Nash set of $\mathcal{G}$ is a convex polytope of dimension $NK-K$, we conclude that any NE lies at the intersection of a $g$-independent $(K-1)$-dimensional and a $g$-dependent $(NK-K)$-dimensional subspace of $\mathbb{R}^{NK}$. However, since $NK-K+K-1 < NK$, the intersection of these subspaces is trivial on a set of full measure with respect to the choice of $g$, implying that there exists a unique NE. □

### 3.2.2.2 Convergence to the Nash Equilibrium

The fact that the cost-efficient power allocation game $\mathcal{G}$ admits a unique equilibrium for almost every channel realization is significant

from the point of view of managing the system because it guarantees a unique stable solution. That said, it is far from clear how the system's users could actually reach this equilibrium state, so our goal in this section will be to provide a distributed, adaptive learning mechanism that can be employed by the system's users in order to reach this stable state.

In the absence of power considerations, [48] examined this problem by means of a continuous-time learning scheme based on the *replicator dynamics* of Evolutionary Game Theory (EGT) [144] driven by the users' so-called marginal utility functions $\frac{\partial R_n}{\partial p_{nk}}$. Unfortunately however, this approach cannot be applied in our case because replicator-driven techniques require the problem's state space to be a product of simplices – which, in our case, would amount to players saturating the total power constraint (3.29) by default.

To overcome this issue, let $p_{n,0}$ denote the *unused* power of user $n$, i.e.

$$p_{n,0} = P_n - p_n = P_n - \sum\nolimits_{k\in\mathcal{K}} p_{nk} \tag{3.34}$$

so that

$$\sum\nolimits_{\alpha=0}^{K} p_{n\alpha} = P_n \tag{3.35}$$

for all $n \in \mathcal{N}$. Accordingly, letting the index "0" denote a virtual, "unused" channel and writing $\mathcal{K}_0 = \mathcal{K} \bigcup \{0\} = \{0, 1, \ldots, K\}$ for the system's artificially augmented channel set, the concave problem (3.33) may be reformulated as:

$$\begin{aligned} &\text{maximize} \quad V_0(\mathbf{p}_1, \ldots, \mathbf{p}_N) \\ &\text{subject to} \quad \mathbf{p}_n \in \Delta_n \equiv \left\{ \mathbf{p}_n \in \mathbb{R}^{\mathcal{K}_0} : p_{n\alpha} \geq 0 \text{ and } \sum\nolimits_{\alpha\in\mathcal{K}_0} p_{n\alpha} = P_n \right\} \end{aligned} \tag{3.36}$$

where now $\mathbf{p}_n = (p_{n,0}, p_{n,1}, \ldots, p_{n,K})$ and

$$V_0(\mathbf{p}_1, \ldots, \mathbf{p}_K) = \sum\nolimits_{k=1}^{K} \log\left(1 + \sum\nolimits_{n\in\mathcal{N}} g_{nk}p_{nk}/\sigma_k^2\right) - \sum\nolimits_{n\in\mathcal{N}} \Gamma_n(P_n - p_{n,0}) \tag{3.37}$$

Hence, drawing on the analysis of [48] for power allocation problems with fixed transmit power $p_n$, we will consider here the *marginal utilities*:

$$v_{n\alpha} = \frac{\partial \mathcal{U}_n}{\partial p_{n\alpha}} = \begin{cases} \Gamma_n'(P_n - p_{n,0}) & \text{if } \alpha = 0 \\ g_{nk}/\left(\sigma_k^2 + \sum_\ell g_{\ell k}p_{\ell k}\right) & \text{otherwise} \end{cases} \tag{3.38}$$

Importantly, these marginal utilities can be calculated by each user with only local information at hand (such as SINR measurements). Indeed, $v_{n,0}$ only depends on the user's total transmit power $p_n = P_n - p_{n,0}$ so the same applies to the user's cost function $\Gamma_n$; furthermore, for $k = 1, \ldots, K$, some easy algebra yields

$$v_{nk} = \frac{1}{p_{nk}} \frac{\mathrm{SINR}_{nk}}{1 + \mathrm{SINR}_{nk}} \tag{3.39}$$

so any learning scheme that relies on these marginal utilities may be implemented in a completely distributed fashion.

**Remark 2.** *A case of particular interest is when the users' cost functions are linear, i.e.,* $\Gamma_n = \lambda_n p_n$ *where* $\lambda_n$ *denotes the cost incurred by the user (monetary or otherwise) per Watt. In this context, the user's marginal cost* $v_{n,0}$ *will be:*

$$v_{n,0} = \lambda_n \tag{3.40}$$

*so a higher price per Watt increases the user's tendency to allocate power to the "unused" channel.*

In view of the above, we will consider the following exponential

learning process in continuous time:

$$\begin{cases} \dot{y}_{n\alpha} = v_{n\alpha} \\ p_{n\alpha} = P_n \dfrac{e^{y_{n\alpha}}}{\sum_{\beta\in\mathcal{K}_0} e^{y_{n\beta}}} \end{cases} \tag{XL$_0$}$$

Of course, in the above formulation, "power" allocated to the virtual, "unused" channel 0 means "power unused due to cost considerations"; accordingly, by exploiting the properties of the exponential map, we may reformulate this learning scheme as:

$$\begin{cases} \dot{z}_{nk} = v_{nk} - v_{n,0} \\ p_{nk} = P_n \dfrac{e^{z_{nk}}}{1 + \sum_{\nu\in\mathcal{K}} e^{z_{n\nu}}} \end{cases} \tag{3.41}$$

This last process admits the following reinforcement interpretation: each (actual) channel $k \in \mathcal{K}$ is scored by aggregating the difference between its marginal utility $v_{nk}$ and the marginal power consumption cost $v_{n,0}$, and power is allocated with exponential sensitivity to these cumulative performance scores.

The first thing that can be verified with respect to the exponential learning scheme (3.41) is that it respects the constraints imposed by the users' power considerations: indeed, $p_{nk} \geq 0$ by definition and $\sum_k p_{nk} = P_n \sum_k \exp(z_{nk}) \big/ \big(1 + \sum_\nu \exp(z_{n\nu})\big) \leq P_n$ for any possible value of the performance scores $z_{nk}$. More importantly, as the next proposition shows, the learning scheme (3.41) guarantees that users converge to a NE of the energy-efficient power allocation game $\mathcal{G}$:

**Proposition 4.** *Let* $\mathbf{p}(t)$ *be the adaptive power allocation policy induced by the continuous-time learning scheme* (3.41) *for some initialization* $z_{nk}(0)$ *of the channels' performance scores. Then, for almost every realization of the system's channel coefficients* $h_{nk}$*, we will have* $\lim_{t\to\infty} \mathbf{p}(t) = \mathbf{p}^*$ *where* $\mathbf{p}^*$ *denotes the game's* (*unique*) *NE.*

*Moreover,* $\mathbf{p}(t)$ *converges to* $\mathbf{p}^*$ *exponentially fast:*

$$D_{\mathrm{KL}}(\mathbf{p}^* \,\|\, \mathbf{p}(t)) = \mathcal{O}(e^{-ct}) \tag{3.42}$$

*where* $c > 0$ *and* $D_{\mathrm{KL}}(\mathbf{p}^*, \mathbf{p}) = \sum_{n,k} p^*_{nk} \log\left(p^*_{nk}/p_{nk}\right)$, *denotes the Kullback-Leibler divergence between* $\mathbf{p}^*$ *and* $\mathbf{p}$.

*Proof.* By decoupling the exponential learning scheme (3.41), it can be shown that its solution trajectories $\mathbf{p}(t)$ satisfy an augmented version of the replicator equation of [48] with an extra strategy to account for the "unused power" channel $0 \in \mathcal{K}_0$. Our claim may then be proved by adapting the proof of Theorem 6 in [48]. □

Despite its appealing convergence properties, (3.41) is a dynamical system that evolves in continuous time, so it is not clear if it can be implemented as a bona fide, discrete-time learning algorithm. In this regard, there are two key challenges to overcome: *a*) to establish a properly discretized version of (3.41) which retains its convergence in discrete time; and *b*) to ensure the algorithm's robustness in the presence of impefect CSI and noisy SINR observations.

To that end, we will work here with the following stochastic diminishing-step discretization of (3.41):

$$\begin{cases} z_{nk}(m+1) = z_{nk}(m) + \gamma_m\big[\hat{v}_{nk}(m) - \hat{v}_{n,0}(m)\big] \\ p_{nk}(m+1) = P_n \dfrac{e^{z_{nk}(m)}}{1 + \sum_\nu e^{z_{n\nu}(m)}} \end{cases} \tag{3.43}$$

where $m = 1, 2, \ldots,$ is the iteration counter of the process, $\gamma_m$ is a variable step size whose role will be discussed below and $\hat{v}_{n\alpha}$ represents a perturbed version of the user's marginal utility at the $n$-th update period. In particular, to account for as wide a range of measurement

errors as possible, we will assume that

$$\hat{v}_{nk}(m) = v_{nk}(m) + \xi_{nk}(m) \tag{3.44}$$

where the observational error $\xi_{nk}$ is a bounded martingale difference (not necessarily independent and identically distributed (i.i.d.)), i.e. $\|\xi_{nk}\| \leq \Sigma$ for some $\Sigma > 0$ and $\mathbb{E}\left[\xi_{nk}(m) \,|\, \xi_{nk}(n-1), \ldots, \xi_{nk}(1)\right] = 0$.

By employing the stochastic optimization techniques of [145], it is then possible to show:

**Proposition 5.** *Let $\gamma_m$ be a variable step size sequence such that $\sum_n \gamma_m \to +\infty$ and $\sum_n \gamma_m^2 < +\infty$. Then, for almost every realization of the system's channel coefficients $h_{nk}$, the iterates of the learning scheme* (3.43) *with imperfect measurements given by* (3.44) *converge to the game's (unique) NE.*

*Proof.* the learning scheme (3.43) can be seen as a greedy mirror descent method [145] for $V$ with respect to the $L^1$ norm on $\mathcal{X}$ and with the Shannon–Gibbs entropy as a "distance-generating function" in the sense of [145]. With this in mind, the analysis of [145] can be used to show that $\mathbb{E}[V_0(\mathbf{p}(m)) - V_0(\mathbf{p}^*)] = \mathcal{O}\left(\sum_{j=1}^m \gamma_j^2 / \sum_{j=1}^m \gamma_j\right)$, where $\mathbf{p}^*$ is the game's unique equilibrium. In turn, this implies that $\mathbf{p}(m) \to \mathbf{p}^*$ and establishes our claim. □

### 3.2.3 Numerical Analysis

To validate the predictions of Section 3.2.2.2 for the performance of the learning scheme (3.43) in power-constrained RANs, we conducted extensive numerical simulations from which we illustrate here a selection of the most representative scenarios.

Throughout this section, and unless explicitly stated otherwise, we will assume a population of $N = 10$ users and $K = 20$ subcarriers, while the channel gain coefficients $h_{nk}$ will be drawn randomly from

$[0, 1]$. For simplicity, we also assume symmetric channels and users, i.e. $\sigma_k = 1$ for all $k \in \mathcal{K}$ and $P_n = P = 1\,\mathrm{W}$ for all $n \in \mathcal{N}$.

With regards to the cost function $c_n$ of the utility model (3.29), we will consider three distinct cases: no pricing (NP), linear pricing (LP), and nonlinear pricing (NLP). Specifically:

1. The NP model is defined trivially as $c_{\mathrm{NP}}(p) = 0$.

2. The LP model is defined as $c_{\mathrm{LP}}(p) = \lambda p$ for some $\lambda > 0$.

3. The NLP model is defined as $c_{\mathrm{NLP}}(p) = \lambda\big(e^{p/P} - 1\big)$.[9].

Fig. 3.6 shows how the users' average transmit power, achieved transmission rate and sum-rate evolve at each iteration of the the learning scheme (3.43). As expected, the end-state of the learning scheme (3.43) depends quite strongly on the number of users and available channels: as could be expected, best performance is achieved in the uncontested regime where the number of available channels is higher than the number of users trying to access them, i.e. $N/K < 1$. Fig. 3.6 also shows how the NLP model affects the power allocation process; in fact, since NLP leads to a sharp increase of the transmission cost for higher powers, it follows that the corresponding loss in transmission rate is not negligible compared to the LP scheme.

In Fig. 3.7 we plot the sum-rate, average allocated power and its respective cost as a function of different pricing models and values of the pricing parameter $\lambda$ for different network configurations. The most interesting result is that three distinct regions can be identified: *a*) For $\lambda$ below a certain threshold $\lambda_l$, the transmission cost in negligible compared to the contribution of the transmission rate in the utility function, so pricing does not impact the system's performance at equilibrium; *b*) in the second region, say $\lambda_l < \lambda < \lambda_u$, the average

[9]Note that the constant term in $c_{NLP}(p)$ does not affect the equilibrium of the game as it represents a flat power rate for each user.

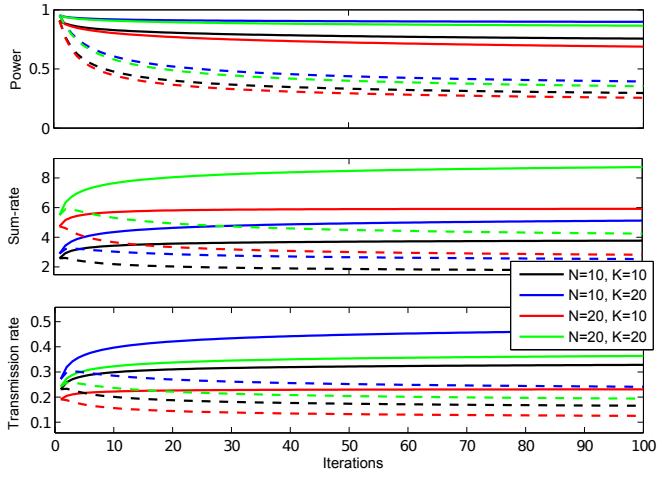



***Figure 3.6:*** *Evolution of the average allocated power, sum-rate and average transmission rate for the LP (solid line) and NLP (dashed line) models for different network configurations.*

allocated power and the sum-rate decrease with $\lambda$, thus leading to a nontrivial trade-off between achievable transmission rate and the cost of power consumption; finally, for large $\lambda > \lambda_u$ the transmission cost is so high that it ends up dominating each user's utility function, so users remain relatively quiet due to the high cost of power consumption. An important result is that the average cost paid by users for each transmission is maximized at $\lambda = \lambda_l$ (recall that for $\lambda \leq \lambda_l$ the exact value of $\lambda$ does not affect network performance and users don't mind paying a small cost in order to maximize their throughput). This result could be interesting and useful in all of those scenarios where the receiver of the uplink channel, e.g., the network operator which sells its channels by applying a pricing model, wants to maximize its revenues while maintaining high network performance.[10] Fig. 3.7 also shows that LP performs better than NLP in terms of the users' transmission rate,

[10]Note that the revenue/profit maximization occurs when $\lambda = \lambda_l$ but, this problem lies beyond the scope of this thesis.

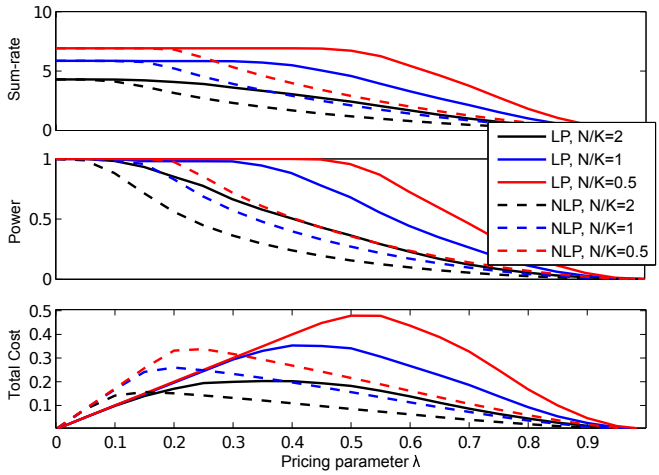



***Figure 3.7:*** *Sum-rate, average allocated power and respective cost as a function of the pricing parameter $\lambda$ for different channel configurations.*

precisely because users incur a higher cost under the NLP model (so users will allocate less power on each available channel thus decreasing their achievable transmission rates).

We also investigated the impact of different values of the ratio between the number of channels $K$ and the number of users $N$ on the system's overall performance. As shown in Fig. 3.7, the congested regime $N > K$ leads to worse aggregate throughput values, as expected; in fact, an increase in the number of users reduces the SINR of each user transmitting on the same channel. On the other hand, if $N \leq K$ there is a reduction in the multiuser interference, so higher transmission rates can be achieved for all users.

Finally, to assess the the learning scheme's convergence speed, we plotted the system's equilibration level (EQL), defined as follows:

$$\mathsf{EQL}(m) = \frac{V_0(\mathbf{p}(m)) - V_{0,\min}}{V_{0,\max} - V_{0,\min}} \tag{3.45}$$

where $\mathbf{p}(m)$ is the users' power profile at the $m$-th iteration of the exponential learning (3.41) and $V_{0,\min}$ (resp. $V_{0,\min}$) denotes the maximum (resp. minimum) value of the game's potential $V_0$ [48]. In Fig. 3.8 we plot the equilibration rate of (3.41) for different pricing models and values of the step-size parameter $\gamma_m$. In the case of a diminishing step size, our exponential learning scheme converges to equilibrium more slowly than if a constant step size is used, and the learning scheme's convergence speed increases with $\gamma_m$. In Fig. 3.8 we further investigate the impact of different pricing strategies in the game. When power consumption carries no cost (the NP regime), the learning scheme converges very fast to an equilibrium point which saturates the users' power constraint; otherwise, the learning scheme's convergence to equilibrium is slower in the LP than in the NLP case, a phenomenon which implies that the optimal choice of step size for the learning scheme (3.43) depends delicately on the users' pricing scheme.[11]

In Fig. 3.9 we plot the game's equilibration level (EQL) under the linear pricing model for $\gamma_m = 1.25, 2.5$ and different values of $K$ and $N$: as expected, the convergence rate with $\gamma_m = 2.5$ is faster compared to the one achieved for $\gamma_m = 1.25$. The most interesting result concerns the scalability of the algorithm: as a matter of fact, the algorithm converges to equilibrium within a number of iterations that is roughly independent of the underlying network configuration.

[11]Importantly, the convergence of the learning scheme (3.43) with a variable step-size $\gamma_m = 1/m$ is guaranteed by Proposition 5; this result does not apply to the constant step-size case, but, nonetheless, the learning scheme converged to the game's (unique) NE in all our simulations.

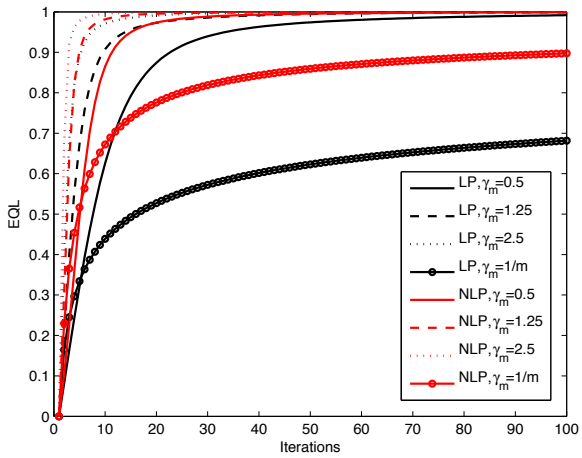



***Figure 3.8:*** *Evolution of the equilibration rate for different pricing models and values of the step-size $\gamma_m$.*

## 3.3 Jamming-proof RAN communications

In this section, we provide a jamming-proof resource allocation scheme that exploits *timing channels* to establish secure communications between network users and the RAN.

A *timing channel* is a communication channel which exploits silence intervals between consecutive events, such as packet transmissions, to encode information [86]. Accordingly, in a timing channel the output alphabet is made up of different time values, and coding consists in defining the inter-arrival time between an event and the following one. Also, timing channels have been shown to be well-suited communication channels to provide jamming-proof communications [35, 71, 90, 91].

In the following we analyze the interactions between a rective jammer and network users whose transmissions are under attack. Specifically, we assume that network users want to maximize the amount of

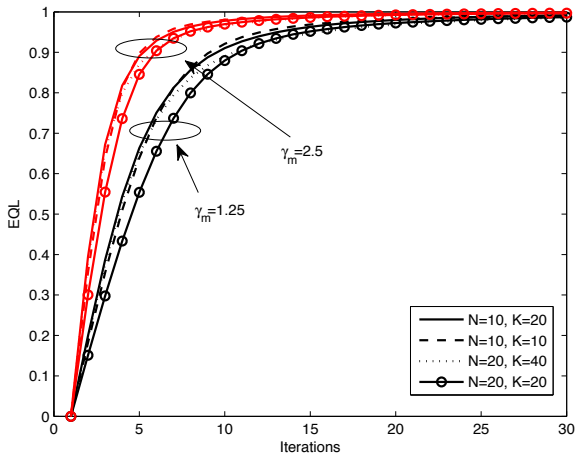



***Figure 3.9:*** *Evolution of the equilibration rate for the LP model for different values of the step-size $\gamma_m$ and network configurations.*

information that can be transmitted per unit of time by means of the timing channel. Instead, the jammer wants to minimize such amount of information while reducing the energy expenditure. As the target node and the jammer have conflicting interests, we develop a game theoretical framework that models their interactions. We investigate both the case in which these two adversaries play their strategies simultaneously, and the situation when the target node (the leader) anticipates the actions of the jammer (the follower). To this purpose, we study both the Nash Equilibria (NEs) and Stackelberg Equilibria (SEs) of our proposed games.

More in detail, we prove the existence, uniqueness and convergence to the NE under best response dynamics; we prove the existence and uniqueness of the equilibrium of the Stackelberg game where the target node plays as a leader and the jammer reacts consequently. Finally, we investigate the impact on the achievable performance of *imperfect knowledge* w.r.t. jammer's system parameters.

### 3.3.1 System Model and Problem Formulation

We consider the challenging case where all frequency channels in $\mathcal{K}$ are attacked by the jammer. Also, to properly exploit timing channels, we assume that only one network user is allowed to access a subcarrier at any given time [12]. However, note that the case where multiple users access the same subcarrier can be tackled by exploiting timing channel-based MAC protocols such as the one proposed in [10].

For the sake of illustration, in this thesis we focus on a single user $n \in \mathcal{N}$ which is attacked by the malicious node. However, the same formulation also holds for any other user in $\mathcal{N}$. Thus, in the following we refer to the malicious node as the *jammer*, $J$, and the user under attack as the *target node*, $T$.

The jammer senses the wireless channel continuously. Upon detecting a possible transmission activity performed by $T$, $J$ starts emitting a jamming signal. As shown in Fig. 3.10, we denote as $T_{AJ}$ the duration of the time interval between the beginning of the packet transmission and the beginning of the jamming signal emission. The duration of the interference signal emission that jams the transmission of the $j$-th packet can be modeled as a continuous random variable, which we call $Y_j$. To maximize the uncertainty on the value of $Y_j$, we assume that it is exponentially distributed with mean value $y$.

We assume that when no attack is performed the target node communicates with the receiver by applying traditional transmissions schemes; on the other hand, when it realizes to be under attack, it exploits the timing channel to transmit part of (or all) the information[13]. The latter is encoded in the duration of the interval between

[12]The approach we propose in this thesis is general and we do not make any assumption of the mechanism that the network operator allocates the considered subcarriers in $\mathcal{K}$.

[13]Attack detection can be achieved by the target node either by means of explicit notification messages sent back to $T$ by the receiver or by inference after missing reception of ACK messages. Details on attack detection operations are however

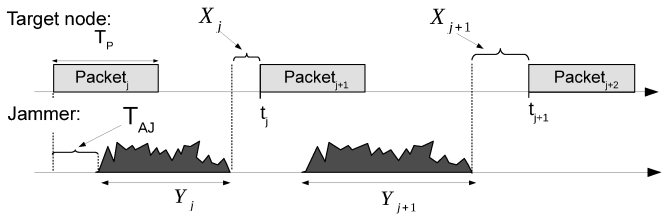



***Figure 3.10:*** *Interactions between the jammer and the target node.*

the instant when the jammer $J$ terminates the emission of the jamming signal and the beginning of the transmission of the next packet. Hence, it is possible to consider a discrete time axis and refer to each timing channel utilization by means of an integer index $j$. The silence period duration scheduled after the transmission of the $j$-th packet and the corresponding jamming signal can be modeled as a continuous random variable, $X_j$, uniformly distributed[14] in the range $[0, x]$. The amount of information transmitted per each use of the timing channel depends on the value of $x$ and the precision $\Delta$ of the clocks of the communicating nodes as shown in [87]. In our model we assume that the parameters $\Delta$ and $T_{AJ}$ which are hardware dependent are known a-priori to both the target node and the jammer, whereas the strategies $x$ and $y$ are estimated by means of a training phase. This is consistent with the complete information assumption which is common in game theoretic frameworks.

To model the interactions between the target node and the jammer we propose a jamming game framework, defined by a 3-tuple $\mathcal{G} = \mathcal{G}(\mathcal{N}, \mathcal{S}, \mathcal{U})$, where $\mathcal{N}$ is the set of players, $\mathcal{S}$ is the strategy set, and $\mathcal{U}$ is the utility set. The set $\mathcal{N}$ is composed by the target node $T$ and the jammer $J$, while the strategy set is $\mathcal{S} = \mathcal{S}_{\mathcal{T}} \times \mathcal{S}_{\mathcal{J}}$, where $\mathcal{S}_{\mathcal{T}}$

out of the scope of this thesis.

[14]The uniform distribution assumption is due to the fact that, as well known, this distribution maximizes the entropy, given the range in which the random variable is defined.

and $\mathcal{S}_{\mathcal{J}}$ are the set of strategies of the target node and the jammer, respectively.

In our model we assume that the jammer is energy-constrained, e.g., it is battery-powered; hence, its choice of $y$ (i.e., the average duration of the signal emission that jams the packet transmission) stems from a trade-off between two requirements, i.e., i) reduce the amount of information that the target node $T$ can transmit to the perspective receiver, and ii) keep the energy consumption as low as possible. Observe that requirement i) would result in the selection of a high value for $y$, whereas requirement ii) would result in a low value for $y$. On the other hand, the target node has to properly choose the value of $x$ (i.e., the maximum silence period duration scheduled following the transmission of the $j-th$ packet and the subsequent jamming signal) in order to maximize the achievable capacity $\mathcal{C}(x, y)$, i.e., the amount of information that can be sent by means of the timing channel, while minimizing its energy consumption. Therefore, it is reasonable to consider that the values of $x$ and $y$ represent the *strategies* for the target node $T$ and the jammer $J$, respectively. Accordingly, the set of strategies for both players, $\mathcal{S}_{\mathcal{T}}$ and $\mathcal{S}_{\mathcal{J}}$, can be defined as the set of all the feasible strategies $x$ and $y$, respectively.

The utility set of the game is defined as $\mathcal{U} = (\mathcal{U}_{\mathcal{T}}, \mathcal{U}_{\mathcal{J}})$, where $\mathcal{U}_{\mathcal{T}}$ and $\mathcal{U}_{\mathcal{J}}$ are the utility functions of the target node and the jammer, respectively. As already said, the target node aims at maximizing its own achievable capacity, $\mathcal{C}(x, y)$ while also minimizing its energy consumption. The jammer, on its side, aims at reducing the capacity achieved by the target node by generating interference signals, whose duration is $y$ (in average), while keeping its own energy consumption low. Accordingly, the utility functions $\mathcal{U}_T(x, y)$ and $\mathcal{U}_J(x, y)$ to be

maximized are defined as follows:

$$\begin{cases} \mathcal{U}_T(x,y) &= +\mathcal{C}(x,y) - c_{T^*} \cdot T_P \cdot P \\ \mathcal{U}_J(x,y) &= -\mathcal{C}(x,y) - c_T \cdot y \cdot P_J \end{cases} \tag{3.46}$$

where $P$ and $P_J$ are the transmission power of the target node and the jammer, respectively, $T_P$ is the duration of a transmitted packet in seconds, $c_{T^*}$ and $c_T$ are positive transmission costs expressed in $[\text{bit}/(\text{s} \cdot \text{J})]$ which weight the two contributions in the utility functions and therefore, in the following will be referred to as *weight parameters*. Note that while the energy consumption of the jammer varies as a function of the strategy $y$ of the jammer itself, on the contrary the energy consumption of the target node during a cycle only depends on the duration $T_P$ of the packet and not on the strategy. Furthermore, a low value of $c_T$ means that the jammer considers its jamming effectiveness more important than its energy consumption, while a high $c_T$ value indicates that the jammer is energy-constrained and, as a consequence, it prefers to save energy rather than reducing the capacity of the target node. We observe that $c_T = 0$ models the case of continuous jamming without any energy constraint, which is of limited interested and out of the scope of this thesis, since we focus on studying the trade-off between the achievable capacity and the consumed energy.

Let us now calculate the capacity $\mathcal{C}(x,y)$ which appears in the utility function (3.46). To this purpose, we denote the interval between two consecutive transmissions executed by $T$ as a *cycle*. The expected duration of a cycle is

$$t_{\text{Cycle}} = T_{AJ} + y + x/2 \tag{3.47}$$

The capacity $\mathcal{C}(x,y)$ can be derived as the expected value of the information transferred during a cycle, $c_{\text{Cycle}}(x,y)$, divided by the expected duration of a cycle, $t_{\text{Cycle}}$. It is easy to show that $c_{\text{Cycle}}(x,y)$ is ap-

proximately

$$c_{\text{Cycle}} = \log_2 (x/\Delta) \tag{3.48}$$

Note that at each timing channel utilization the target node $T$ is expected to transmit at least one bit; then, from (3.48) it follows that $x \geq 2\Delta$.

(3.47) and (3.48) can be exploited to calculate the capacity $\mathcal{C}(x, y)$, i.e.,

$$\mathcal{C}(x, y) = \frac{\log_2 (x/\Delta)}{T_{AJ} + y + x/2} \tag{3.49}$$

### 3.3.2 Nash Equilibrium Analysis

In this Section we solve the game described in Section 3.3.1, and we find the NE, in which both players achieve their highest utility given the strategy profile of the opponent. In the following we also provide proofs of the *existence, uniqueness* and *convergence* to the NE under best response dynamics.

Let us define a NE for game $\mathcal{G}$:

**Definition 1.** *A strategy profile* $(x^*, y^*) \in \mathcal{S}$ *is a NE if* $\forall (x', y') \in \mathcal{S}$

$$\mathcal{U}_T(x^*, y^*) \geqslant \mathcal{U}_T(x', y^*)$$
$$\mathcal{U}_J(x^*, y^*) \geqslant \mathcal{U}_J(x^*, y')$$

*that is,* $(x^*, y^*)$ *is a strategy profile where no player has incentive to deviate unilaterally.*

One possible way to study the NE and its properties is to look at the *best response functions* (BRs). A best response function is a function that maximizes the utility function of a player, given the opponents' strategy profile. Let $b_T(y)$ be the BR of the target node and $b_J(x)$ the BR of the jammer. These functions can be characterized as follows:

$$b_T(y) = \arg\max_{x \in \mathcal{S}_\mathcal{T}} \mathcal{U}_\mathcal{T}(x, y)$$

$$b_J(x) = \arg\max_{y \in \mathcal{S}_\mathcal{J}} \mathcal{U}_\mathcal{J}(x, y)$$

In our model it is possible to analytically derive the closed form of the above BRs by analyzing the first derivatives of $\mathcal{U}_\mathcal{T}(x, y)$ and $\mathcal{U}_\mathcal{J}(x, y)$, and imposing that $\frac{\partial}{\partial x}\mathcal{U}_\mathcal{T}(x, y) = 0$ and $\frac{\partial}{\partial y}\mathcal{U}_\mathcal{J}(x, y) = 0$.

It is easy to see that $\frac{\partial}{\partial x}\mathcal{U}_\mathcal{T}(x, y) = 0$ leads to

$$\frac{1}{x} - \frac{1}{2}\log\left(\frac{x}{\Delta}\right)\frac{1}{T_{AJ} + y + \frac{x}{2}} = 0 \tag{3.50}$$

(3.50) can be rewritten as follows:

$$\frac{2(T_{AJ} + y)}{e\Delta} = \frac{x}{e\Delta} \cdot \log\frac{x}{e\Delta} \tag{3.51}$$

Note that (3.51) is in the form $\beta = \alpha \log \alpha$, and, by exploiting the definition of *Lambert W-function*, say $W(z)$, which, for any complex $z$, satisfies $z = W(z)e^{W(z)}$, it has solution $\alpha = e^{W(\beta)}$.

Therefore, (3.51) can also be rewritten as

$$x = \Delta e^{W\left(\frac{2(T_{AJ}+y)}{e\Delta}\right)+1}$$

which is, by definition, $b_T(y)$.

In order to derive the closed form of $b_J(x)$ we first solve $\frac{\partial}{\partial y}\mathcal{U}_\mathcal{J}(x, y) = 0$. It can be easily proven that $\frac{\partial}{\partial y}\mathcal{U}_\mathcal{J}(x, y) = 0$ leads to

$$\log\left(\frac{x}{\Delta}\right) = \eta\left(T_{AJ} + \frac{x}{2} + y\right)^2$$

which can be rewritten as follows:

$$b_J(x) = \sqrt{\frac{\log(\frac{x}{\Delta})}{\eta}} - T_{AJ} - \frac{x}{2}$$

where $\eta = c_T \cdot P_J \cdot \log 2$.

Therefore, we can write

$$b_T(y) = \Delta e^{\psi(y)+1} \tag{3.52}$$

$$b_J(x) = \begin{cases} \chi(x), & \text{if } \chi(x) \geqslant 0 \\ 0, & \text{if } \chi(x) < 0 \end{cases} \tag{3.53}$$

where

$$\psi(y) = W\left(\frac{2[T_{AJ}+y]}{e\Delta}\right) \quad , \quad \chi(x) = \sqrt{\frac{\log(\frac{x}{\Delta})}{\eta}} - T_{AJ} - \frac{x}{2} \tag{3.54}$$

Note that the best response of the jammer $b_J(x)$ depends on the value of the weight parameter $c_T$. Also, it can be shown that there exists a critical value of the weight parameter, say $c_T^{(\max)}$, such that $b_J(x) < 0$ $\forall x \in \mathcal{S}_{\mathcal{T}}$, $\forall c_T \geq c_T^{(\max)}$. In fact, since the function $\chi(x)$ is strictly decreasing in $c_T$, $\lim_{c_T \to +\infty} \chi(x) < 0$ and $\lim_{c_T \to 0} \chi(x) = +\infty$, the *intermediate value theorem* ensures the existence of $c_T^{(\max)}$. By looking at the first derivative of the $\chi(x)$ function in (3.54), it can be shown that $c_T^{(\max)} = \frac{1}{P_J \log(2)} \frac{1}{2\Delta(\Delta+T)}$. Therefore, if $c_T \geq c_T^{(\max)}$ the only possible strategy of the jammer is $b_J(x) = 0$, and then, as the strategy set of the jammer ($\mathcal{S}_{\mathcal{J}}$) is a singleton, the game has a trivial outcome.

#### 3.3.2.1 Existence and Uniqueness of the Nash Equilibrium

It is well known that the intersection points between $b_T(y)$ and $b_J(x)$ are the NEs of the game. Therefore, to demonstrate the existence of at least one NE, it suffices to prove that $b_T(y)$ and $b_J(x)$ have one or more intersection points. In other words, it is sufficient to find one or more pairs $(x^*, y^*) \in \mathcal{S}$ such that

$$(b_T(y^*), b_J(x^*)) = (x^*, y^*) \tag{3.55}$$

To this aim, in the following we provide some structural properties of the utility functions, $\mathcal{U}_{\mathcal{T}}(x,y)$ and $\mathcal{U}_{\mathcal{J}}(x,y)$, that will be useful in solving (3.55).

**Lemma 2.** *For the utility functions $\mathcal{U}_{\mathcal{T}}(x,y)$ and $\mathcal{U}_{\mathcal{J}}(x,y)$, the following properties hold* [15]*:*

- *$\mathcal{U}_{\mathcal{T}}(x,y)$ is strictly concave for $x \in [2\Delta, x']$ and is monotonically decreasing for $x > x'$ where $x' = b_T(y)$*
- *$\mathcal{U}_{\mathcal{J}}(x,y)$ is strictly concave $\forall y \in \mathcal{S}_{\mathcal{J}}$.*

**Theorem 2** (NE existence)**.** *The game $\mathcal{G}$ admits at least an NE.*

*Proof.* If we limit the strategy of the target node to $[2\Delta, x']$, it follows from Lemma 2 that there exists at least an NE since both the utility functions are concave in the restraint strategy set [146]. However, this does not still prove the existence of the NE in the non-restraint strategy set $\mathcal{S}_{\mathcal{T}}$. Let $(x^*, y^*)$ denote the NE with a restraint strategy set $[2\Delta, x']$; we can easily observe that $(x^*, y^*)$ is also the NE of the jamming game with non-restraint strategy set. To show this, recall Lemma 2 that states that $\mathcal{U}_{\mathcal{T}}(x,y)$ is monotonically decreasing for $x > x'$. The transmitter has thus no incentive to deviate from $(x^*, y^*)$ and the jammer has no incentive to deviate from it either. Therefore, $(x^*, y^*)$ is the NE of the jamming game. □

After proving the NE existence in Theorem 2, let us prove the uniqueness of the NE, that is, there is only one strategy profile such that no player has incentive to deviate unilaterally.

[15] The proof of Lemma 2 which is straightforward (although quite long), consists in calculating the first and second derivatives of the utility functions and studying them.

**Theorem 3** (NE uniqueness)**.** *The game $\mathcal{G}$ admits a unique NE that can be expressed as*

$$(x_{\text{NE}}, y_{\text{NE}}) = \\ = \begin{cases} \left( \Delta e^{\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right)}, \frac{\Delta}{2}\left[\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right) - 1\right] e^{\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right)} - T_{AJ} \right) & \text{if } c_T < \tilde{c_T} \\ \left( \Delta e^{W(\frac{2T}{e\Delta})+1}, 0 \right) & \text{otherwise} \end{cases} \tag{3.56}$$

*where $\eta = c_T \cdot P_J \cdot \log 2$ and*

$$\tilde{c_T} = \frac{4}{\Delta^2 P_J \log 2} e^{-2\left[W\left(\frac{2T}{e\Delta}\right)+1\right]} / \left( W\left(\frac{2T}{e\Delta}\right) + 1 \right) \tag{3.57}$$

The proof consists in exploiting formal and structural properties of the best response functions to show that their intersection is unique, that is, (3.55) admits a unique solution. For a detailed proof see Appendix A

#### 3.3.2.2 Convergence to the Nash Equilibrium

We now analyze the convergence of the game to the NE when players follow *Best Response Dynamics* (BRD). In BRD the game starts from any initial point $(x^{(0)}, y^{(0)}) \in \mathcal{S}$ and, at each successive step, each player plays its strategy by following its best response function. Thereby, at the $i$-th iteration the strategy profile $(x^{(i)}, y^{(i)})$ can be formally expressed by the following BRD iterative algorithm:

$$\begin{cases} x^{(i)} = b_T(y^{(i-1)}) \\ y^{(i)} = b_J(x^{(i-1)}) \end{cases}$$

Let $\mathbf{b}(x, y) = (b_T(y), b_J(x))^T$ be the best response vector and $J_{\mathbf{b}}$

be the Jacobian of $\mathbf{b}(x, y)$ defined as follows

$$J_{\mathbf{b}} = \begin{bmatrix} \frac{\partial}{\partial x} b_T(y) & \frac{\partial}{\partial y} b_T(y) \\ \frac{\partial}{\partial x} b_J(x) & \frac{\partial}{\partial y} b_J(x) \end{bmatrix} = \begin{bmatrix} 0 & \frac{\partial}{\partial y} b_T(y) \\ \frac{\partial}{\partial x} b_J(x) & 0 \end{bmatrix} \tag{3.58}$$

It has been demonstrated [147] that, if the Jacobian infinity matrix norm $||J_{\mathbf{b}}||_\infty < 1$, the BRD always converges to the unique NE. In the following we prove the following theorem:

**Theorem 4** (NE convergence - sufficient condition)**.** *The relationship*

$$c_T > \frac{1}{9\Delta^2 \log 2P_J} \frac{1}{\left(W\left(\frac{2T_{AJ}}{e\Delta}\right)+1\right) e^{2\left(W\left(\frac{2T_{AJ}}{e\Delta}\right)+1\right)}} \tag{3.59}$$

*is a sufficient condition for the game $\mathcal{G}$ to converge to the NE. Furthermore, it converges to the NE in at most $\log_{J_b^{max}} \frac{\epsilon}{||s^1 - s^0||}$ iterations for any $\epsilon$, where $J_b^{max} = \max J_{\boldsymbol{b}}$ and $s^i = (x^i, y^i)$.*

To demonstrate the theorem,

1. we prove that the relationship

$$\max_{x \in \mathcal{S}_{\mathcal{T}}} \left( \frac{1}{\eta x^2 \log(\frac{x}{\Delta})} \right) < 9 \tag{3.60}$$

   is a sufficient condition for the BRD to converge to the NE in at most $\log_{J_b^{max}} \frac{\epsilon}{||s^1 - s^0||}$ iterations. This is the focus of Lemma 3;

2. we define a game $\mathcal{G}'$ and demonstrate that $\mathcal{G}$ converges to $\mathcal{G}'$ in two iterations at most. This is the focus of Lemma 4;

3. we demonstrate that the condition in (3.59) is a sufficient condition for $\mathcal{G}'$ to satisfy (3.60) and converge to the same NE of $\mathcal{G}$. This is the focus of Lemma 5.

**Lemma 3.** *The BRD converges to the unique NE from any* $(x^{(0)}, y^{(0)}) \in \mathcal{S}$ *if* $\max_{x \in \mathcal{S}_\mathcal{T}} \left( \frac{1}{\eta x^2 \log(\frac{x}{\Delta})} \right) < 9$ *in at most* $\log_{J_b^{max}} \frac{\epsilon}{||s^1 - s^0||}$ *iterations.*

The proof is based on showing that the above relationship is a sufficient condition for the Jacobian infinity matrix norm $||J_\mathbf{b}||_\infty$ to be always lower than 1, and thus, according to [147], convergence of the BRD follows. We refer the reader to Appendix B for a detailed proof of Lemma 2.

Let us now observe that $b_J(x)$ is lower-bounded as it is non-negative ($b_J(x) \geqslant 0$) and, since it is concave, it has a maximum, say $y_M$, for $\hat{x} = \Delta e^{\frac{1}{2} W(\frac{2}{\eta \Delta^2})}$, and thus it is upper-bounded ($b_J(x) \leqslant y_M = b_J(\hat{x})$). Also, it is easy to prove that $b_T(y)$ is a non-negative strictly increasing function, hence, it is lower-bounded by $x_m = b_T(0)$. We can thus define a new strategy set $\mathcal{S}' = \mathcal{S}_\mathcal{T}{}' \times \mathcal{S}_\mathcal{J}{}' = [x_m, x_M] \times [0, y_M]$, where $\mathcal{S}' \subset \mathcal{S}$ and $x_M = b_T(y_M)$, which is relevant in the following lemma:

**Lemma 4.** *Given any starting point* $(x^{(0)}, y^{(0)}) \in \mathcal{S}$*, the BRD is bounded in* $\mathcal{S}'$ *in at most two iterations. That is,* $(x^{(i)}, y^{(i)}) \in \mathcal{S}'$ *for* $i = 2, 3, ..., +\infty$*.*

*Proof.* Let $\mathcal{S}^{(1)}$ be the strategy set at the first iteration. From (3.52) and (3.53) we have that $b_J(x)$ is lower and upper-bounded by $y = 0$ and $y = y_M$, respectively, thus $y^{(1)} \in [0, y_M]$. Furthermore, as $b_T(x)$ is lower-bounded by $x = x_m$ and $y^{(0)} \in \mathcal{S}_\mathcal{J} = [0, +\infty[$, it follows that $x^{(1)} \in [x_m, +\infty)$. Hence, we have that $\mathcal{S}^{(1)} = \mathcal{S}_\mathcal{T}{}^{(1)} \times \mathcal{S}_\mathcal{J}{}^{(1)} = [x_m, +\infty) \times [0, y_M]$, $\mathcal{S}^{(1)} \subset \mathcal{S}$. Due to the boundedness of $y^{(1)}$ which assumes values in $\mathcal{S}_\mathcal{J}{}^{(1)}$, it can be shown that at the second iteration $x^{(2)} \in [x_m, x_M]$ while $y^{(2)} \in [0, y_M]$, thus, we have that $(x^{(2)}, y^{(2)}) \in \mathcal{S}'$. We can extend the same reasoning to the $j$-th iteration ($\forall j = 3, 4, ..., \infty$) to obtain that $(x^{(j-1)}, y^{(j-1)}) \in \mathcal{S}'$. Therefore, it follows that $(x^{(j)}, y^{(j)})$ is still in $\mathcal{S}'$, which concludes the proof. □

**Lemma 5** (NE convergence)**.** *If the parameter $c_T$ satisfies the condition:*

$$c_T > c'_T = \frac{1}{9\Delta^2 \log 2P_J} \frac{1}{\left(W\left(\frac{2T_{AJ}}{e\Delta}\right)+1\right) e^{2\left(W\left(\frac{2T_{AJ}}{e\Delta}\right)+1\right)}} \tag{3.61}$$

*then $\mathcal{G}'$ converges to the NE of $\mathcal{G}$.*

*Proof.* Since the function on the left-hand side of (3.60) is non-negative and strictly decreasing, and the minimum value of $\mathcal{S}_\mathcal{T}$ is $x_m = \Delta e^{W\left(\frac{2T_{AJ}}{e\Delta}\right)+1}$, then

$$\max_{x \in \mathcal{S}_\mathcal{T}} \left( \frac{1}{\eta x^2 \log(\frac{x}{\Delta})} \right) = \frac{1}{\eta x_m^2 \log(\frac{x_m}{\Delta})} \tag{3.62}$$

It is easy to show that if (3.61) holds, then

$$\frac{1}{\eta x_m^2 \log(\frac{x_m}{\Delta})} < 9$$

and therefore, recalling (3.62), (3.60) holds. From Lemma 3 we thus obtain that $\mathcal{G}'$ converges to its NE.

We still need to demonstrate that $\mathcal{G}$ and $\mathcal{G}'$ converge to the same equilibrium point. To this purpose it is sufficient to prove that the equilibrium point of $\mathcal{G}$ is in $\mathcal{S}'$. Theorem 3 guarantees that the game $\mathcal{G}$ admits a unique equilibrium, which has to be in $\mathcal{S}$. Let $(x_{\text{NE}}, y_{\text{NE}})$ be the NE, i.e., the unique intersection point between $b_T(y)$ and $b_J(x)$. As $b_J(x)$ takes values in $[0, y_M]$ it follows that $y_{\text{NE}} \in [0, y_M]$; therefore, $x_{\text{NE}} = b_T(y_{\text{NE}}) \in [x_m, x_M]$. It follows that $(x_{\text{NE}}, y_{\text{NE}}) \in \mathcal{S}'$, which concludes the proof. □

### 3.3.3 Stackelberg Game

In a Stackelberg game one of the players acts as the leader by anticipating the best response of the follower. In our scenario, the jammer plays its strategy when a communication from the target node is detected on the monitored channel; thus, it is natural to assume that the target node acts as the leader followed by the jammer. Obviously, given the strategy of the target node $x$, the jammer will play the strategy that maximizes its utility, that is, its best response $b_J(x)$[16]. This hierarchical structure of the game allows the leader to achieve a utility which is at least equal to the utility achieved in the ordinary game $\mathcal{G}$ at the NE, if we assume *perfect knowledge*, that is, the target node is completely aware of the utility function of the jammer and its parameters, and thus it is able to evaluate $b_J(x)$. Whereas, if some parameters in the utility function of the jammer are unknown at the target node, i.e., the *imperfect knowledge* case, the above result is no more guaranteed as it is impossible to evaluate the exact form of $b_J(x)$. In this section we analyze the Stackelberg game and provide useful results about its equilibrium points, referred to as Stackelberg Equilibria (SEs).

**Definition 2.** *A strategy profile $(x^*, y^*) \in \mathcal{S}$ is a Stackelberg Equilibrium (SE) if $y^* \in {\mathcal{S}_{\mathcal{J}}}^{\mathrm{NE}}(x)$ and*

$$x^* = \arg\max_{x'} \ \mathcal{U}_{\mathcal{T}}(x', y^*)$$

*where ${\mathcal{S}_{\mathcal{J}}}^{\mathrm{NE}}(x)$ is the set of NE for the follower when the leader plays its strategy $x$.*

In the following we will prove that, in the case of perfect knowledge, there is a unique SE for any value of the weight parameter $c_T$,

[16]In the following, given that the value of $c_{T^*}$ does not impact on the game, for worth of simplicity we assume that $c_{T^*} = 0$.

and we demonstrate that the target node can inhibit the jammer under the perfect knowledge assumption. Next, we will investigate the implications of imperfect knowledge on the game outcome.

#### 3.3.3.1 Perfect knowledge

Under the perfect knowledge assumption, the target node selects $x$ in such a way that $\mathcal{U}_T(x, b_J(x))$ is maximized, where $\mathcal{U}_T(x, b_J(x))$ is calculated in (3.63a) and (3.63b) by replacing expression (3.53) in (3.49) and (3.46).

$$\mathcal{U}_T(x, b_J(x)) = \\ = \begin{cases} \sqrt{c_T P_J \log_2\left(\frac{x}{\Delta}\right)} - c_{T^*} \cdot T_P \cdot P & \text{if } \chi(x) > 0 \text{ (3.63a)} \\ \log_2\left(\frac{x}{\Delta}\right) / \left(T_{AJ} + \frac{x}{2}\right) - c_{T^*} \cdot T_P \cdot P & \text{otherwise (3.63b)} \end{cases}$$

By analyzing the first derivative of $\chi(x)$, it can be shown that $\chi(x)$ has a maximum in $\hat{x} = \Delta e^{\frac{1}{2}W(\frac{2}{\eta\Delta^2})}$ and, consequently, $\chi(x)$ is strictly decreasing for $x > \hat{x}$ and strictly increasing for $x < \hat{x}$.

In the following we show that for any value of $c_T$ there exists a unique Stackelberg Equilibrium, and this is when the jammer does not jam the timing channel[17]. Furthermore, we show that the leader can improve its utility at the Stackelberg equilibrium if and only if $c_T < \tilde{c_T}$.

**Theorem 5.** *For any value of the parameter $c_T$, the Stackelberg game $\mathcal{G}_\mathcal{T}$ has a unique equilibrium.*

*Proof.* First, we prove that the game admits a unique equilibrium for $c_T \geq c_T^{(\max)}$. Recall that $c_T \geq c_T^{(\max)}$ implies $b_J(x) = 0$; therefore,

[17]In this case the jammer is expected to transmit the interference signal for a short time interval only because this suffices to disrupt communications, as occurs in traditional communication channels.

$\mathcal{S}_\mathcal{J}$ is singleton and the unique feasible strategy for the jammer at the SE is $y_{\text{SE}} = 0$. In fact, due to the high cost associated to the emission of the jamming signal, the jammer is inhibited $\forall x \in \mathcal{S}_\mathcal{T}$. Hence, it can be easily proved that the strategy profile at the SE is $(x_{\text{SE}}, y_{\text{SE}}) = (\Delta e^{W(\frac{2T_{AJ}}{e\Delta})+1}, 0)$, that is, at the SE the target node selects the strategy that maximizes the capacity of the non-jammed timing channel (where indeed $y_{\text{SE}} = 0$).

Instead, if $c_T < c_T^{(\max)}$, from (3.54) we have that $\chi(\hat{x}) > 0$. Thus, for the intermediate value theorem there exist $x_1 < \hat{x}$ and $x_2 > \hat{x}$ such that $\chi(x_1) = \chi(x_2) = 0$, as shown in Fig. 3.11.

Let us denote $\mathcal{S}_{\mathcal{T}1} = \{x \in [2\Delta, x_1]\}$, $\mathcal{S}_{\mathcal{T}2} = \{x \in [x_1, x_2]\}$, $\mathcal{S}_{\mathcal{T}3} = \{\mathcal{S}_\mathcal{T} \setminus (\mathcal{S}_{\mathcal{T}1} \cup \mathcal{S}_{\mathcal{T}2})\}$, and $x' = \Delta e^{W(\frac{2T_{AJ}}{e\Delta})+1}$. It can be easily proved that $x'$ maximizes (3.63b) and, since $\chi(x') > 0$, it follows that $x' \in \mathcal{S}_{\mathcal{T}2}$. Therefore, the utility function of the target node as defined in (3.63b) increases for $x < x'$ and decreases for $x > x'$. The latter is fundamental to prove the theorem; in fact, as shown in Fig. 3.11, for $x \in \mathcal{S}_{\mathcal{T}1}$ the utility of the target node is defined by (3.63b) and strictly increases as $x$ increases; therefore, we have that in $\mathcal{S}_{\mathcal{T}1}$ the maximum utility is achieved in $x_1$. On the contrary, in $\mathcal{S}_{\mathcal{T}2}$ the utility is defined by (3.63a), which is a strictly increasing function that achieves its maximum value for $x = x_2$. Finally, for $x \in \mathcal{S}_{\mathcal{T}3}$ we have that the utility of the transmitter defined by (3.63b) strictly decreases as $x > x'$; hence, the maximum value is achieved for $x = x_2$.

Since $\mathcal{U}_T(x, b_J(x)) < \mathcal{U}_T(x_2, b_J(x_2))$ with $x \neq x_2$, it follows that, to maximize its own utility, the target node must play the unique strategy $x = x_2$. Note that $\chi(x_2) = 0$ by definition, thus from (3.53) we have that the strategy of the jammer at the equilibrium is $y_{\text{SE}} = 0$. Therefore, $x_{\text{SE}} = x_2$ is the strategy of the target node at the SE, and we can identify the unique SE as $(x_{\text{SE}}, y_{\text{SE}}) = (x_2, 0)$, which concludes the proof. □

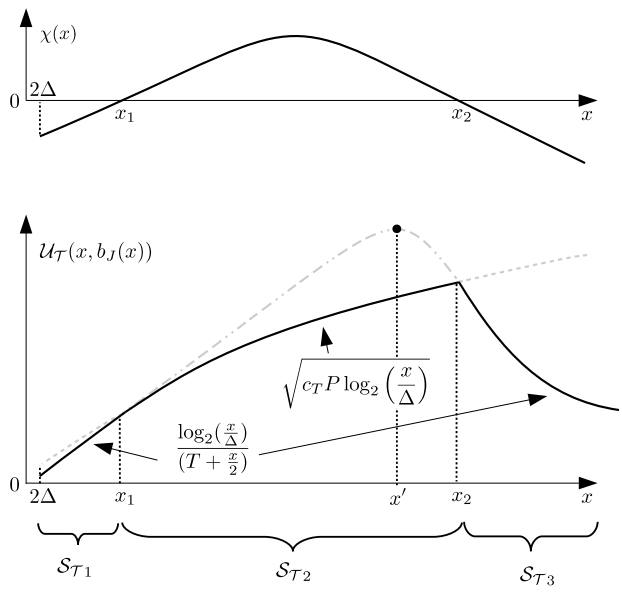



***Figure 3.11:*** *Graphical representation of $\chi(x)$ and $\mathcal{U}_{\mathcal{T}}(x, b_J(x))$ in the Stackelberg game. The solid line is the actual utility of the target node in each strategy subset.*

Let us remark that the above Theorem also highlights an insightful side-effect: at the Stackelberg equilibrium, pursuing the goal of inhibiting the jammer makes the target node prefer to increase transmission delay rather than reduce its achievable capacity.

Let us also note that, although an analytical closed form for $x_{\mathrm{SE}}$ cannot be easily derived, its value can be determined by means of numerical search algorithms such as the bisection search algorithm. Obviously, such algorithms will not give the exact value of $x_{\mathrm{SE}}$; in fact, they will return an interval $[x_m, x_M]$ small as desired, containing the solution, i.e., $x_{\mathrm{SE}} \in [x_m, x_M]$, and eventually the target node will select the minimum or the maximum value of the interval which gives the highest utility function. Let $\epsilon(x_m, x_M)$ denote the loss in the utility of the target node due to the fact that it cannot determine the exact value of $x_{\mathrm{SE}}$. Given that the utility function is continuous and that its

derivative is upperbounded by $u_{\max} = \sqrt{c_T \cdot P}/(4\Delta \log 2)$ in $[x_1, x_{\text{SE}}]$, it is possible to show that selecting the interval size in such a way that

$$x_M - x_m \leq \epsilon^* / u_{\max} \tag{3.64}$$

the loss in the utility of the target node, $\epsilon(x_m, x_M)$, is lower than $\epsilon^*$. In other terms, by using numerical search algorithms such as the bisection search algorithm, the target node can make the loss in its utility as small as desired.

In the following we provide an approximation $x'_{\text{SE}}$ that can be helpful from a practical point of view. Let us assume that $\left(T_{AJ} + \frac{x}{2}\right) \approx \frac{x}{2}$, therefore, (3.54) can be rewritten as follows

$$\frac{\log(\frac{x}{\Delta})}{\log(2) c_T P} = \left(\frac{x}{2}\right)^2 \tag{3.65}$$

By means of simple manipulations it can be easily shown that (3.65) admits the following solution:

$$x'_{\text{SE}} = \Delta e^{-\frac{1}{2} W\left(-\frac{\log(2) c_T P \Delta^2}{2}\right)} \tag{3.66}$$

In Section 3.3.4 we will provide numerical results that show how much the approximation in (3.66) affects the outcome of the Stackelberg game.

**Theorem 6.** *In the Stackelberg game the target node improves its utility as compared to the NE if and only if* $0 < c_T < \tilde{c_T}$.

*Proof.* Let us start with the proof of the sufficient condition implied by the Theorem 6. According to (3.56) and (3.63a), proving that $\mathcal{U}_T(x_{\text{SE}}, b_J(x_{\text{SE}})) > \mathcal{U}_T(x_{\text{NE}}, y_{\text{NE}})$ is equivalent to showing that

$$\sqrt{c_T P \log_2(\frac{x_{\text{SE}}}{\Delta})} > \frac{1}{\log 2} \frac{2}{\Delta} e^{-\frac{1}{2} W(\frac{8}{\eta \Delta^2})}$$

that is

$$\frac{1}{2}W(\frac{8}{\eta\Delta^2}) < \log(\frac{x_{\mathrm{SE}}}{\Delta})$$

This only holds if $x_{\mathrm{SE}} > \Delta e^{\frac{1}{2}W(\frac{8}{\eta\Delta^2})} = x_{\mathrm{NE}}$. Recall that if $0 < c_T < \tilde{c}_T$, the NE is an interior NE, that is, $\chi(x_{\mathrm{NE}}) > 0$. Therefore, as $\chi(x_{\mathrm{SE}}) = 0$, it must hold that $x_{\mathrm{NE}} < x_{\mathrm{SE}}$, which proves the sufficiency condition. As for the necessary condition, we have to show that, if $c_T \geqslant \tilde{c_T}$, no improvement can be achieved by the target node. In fact, if $c_T \geqslant \tilde{c_T}$ it is straightforward to prove that the NE and the SE coincide, and thus, the utilities of the target node at the SE and NE are equal. □

#### 3.3.3.2 Imperfect knowledge

We now investigate the implications of *imperfect knowledge* on the weight parameter $c_T$ in (3.46). In Theorem 5 we proved that the optimal strategy in the Stackelberg game is $x_{\mathrm{SE}}$ such that $\chi(x_{\mathrm{SE}}) = 0$. According to (3.54) the value of $c_T$ is needed to evaluate $x_{\mathrm{SE}}$. However, it is reasonable to assume that in realistic scenarios the value of $c_T$ is not available at the target node, while instead, only statistical information on the distribution of $c_T$ is likely known. Let us denote as $f_{c_T}(\xi)$ the probability density function (pdf) of the random variable representing the weight parameter $c_T$. We also denote as $g(\xi)$ the function returning the strategy of the target node at the SE, $x_{\mathrm{SE}}$, when the weight parameter for the jammer is $c_T = \xi$.

The resulting utility function of the target node $\mathbf{U}_T^{\xi} = \mathcal{U}_T(g(\xi), b_J(g(\xi))$ can be calculated as

$$\mathbf{U}_T^{\xi} = \begin{cases} \sqrt{c_T P \log_2\left(\dfrac{g(\xi)}{\Delta}\right)} & \text{if } \xi > c_T \qquad (3.67a) \\ \log_2\left(\dfrac{g(\xi)}{\Delta}\right) / \left(T_{AJ} + \dfrac{g(\xi)}{2}\right) & \text{if } \xi \leq c_T \qquad (3.67b) \end{cases}$$

Let us refer to $E\{\mathbf{U}_T^{\xi}\}$ as the expected value of the utility function of the target node. Assuming that $f_{c_T}(\xi)$ is a continuous function, it follows that

$$E\{\mathbf{U}_T^{\xi}\} = \int_{-\infty}^{+\infty} \mathcal{U}_T(\xi|c_T=\alpha) f_{c_T}(\alpha) d\alpha = \\ = \int_{-\infty}^{\xi} \mathcal{U}_T(\xi|c_T=\alpha) f_{c_T}(\alpha) d\alpha + \int_{\xi}^{+\infty} \mathcal{U}_T(\xi|c_T=\alpha) f_{c_T}(\alpha) d\alpha$$

From (3.67a) and (3.67b) we have

$$E\{\mathbf{U}_T^{\xi}\} = \int_{-\infty}^{\xi} \sqrt{\alpha P \log_2(\frac{g(\xi)}{\Delta})} f_{c_T}(\alpha) d\alpha + \int_{\xi}^{+\infty} \frac{\log_2(\frac{g(\xi)}{\Delta})}{(T_{AJ} + \frac{g(\xi)}{2})} f_{c_T}(\alpha) d\alpha = \\ = \sqrt{P \log_2(\frac{g(\xi)}{\Delta})} \int_{-\infty}^{\xi} \sqrt{\alpha} f_{c_T}(\alpha) d\alpha + \frac{\log_2(\frac{g(\xi)}{\Delta})}{(T_{AJ} + \frac{g(\xi)}{2})} \int_{\xi}^{+\infty} f_{c_T}(\alpha) d\alpha \tag{3.68}$$

By exploiting the relationship in (3.54), (3.68) can be rewritten as

$$E\{\mathbf{U}_T^{\xi}\} = P\left(T_{AJ} + \frac{g(\xi)}{2}\right) \sqrt{\xi} \left[ \int_{-\infty}^{\xi} \sqrt{\alpha} f_{c_T}(\alpha) d\alpha + \sqrt{\xi} \int_{\xi}^{+\infty} f_{c_T}(\alpha) d\alpha \right] \tag{3.69}$$

Note that the target node has first to find $\xi^* = \arg\max_{\xi} E\{\mathbf{U}_T^{\xi}\}$, and then, the optimal strategy is evaluated as $x_{\mathrm{SE}}(\xi^*)$ such that $\chi(x_{\mathrm{SE}}(\xi^*)) = 0$.

In the following we analyze the especially relevant case when the random variable $\xi$ is uniformly distributed in a closed interval[18], that

[18]Note that the uniform distribution represents the worst case, as it is the distribution that maximizes the uncertainty on the actual value of $c_T$, given that a minimum and a maximum values are given.

is, the pdf of $\xi$ is defined as

$$f_{c_T}(\xi) = \begin{cases} \frac{1}{\xi_{max}-\xi_{min}} & \text{if } \xi \in [\xi_{min}, \xi_{max}] \\ 0 \quad \text{otherwise} \end{cases} \tag{3.70}$$

By substituting (3.70) in (3.69), we obtain the following expression

$$E\{\mathbf{U}_T^{\xi}\} = P\frac{\left(T_{AJ} + \frac{g(\xi)}{2}\right)}{\xi_{max} - \xi_{min}} \left[\xi\xi_{max} - \frac{1}{3}\xi^2 - \frac{2}{3}\xi^{\frac{1}{2}}\xi_{min}^{\frac{3}{2}}\right] \tag{3.71}$$

In order to maximize the expected utility we study the first derivative of (3.71), which leads to:

$$\frac{W\left(-\frac{P\log(2)\Delta^2}{2}\xi\right)}{1+W\left(-\frac{P\log(2)\Delta^2}{2}\xi\right)}\left(\xi_{max} - \frac{1}{3}\xi - \frac{2}{3}\frac{\xi_{min}^{\frac{3}{2}}}{\sqrt{\xi}}\right) = 2\xi_{max} - \frac{4}{3}\xi - \frac{2}{3}\frac{\xi_{min}^{\frac{3}{2}}}{\sqrt{\xi}} \tag{3.72}$$

The solution of (3.72), say $\xi_{opt}$, is the value of $\xi$ that maximizes the expected utility of the target node. Regrettably, $\xi_{opt}$ can be evaluated only numerically. Thus, in the aim of providing practical methods to choose $\xi$, in the next section we will discuss some analytical results that show how $\xi = \xi_{max}$ well approximates $\xi_{opt}$. In fact, if we assume $W\left(-\frac{P\log(2)\Delta^2}{2}\xi\right) / \left[1+W\left(-\frac{P\log(2)\Delta^2}{2}\xi\right)\right] \approx 1$, then, (3.72) can be reformulated as

$$\xi_{max} - \frac{1}{3}\xi = 2\xi_{max} - \frac{4}{3}\xi$$

whose solution is $\xi = \xi_{max}$. Furthermore, we will show that the above approximation guarantees high efficiency at the SE even if the uncertainty on the actual value of $c_T$ is high, as in the case of a uniform distribution.

| Name | Value | Unit |
|---|---|---|
| $T_{AJ}$ | 15 | $\mu$s |
| $\Delta$ | 1 | $\mu$s |
| $P = P_J$ | 2 | W |
| $T_P$ | 50 | $\mu$s |

***Table 3.2:*** *Parameter settings used in our simulations.*

### 3.3.4 Numerical Analysis

In this section we apply the theoretical framework developed in the previous sections to numerically analyze the equilibrium properties for both the ordinary and Stackelberg games. As introduced in Section 3.3.1, the settings of the relevant parameters are those in Table 3.2. It is also assumed that both the target node and the jammer transmit their respective signals by using the same transmitting power, i.e., $P = P_J$.

#### 3.3.4.1 Nash Game

In Fig. 3.12 we show the best response functions of both the target node and the jammer for different values of the weight parameter $c_T$. As already said, the NE is the intersection point between the best response functions. As expected, the best response of the target node does not depend on the value of $c_T$, while this is not true for the best response of the jammer. Note that for high $c_T$ values the jammer reduces its jamming signal duration $y$, and the strategy of the target node consists in reducing the maximum silence duration $x$.

Figs. 3.13 and 3.14 illustrate the strategy of the players at the NE as a function of $c_T$ for different values of the transmitting power $P$. Note that, as $c_T$ increases, the target node decreases the maximum silence duration and the jammer reduces the jamming signal duration

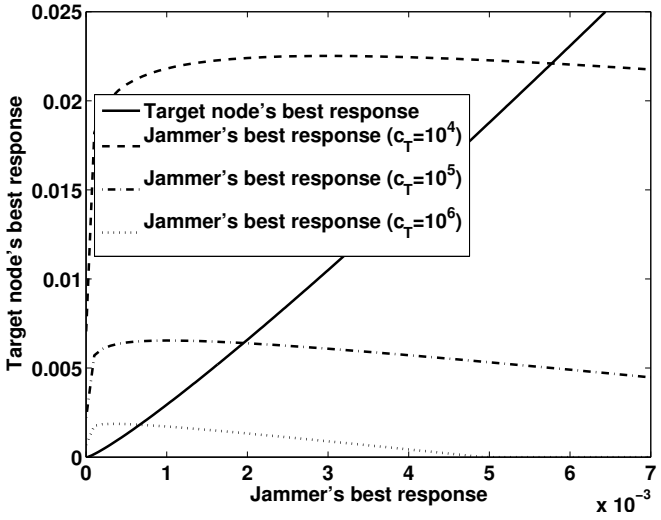



***Figure 3.12:*** *Best response functions for both the target node and the jammer.*

as well. In fact, upon increasing $c_T$ the jammer acts in an energy preserving fashion and this causes a decrease in $y$. Such a behavior allows the target node to behave more aggressively by reducing the maximum silence duration $x$. Furthermore, upon increasing $P$, the strategies $x$ and $y$ decrease as higher $P$ values force the jammer to reduce the jamming signal duration and, thus, the energy consumption. Also, the target node can reduce $x$, thus increasing its achieved capacity.

Figs. 3.15 and 3.16 illustrate how the BRD evolves at each iteration for different values of the weight $c_T$. Since we proved that the game converges to the NE, Figs. 3.15 and 3.16 show how, as expected, the players' strategies converge to the strategy set $\mathcal{S}'$ in 2 iterations (as discussed in Lemma 4) and to the NE in at most 7 iterations[19]. It is also shown that an increase in the value of $c_T$ causes a decrease in

[19]Note that, although we proved that the convergence to the NE is guaranteed only if $c_T < \tilde{c_T}$, in our simulations the game always converges to the NE in a few iterations, independently of the value of $c_T$.

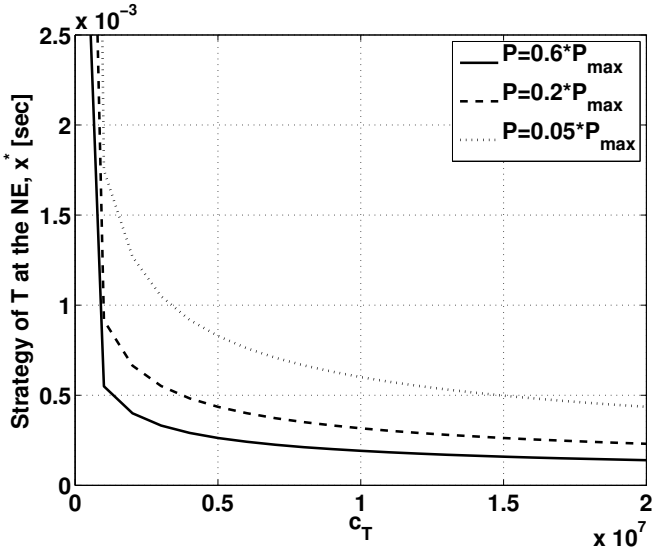



***Figure 3.13:*** *Strategy of the target node at the NE as a function of the weight parameter $c_T$ for different values of the transmitting power $P$, ($P_{max} = 2$).*

the strategies of both players due to the aggressive behavior of the jammer.

#### 3.3.4.2 Stackelberg Game

We now turn to the analysis of the Stackelberg game, where the target node anticipates the jammer's reaction. In this regard, Fig. 3.17 compares the utilities achieved by each player at the NE and SE. Note that, as proven in Theorem 6, the utility achieved by the target node at the SE is higher than, or at least equal to, the utility achieved at the NE. Moreover, at the SE the utility is higher than at the NE for the jammer as well. In fact, the target node increases the maximum silence duration $x$, that is, it increases transmission delay, and inhibits the jammer. Accordingly, the jammer stops its disrupting attack, and thus, it saves energy; as a result, its utility increases when compared to

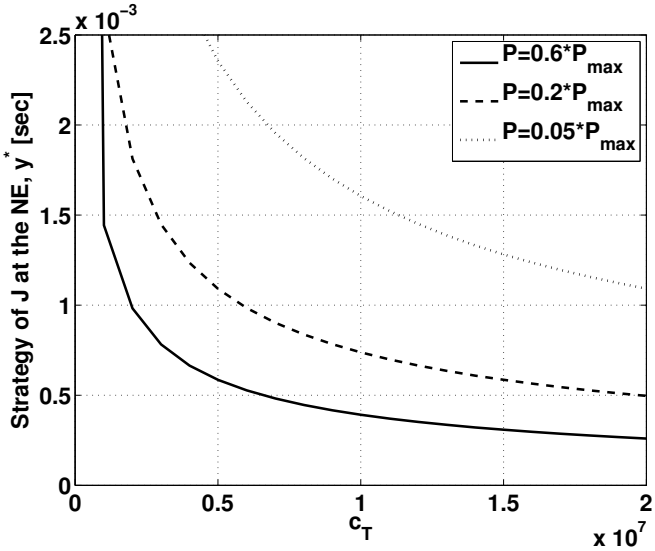



***Figure 3.14:*** *Strategy of the jammer at the NE as a function of the weight parameter $c_T$ for different values of the transmitting power $P$, ($P_{max} = 2$).*

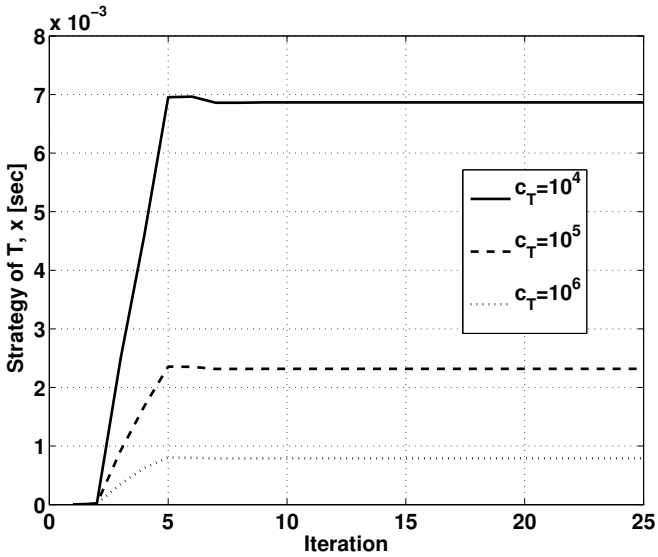



***Figure 3.15:*** *Strategy of the target node at each iteration.*

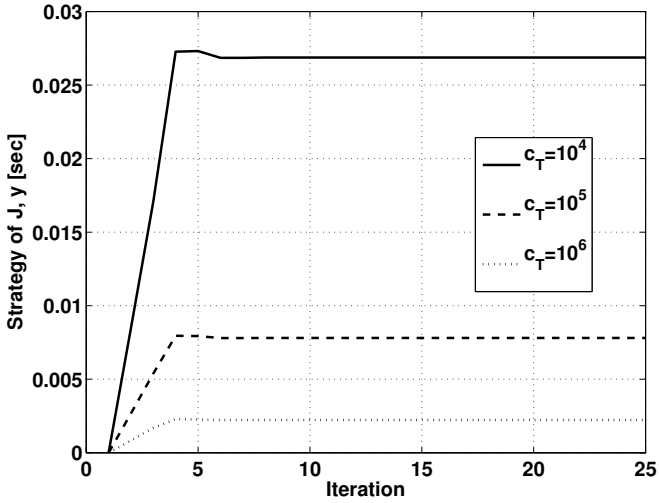



***Figure 3.16:*** *Strategy of the jammer at each iteration.*

that at the NE. We further observe that, as expected, for high values of $c_T$, the improvement in the achieved utility becomes negligible, as already proven in Theorem 6.

Under the perfect knowledge assumption, at the SE the strategy of the target node, $x_{\text{SE}}$, coincides with the solution of $\chi(x) = 0$, which can also be approximated to $x'_{\text{SE}}$ as given in (3.66). Accordingly, in Fig. 3.18(a) we compare the utilities of the target node at the SE, in its exact and approximated strategies $x_{\text{SE}}$ and $x'_{\text{SE}}$, respectively. Fig. 3.18(b) shows that the approximation accuracy of $x'_{\text{SE}}$, defined as the ratio between $\mathcal{U}_T(x'_{\text{SE}}, b_J(x'_{\text{SE}}))$ and $\mathcal{U}_T(x_{\text{SE}}, b_J(x_{\text{SE}}))$, strongly depends on the value of $c_T$. As shown in Fig. 3.18(c), the error introduced by the approximation $\left(T_{AJ} + \frac{x_{\text{SE}}}{2}\right) \approx \frac{x'_{\text{SE}}}{2}$ is low when low values of $c_T$ are considered, because, in this case, the strategy of T at the SE, $x_{\text{SE}}$, consists in choosing larger silence durations, and thus $\frac{x_{\text{SE}}}{2} \gg T_{AJ}$. On the contrary, when $c_T$ is high, there is no need for the target node to choose high $x_{\text{SE}}$ values, thus the above approximation introduces a non-negligible error on the estimate of $x'_{\text{SE}}$. Note that, although

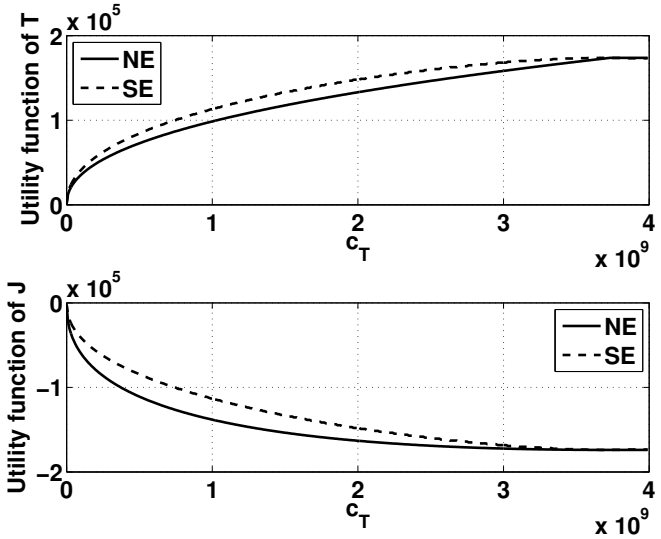


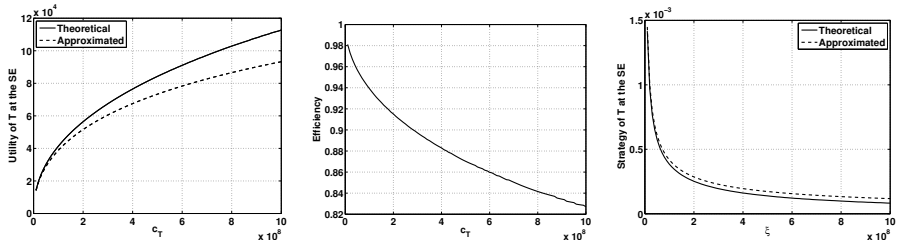

***Figure 3.17:*** *Comparison between the utilities achieved by each player at the NE and SE as a function of the weight parameter $c_T$ ($c_{T^*} \cdot P = 2 \cdot 10^6$).*

the approximation is affected by errors, Fig. 3.18(b) shows that the approximation accuracy is still high (i.e. larger than 82%).

To evaluate the impact of imperfect knowledge on the utility of the target node, let us now define the *equilibrium efficiency* $e(\xi)$ as follows:

$$e(\xi) = \frac{\mathbf{U}_T^{\xi}}{\mathbf{U}_T^{c_T}} \tag{3.73}$$

Fig. 3.19 illustrates the equilibrium efficiency of the target node as a function of $c_T$ for different choices of $\xi$. More in detail, we considered $\xi \in \{\xi_{opt}, \xi_{mean}, \xi_{max}, \xi_{min}\}$, where $\xi_{mean} = (\xi_{max} + \xi_{min})/2$, $\xi_{min} = 10^5$ and $\xi_{max} = 10^9$. Note that in our simulations $\xi_{min} = 10^5$ and $\xi_{max} = 10^9$ are realistic setting assumptions. In fact, lower values of $\xi_{min}$ or higher values of $\xi_{max}$ lead to unbalanced settings as one of the terms in (3.46) will always dominate the other. The most important result is that the equilibrium efficiency when $\xi \in \{\xi_{opt}, \xi_{mean}, \xi_{max}\}$




(a) (b) (c)

***Figure 3.18:*** *Impact of the approximation $x'_{\mathrm{SE}}$ in (3.66) on the Stackelberg game outcome as a function of the weight parameter $c_T$ ($c_T \cdot P = 2 \cdot 10^6$).*

is always higher than 75%, while the case $\xi = \xi_{min}$ achieves a very low equilibrium efficiency (and thus, it is not reported in Fig. 3.19). As demonstrated in Section 3.3.3.2, Fig. 3.19 shows that $\xi_{max}$ well approximates $\xi_{opt}$, i.e., $e(\xi_{opt}) \simeq e(\xi_{max})$. Therefore, from a practical point of view, if the computation of $\xi_{opt}$ is not feasible (e.g., high computational cost and low hardware capabilities) it is still possible to achieve a high equilibrium efficiency by choosing $\xi = \xi_{max}$.

Finally, in Fig. 3.20 we compare the utility functions of the target node and the jammer obtained at the NE and SE with what is obtained in the cases the two players select their strategies without considering the strategies of each other. More specifically we will consider the two following cases:

- *Case A*: The target node selects its strategy $x$ in such a way that its capacity is maximized without considering that the jammer will try to disrupt the communication in the timing channel as well. In other terms, the target node will assume that $y \approx 0$.

- *Case B*: The jammer selects its strategy $y$ assuming that the target node is not aware that it (the jammer itself) is trying to disrupt the communication in the timing channel. In other terms, the jammer will assume that $x \approx b_T(0)$.

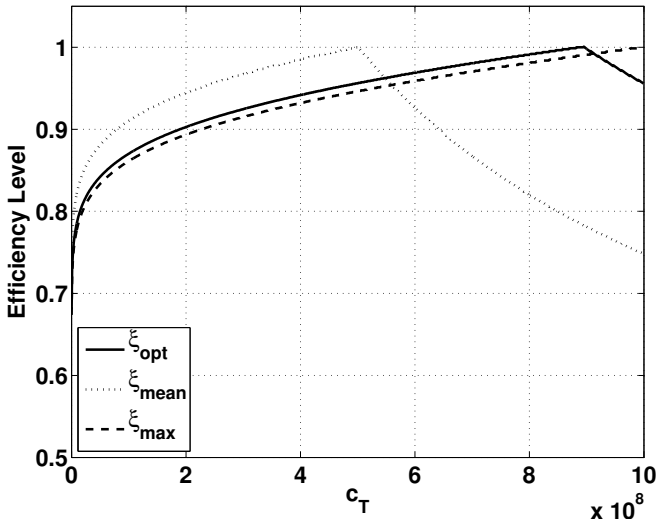



***Figure 3.19:*** *Equilibrium efficiency $e(\xi)$ as a function of the weight parameter $c_T$ ($c_T \cdot P = 2 \cdot 10^6$).*

When compared to the NE and SE cases the utility function of the target node will decrease in Case A and increase in Case B. The viceversa holds for the utility function of the jammer. We observe that the gap between the utility functions obtained in Cases A and B compared to the NE and SE decrease when the cost $c_T$ increases. This is because when the cost $c_T$ increases the jammer becomes more concerned about the energy consumption and therefore the value $y_{\text{NE}}$ becomes smaller. Accordingly, the assumptions considered in Cases A and B become accurate and consequently the behavior approaches what is obtained when each player takes the strategy of the opponent into account.

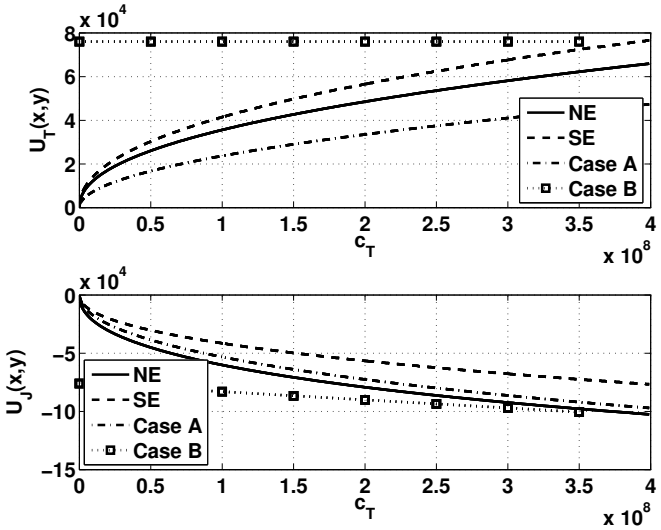



***Figure 3.20:*** *Comparison between the utility of the target node and the jammer when they work at the NE, at the SE and what is obtained in Case A and B.*

CHAPTER

# FOUR

# NETWORK AND SERVICE MANAGEMENT FOR THE MULTI-TENANT BACKHAUL

To provide efficient network and service management in the backhaul, opportunistic and dynamic resource allocation is desirable. Recently, software-defined networking and virtualization of network functions have attracted interests from both the academia and the industry as they provide flexible and dynamic resource management. Also, by exploiting SDNs and NFV it is possible to support the multitenancy concept.

Multitenancy allows several tenants to share the backhaul. On the one hand, it makes possible to share network resources and, if proper network management and control is performed, to avoid resources wastage. On the other hand, it also leads to competition among tenants. In fact, tenants are often competitors and aim at maximizing their revenues. Therefore, the competitive and non-cooperative nature of multi-tenant networks calls for privacy-preserving resource allocation schemes. To this scope, allocation of resources has to be performed by avoiding (or limiting) interactions and exchange of messages among tenants.

Accordingly, to address the above issues we exploit analytical tools from game theory which naturally provides mechanisms to model interactions such those we consider in this section. Specifically, in Section 4.1 we propose an auction-based game-theoretic resource allocation scheme that relies on a SDN approach for dynamic and flexible management of network resources in a backhaul shared among multiple tenants. Instead, in Section 4.2 we exploit tools from hierarchical and evolutionary game theories to provide service and network function management in the CN through NFV.

## 4.1 Network Management in the Multi-tenant Backhaul

In SDNs, control and data planes are decoupled. Network control and management are centralized and implemented in software, while the data/forwarding plane consists of an underlying physical network composed by several SDN-compliant switches and links. Although there are several ways to implement SDNs, in this thesis we consider OpenFlow [98] as it is the most popular implementation of SDNs. As we show later, OpenFlow specifications already provide procedures to support dynamic resource allocation in multi-tenant backhauls.

In a SDN, multiple tenants can coexist; thus, to properly manage their interactions, OpenFlow provides a FlowVisor [111], which is a high-level controller that is designed to act as a proxy between the physical network and multiple tenants. By exploiting FlowVisor protocols, OpenFlow fully supports the multitenancy principle. In fact, FlowVisor and OpenFlow together allow the backhaul owner to divide the backhaul resources into *slices* and give full control of each slice to one tenant that, to this purpose, runs a software program referred

to as *Controller*[1]. OpenFlow and FlowVisor ensure isolation between slices and therefore, each Controller can use its share of the network resources as if it was the sole controller doing it.

In this thesis, we consider OpenFlow and we address the case where the FlowVisor reserves a portion of the network resources and divides it among the Controllers that compete with each other to obtain such resources. Our problem formulation is general and can be applied to several resource allocation problems in SDN scenarios. However, for illustration purposes we focus on two relevant resource allocation problems where each Controller competes to obtain either additional space in OpenFlow routing tables, i.e., *Flow Tables*, to store its routing policies, or bandwidth on a certain backhaul link to improve its achievable throughput.

### 4.1.1 System Model and Problem formulation

The backhaul consists of a physical network of switches operating under an OpenFlow framework that enables software-defined networking. The behavior of OpenFlow switches is completely determined by the content of a Flow Table (or several) [98]. Each entry of such tables specifies a flow and how packets of such flow must be treated. More specifically, each entry contains three sections [98]:

- *Rules:* it is used to identify packets belonging to the flow. In fact, it specifies the values in the header of the packets which determine the belonging to the considered flow.

- *Actions:* it specifies how packets belonging to the flow should be treated, e.g., where a packet should be forwarded, whether it

[1]In the following of this thesis,the terms tenant and Controller are used interchangeably. Also, we will refer to the backhaul operator that possesses the backhaul as the FlowVisor.

should be dropped, whether and how they should be modified, etc.

- *Stats:* it contains statistical information which can be used by the elements in the control plane to tune their policies.

The FlowVisor manages a set of OpenFlow-compliant network *slices* (logical sub-networks composed of switches and links that are isolated from one another even if they share the same underlying physical resources of the backhaul), each of which is assigned to a single Controller who is able to fully control decisions based on its own requirements and forwarding policies. The Controllers enforce their policies by inserting appropriate entries in the Flow Table. Instead, the FlowVisor is able to control both the backhaul topology and the network status by monitoring parameters such as delay, network load and link utilization.

In this context, we consider an auction-based resource management scheme for SDNs where a portion of the available resources in the backhaul is leased to interested tenants $\mathcal{N}_{\mathcal{C}} = \{1, \ldots, N_C\}$. Let $\hat{R}$ denote the total amount of the shared resource. Without losing in generality, we assume that $\hat{R}$ is finite. i.e, $\hat{R} < +\infty$. However, note that we do not make any assumption on the type of the shared resource. In fact, in our context backhaul resources could be identified with bandwidth (like in traditional networks) as well as Flow Table entries. It follows that our model is able to capture a wide variety of scenarios. Furthermore, let $R \equiv \xi \hat{R}$ denote the fraction of the total resource that the FlowVisor decides to lease via the auction mechanism to the Controllers that want to improve the capability of their slices.

In the auction, the FlowVisor acts as the auctioneer and the participating Controllers act as independent bidders that do not com-

municate with each other[2]. Each bidder $i \in \mathcal{N}_\mathcal{C}$ submits a bid $b_i \in \mathcal{B}_i \equiv [0, B_i]$ where $B_i$ denotes the maximum admissible bid of the $i$-th Controller. Note that since $B_i$ is a function of $i$, our model can also capture the case in which Controllers are heterogeneous w.r.t. their economic resources. When the auction is over, the FlowVisor collects all bids and assigns to each bidder $i \in \mathcal{N}_\mathcal{C}$ a fraction $\varrho_i$ of the available resource proportional to the corresponding bid, i.e.

$$\varrho_i(b) = R\frac{b_i}{\sum_{j\in\mathcal{N}_\mathcal{C}} b_j} \tag{4.1}$$

In (4.1), $\varrho_i$ is the portion of the available resource that is assigned to Controller $i$ as a function of the bid $b_i$. The allocation of said portion of available resource is temporary and resource leasing expires as soon as the auction is repeated [117] – typically in time intervals of fixed or variable width according to traffic dynamics of the considered application scenario. To the best of our knowledge, even though this proportional allocation auction scheme has been exploited in other relevant application scenarios [148], its use in this context is novel. For continuous good allocation problems of this type, we could alternatively consider an optimal auction in the sense of Myerson [149] or a continuous second-price sealed-bid auction (Vickrey auction) for multiple continuous goods [150, 151]. The advantage of using (4.1) is that it is much simpler to implement, so it can be deployed in the context of SDNs with minimal computational overhead.

It is evident that each Controller may have a different interest on a particular resource depending on several factors, such as the amount of traffic that flows through a link or a switch, the resource already assigned to the Controller, the number of network users, etc. To account for this, let $\theta_i \in [0, 1]$ denote the *interest factor* of bidder $i$

[2]For simplicity, in what follows, we will use the terms "bidder", "Controller" and "player" interchangeably.

representing the benefit that it expects to receive for a unit of resource (specifically, $\theta_i = 0$ implies that the $i$-th bidder has no interest in the auction)[3]. On the other hand, from a benefit perspective, we assume that the user's benefit from a resource can be modeled as an increasing concave function $\omega_i(b)\colon \mathcal{B}_i \to \mathbb{R}$ in the submitted bid, i.e., Controllers experience diminishing returns when increasing the amount of the submitted bid.

Furthermore, when submitting a bid, each Controller faces an incurred cost (e.g., a monetary cost) which can vary from one Controller to another. The effective cost to each Controller might be different than the actual monetary value of the Controller's bid. The reasons for this are diverse: for instance, each bid could be subject to a value added tax (VAT) or other form of taxation which would increase the actual cost to the user. When the auction ends, additional resources are allocated to participating Controllers. Such resources need to be properly managed and, in general, lead to an additional cost in the network management process. Therefore, Controllers can be charged for this additional management cost. Furthermore, the same monetary amount may have a vastly different impact to Controllers with different budgets, and this impact does not have to be linear in the monetary value of each Controller's bid. For example, when Controllers have a finite budget, doubling a large bid is much costlier (in relative terms) than doubling a small bid [152]. Thus, in tune with standard economic assumptions [152], we posit in what follows that the *effective cost* to the $i$-th Controller is given by an increasing convex cost function $\Gamma_i\colon \mathcal{B}_i \to \mathbb{R}$, called the *cost function* of Controller $i$.

In view of this cost-benefit analysis, the *utility* of bidder $i$ can be

[3]In principle, $\theta_i$ changes over time, however, we assume that auctions take place often enough so that the evolution of $\theta_i$ can be neglected. This assumption is justified by the rapid convergence to the equilibrium as discussed in the following Section 4.1.3.

expressed as the difference between the bidder's expected benefit and the cost to achieve it. More specifically:

$$\mathcal{U}_i(b) = \omega_i(b) - \Gamma_i(b_i), \tag{4.2}$$

where $\omega_i(b)$ is the benefit function for Controller $i$ (which also incorporates the dependency on $\varrho_i$), and $b = (b_1, \ldots, b_{N_C}) \in \mathcal{B} \equiv \prod_i \mathcal{B}_i$ denotes the *bid profile* of the Controllers participating in the auction.

In reality, different scenarios have different features and issues. Therefore, no unique model to represent users' benefits and costs in SDNs exists. On the contrary, the definitions of $\omega_i(b)$ and $\Gamma_i(b_i)$ in (4.2) depend on the considered application scenario. For this reason, in Section 4.1.1.1 we consider two relevant SDNs scenarios and we propose different benefit function models that are representative of a wide variety of SDNs scenarios. Finally, in Section 4.1.1.2 we propose several cost function models that reflect realistic costs experienced in SDN scenarios.

#### 4.1.1.1 Benefit Function Models

**Flow Table Auction** This is the case where the auctioneer, i.e., the FlowVisor, decides to sell a fraction (or all) the available space of a given flow table stored inside a specific switch. We assume that each switch stores a single flow table, and each flow table is stored inside a cache memory. Accordingly, the same flow table is shared among all Controllers that have access to the same switch. To process a flow, the corresponding flow entry has to be defined in the flow table. We assume that each flow entry requires the same amount of space in the flow table. Let $f$ denote this amount of space. Upon receiving each packet, the receiving switch first tries to find a matching rule, or matching entry, for the flow to which the packet belongs to. Then, if a matching rule exists, the corresponding action is performed. Instead,

if no matching entries are found, the switch asks the corresponding Controller for a rule for that flow through a `PacketIn` message. Then, a `FlowMod` message which contains the rule and the corresponding action to be executed for all packets belonging to the same flow is sent by the Controller to the requesting switch. Finally, if the flow table has some available storage space, the requesting switch stores the new flow entry in its flow table.

It is important to focus on the impact of the above operations on backhaul performance. Clearly, if a matching rule is found, the search operation introduces a small delay, say $T_H$. On the contrary, when no matches exist, the transmission of both `PacketIn` and `FlowMod` messages causes a requesting delay $T_R$ that depends on the distance between the requesting switch and the corresponding Controller. For example, the requesting delay $T_R$ can grow up to approximately 100 ms in many cases [153].

In realistic scenarios, we have that $T_R >> T_H$. Also, note that cache memories, i.e., the storage space available for each flow table, are finite. Accordingly, the number of flow entries that can be stored in flow tables is limited. Thus, all Controllers compete with each other to get as much space as possible to store their own entries and reduce the expected forwarding delay:

$$T_i = T_H P_i^H + (1 - P_i^H) T_R \tag{4.3}$$

where $P_i^H$ is the *hit probability* of the $i$-th Controller, i.e., the probability that a rule for the considered flow already exists in the table. It can be easily shown that minimizing the expected delay amounts to maximizing the hit probability $P_i^H$. Also, one can show that the hit probability depends on the amount $\varrho_i$ of available resources to store the additional rules in the flow table that the Controller obtains at the end of the auction.

Accordingly, in the flow table auction scenario the benefit function $\omega_i(b)$ of the $i$-th Controller is given by:

$$\omega_i(b) = \theta_i P_i^H(b) \tag{4.4}$$

However, the hit probability for a given flow depends on the probability distribution of that flow. Therefore, we consider two relevant scenarios depending on the flows' distribution:

- *Uniform distribution:* in this case, Controller's flows are uniformly distributed. Accordingly, the benefit function can be written as
$$\omega_i(b) = \theta_i \frac{L_i + \varrho_i/f}{N_i} \tag{4.5}$$
where $L_i$ is the number of rules already stored by Controller $i$ in the flow table, $N_i$ is the number of flows of Controller $i$ and $\varrho_i$ is defined in (4.1) and represents the space in the flow table, i.e., the cache size, obtained at the end of the auction by Controller $i$. Accordingly, $\varrho_i/f$ represents the number of additional flow entries (and thus, rules) that can be stored in the flow table by the $i$-th Controller normalized by the cache size. Clearly, this is the worst case scenario where all flows are expected to arrive with the same probability and the uncertainty in predicting flow arrivals is maximized.

- *Zipf distribution:* in this case, we assume that flows are distributed according to a Zipf's law [154]. Consequently, there are some popular flows that are expected to arrive with higher probability than unpopular flows. Under the finite cache and infinite request stream assumptions, it has been shown in [155] that the hit probability is asymptotically $\approx \mathcal{O}(R_{\text{cache}}^{1-\alpha})$, where $\alpha \in [0, 1]$ is the Zipf distribution parameter and $R_{\text{cache}}$ is the available storage space in the cache. Accordingly, we have

$P_i^H(b) \approx \mathcal{O}(L_i + \varrho_i/f)^{1-\alpha}$. Finally, by exploiting the Bernoulli's inequality for binomial series, the benefit function $\omega_i(b)$ can be approximated and linearized as follows:

$$\omega_i(b) = \theta_i L_i^{-\alpha}(L_i + (1-\alpha)\varrho_i/f) \tag{4.6}$$

**Bandwidth Auction** The FlowVisor is able to continuously monitor bandwidth and link utilization on each physical link in the backhaul. Therefore, to avoid inefficient resource allocation and to improve link utilization, it is possible to sell the available unused bandwidth on the backhaul link in question.

In this scenario, each Controller seeks to maximize the amount of bandwidth achieved on the considered link at the end of the auction. Accordingly, and in line with traditional auction-based bandwidth allocation schemes, we define the following benefit function:

$$\omega_i(b) = \theta_i \varrho_i \tag{4.7}$$

In view of the above examples, we will mostly focus on the case where the Controllers' benefit functions are linear in the fraction of the obtained resource, i.e., $\omega_i = a_i + c_i\varrho_i$ for some suitable constants $a_i$ and $c_i$.

#### 4.1.1.2 Cost Function Models

When the auction mechanism is over, additional backhaul resources are allocated to SDN Controllers. To manage such resources, additional network management procedures have to be performed by the backhaul operator, leading to extra costs in the network management process. For example, at the end of the flow table auction new flow entries are created and inserted in the flow table. For each incoming flow the switch accesses the flow table and searches for a matching

entry. Such a search procedure introduces a cost which is expected to increase as the number of stored entries increases as well [156]. Intuitively, highly populated flow tables lead to larger search times. On the contrary, when the number of stored flow entries is small, the time needed to access them is small. For example, the time complexity of most search algorithms is well known to be either $\mathcal{O}(F)$ or $\mathcal{O}(\log F)$, where $F$ is the number of entries in the flow table. Thus, since the auction increases the number of entries stored in the flow table, it also increases the cost to search and access such entries.

Similarly, the bandwidth auction also causes an increase in the amount of available bandwidth on each considered link. Thus, Controllers are likely to increase the transmission rate and the amount of data flowing through the backhaul. As a consequence, the utilization factor of the network is expected to increase as well. Thus, to guarantee the correct operation of the network and manage this additional data traffic, additional resource management and control is required (e.g., congestion control, traffic balancing and other similar procedures).

Accordingly, we consider two different models for the cost function $\Gamma_i(b_i)$ in (4.2): linear cost (LC), and non-linear cost (NLC). More in detail:

- in the LC scheme we define $\Gamma_i(b_i) = \lambda b_i / B_i$;
- in the NLC scheme we consider an exponential-based cost scheme [13], i.e., $\Gamma_i(b_i) = \lambda(e^{b_i/B_i} - 1)$.

where $\lambda \geq 0$ is a non-negative cost imposed by the auctioneer to any bidder that buys some resources.

With all this in mind, we can define the following non-cooperative auction game $\mathcal{G} = \mathcal{G}(\mathcal{N}_\mathcal{C}, \mathcal{B}, \mathcal{U})$:

- The set of *players* (or *bidders*) is $\mathcal{N}_\mathcal{C} = \{1, \ldots, N_C\}$.

- The *action space* of each bidder $i \in \mathcal{N}_C$ is $\mathcal{B}_i$.

- The players' *utility functions* $\mathcal{U}_i \colon \mathcal{B} \equiv \prod_i \mathcal{B}_i \to \mathbb{R}$, given as in (4.2).

Accordingly, we will say that a bid profile $b \in \mathcal{B}$ is a NE of $\mathcal{G}$ when no bidder has an incentive to change its bid unilaterally. In a more formal way, this can be stated as:

**Definition 3.** *A bid profile $b^* = (b_1^*, \ldots, b_{N_C}^*) \in \mathcal{B}$ is called a Nash equilibrium (NE) of the auction game $\mathcal{G}$ if*

$$\mathcal{U}_i(b_i^*, b_{-i}^*) \geq \mathcal{U}_i(b_i, b_{-i}^*) \tag{4.8}$$

*for every bid $b_i \in \mathcal{B}_i$ of player $i$ and for all $i \in \mathcal{N}_C$.*

The Nash equilibria of $\mathcal{G}$ represent stable resource management policies where each Controller is satisfied with respect to its individual cost-benefit characteristics and with the existing resource allocation scheme.

As such, the analysis in the following sections will focus on the NE of $\mathcal{G}$ and we will show that the system admits a unique stable state which can be reached by the players in a distributed fashion, i.e., without requiring any coordination among Controllers or the transmission of private information (for example related to each Controller's bids).

### 4.1.2 Nash Equilibrium Analysis

In this section, we examine the existence and uniqueness of the Nash equilibrium and propose an exponential learning mechanism through which the bidders can converge to the equilibrium.

#### 4.1.2.1 Existence and Uniqueness of the Nash Equilibrium

**Theorem 7** (NE existence and uniqueness)**.** *The game $\mathcal{G}$ always admits a NE. In addition, if $\omega_i(b)$ is linear in the amount of the obtained resource $\varrho_i(b)$, the NE of $\mathcal{G}$ is unique.*

*Proof.* Note first that each player's payoff function $\mathcal{U}_i$ is concave in $b_i$ being it defined as the difference between a concave and a convex function; thus, existence of NE follows from the general theory of [146]. To prove the uniqueness of the NE, we define the linear benefit function $\omega_i(b) = a_i + c_i\varrho_i(b)$, where $a_i$ and $c_i$ are two non-negative real numbers, and $\varrho_i(b)$ is defined in (4.1) and represents the resource obtained by Controller $i$ at the end of the auction. Likewise, we prove uniqueness by establishing the Diagonal Strict Concavity (DSC) condition [146], i.e.

$$\sum\nolimits_{i\in\mathcal{N}_C} r_i v_i(b)\cdot[b_i - b_i^*] < 0 \quad \text{for all } b\in\mathcal{B}, \tag{4.9}$$

for some $r_1,\ldots,r_{N_C} > 0$ and for all Nash equilibria $b^*$ of $\mathcal{G}$, where $v_i(b)$ is the so called players' *marginal utility* that we introduce in (4.10). For that, by using the analysis in [157], it suffices to show that *a*) each utility function $\mathcal{U}_i$ is *strictly* concave in $b_i$ and convex in $b_{-i}$; and *b*) the function $\sigma(b,r) = \sum_{i\in\mathcal{N}_C} r_i\mathcal{U}_i(b)$ is concave in $b$ for some $r = (r_1,\ldots,r_{N_C}) \in \mathbb{R}^{N_C}$ with $r_i > 0$.

For the first condition, strict concavity is ensured by the fact that $\mathcal{U}_i(b_i, b_{-i})$ is a sum of a strictly concave and a concave function (in $b_i$); convexity in $b_{-i}$ is also straightforward. For the second condition, if $r_i = c_i^{-1}$, then we can rewrite $\sigma(b,r) = 1 + \sum_{i\in\mathcal{N}_C} r_i a_i - \sum_{i\in\mathcal{N}_C} r_i\Gamma_i(b_i)$; given that the cost functions $\Gamma_i$ are convex, $\sigma(b,r)$ is concave in $b$. Therefore, the claim of the theorem follows from [157]. □

**Remark 3.** *From Theorem 7, we have that the game $\mathcal{G}$ with benefit and cost functions being defined as in Sections 4.1.1.1 and 4.1.1.2 admits a unique NE.*

#### 4.1.2.2 Convergence to the Nash Equilibrium

Theorem 7 shows that the auction initiated by the FlowVisor admits a single stable state where each Controller is unilaterally satisfied with respect to its individual assigned resource and cost-benefit characteristics. However, we still have to prove whether this stable policy can be reached in a distributed fashion. To that end, we propose below a simple learning scheme that allows the bidders to converge to game's unique NE.

The key quantity in what follows will be the players' *marginal utility functions*

$$v_i(b) = \frac{\partial \mathcal{U}_i}{\partial b_i}, \tag{4.10}$$

i.e., the marginal increase (or decrease) in the utility of player $i$ with respect to its own bid $b_i \in \mathcal{B}_i$. In particular, an easy differentiation yields:

$$v_i(b) = \theta_i R \frac{\sum_j b_j - b_i}{\left[\sum_j b_j\right]^2} - \Gamma_i'(b_i) = \theta_i \frac{\varrho_i(R - \varrho_i)}{b_i R} - \Gamma_i'(b_i), \tag{4.11}$$

where $\varrho_i$ is the amount of resource obtained by player $i$ in the auction as defined in (4.1). Accordingly, the marginal utility of player $i$ can be calculated with local information only (knowledge of the amount of resource obtained and the player's own cost function $\Gamma_i$), so any learning scheme that relies only on $v_i$ will also cope with auction privacy requirements.

According to this, in order to increase their utilities, players simply need to pursue the direction of marginal utility increase while maintaining their bids $b_i$ at an admissible level – i.e. between 0 and $B_i$. To

this end, we propose the following exponential learning scheme:

$$\begin{cases} z_i(m+1) & = z_i(m) + \gamma_m v_i(b(m)), \\ b_i(m+1) & = \frac{B_i}{1+\exp(-z_i(m+1))}, \end{cases} \tag{4.12}$$

where $m$ is the iteration index and $\gamma_m$ is a decreasing step-size sequence. Importantly, this learning scheme is *reinforcing* because the update step of (4.12) increases with $v_i$; it is also *fully distributed* and *privacy-preserving* because the marginal utilities $v_i$ can be obtained directly from (4.11) without any exchange of private information among bidders.

The main result concerning the learning scheme (4.12) is that it converges to the unique Nash equilibrium of the auction game $\mathcal{G}$. More formally, we have:

**Theorem 8.** *If the user's benefit $\omega_i(b)$ is an affine function of the obtained resource $\varrho_i(b)$ and the algorithm's step-size sequence $\gamma_m$ satisfies $\sum_m \gamma_m = +\infty$ and $\sum_m \gamma_m^2 < +\infty$, then the learning scheme* (4.12) *converges to the unique equilibrium of the auction game $\mathcal{G}$.*

*Proof.* The main steps of the proof consist in i) deriving the mean dynamics of (4.12) in continuous-time, ii) establishing their convergence to equilibrium using the DSC condition given in (4.9), iii) obtaining the corresponding discrete-time results using the theory of stochastic approximation [158]. For simplicity, we take $B_i = 1$; the general case follows by rescaling.

Note first that the continuous-time equivalent of (4.12) is

$$\dot{z}_i = v_i, \qquad b_i = \left[1 + \exp(-z_i)\right]^{-1}, \tag{4.13}$$

or, after decoupling $b_i$ and $z_i$:

$$\dot{b}_i = b_i(1 - b_i)v_i. \tag{4.14}$$

Hence, let $r_i = 1/(\theta_i R)$, as in the proof of Theorem 7, and set

$$V(b) = \sum\nolimits_i r_i \left[ b_i^* \log \frac{b_i^*}{b_i} + (1 - b_i^*) \log \frac{1 - b_i^*}{1 - b_i} \right], \tag{4.15}$$

where $b^*$ is the unique NE of $\mathcal{G}$. Some algebra then yields

$$\dot{V} = \sum\nolimits_i r_i v_i(b) \cdot [b_i - b_i^*], \tag{4.16}$$

so, from (4.9) we have $\dot{V} \leq 0$ with equality iff $b = b^*$. This shows that $V$ is a strict Lyapunov function for (4.13), i.e., $b(t) \to b^*$.

For the discrete-time analysis, let $V_m = V(b(m))$ where $b(m)$ is the $m$-th iterate of Algorithm 2. Then, writing $V$ in terms of $z$ as

$$V = \sum\nolimits_i r_i \left[ \log(1 + e^{z_i}) - z_i b_i^* \right], \tag{4.17}$$

a first-order Taylor expansion yields:

$$V_{n+1} \leq V_m + \gamma_m \sum\nolimits_i r_i v_i(b(m)) \cdot [b_i(m) - b_i^*] + \mathcal{O}(\gamma_m^2), \tag{4.18}$$

where the $\mathcal{O}(\gamma_m^2)$ remainder is uniformly bounded by $\frac{1}{2} M \gamma_m^2$ for some $M > 0$ (simply note that the marginal utilities $v_i$ are bounded on $\mathcal{B}$). Now, if the iterates $b(m)$ stay a bounded distance away from $b^*$, (4.9) shows that we have $V_{m+1} \leq V_m - \gamma_m c + \mathcal{O}(\gamma_m^2)$ for some $c > 0$. Letting $m \to \infty$ and telescoping, we obtain the contradiction $V_m \to -\infty$ (note that $V \geq 0$ by construction and $\sum_m \gamma_m^2 < \sum_m \gamma_m = +\infty$ by assumption), so $b(m)$ must come arbitrarily close to $b^*$ infinitely often. Since $b(m)$ is a stochastic approximation of the mean dynamics (4.13) in the sense of [158], convergence to the unique equilibrium follows from Theorem 6.9 in [158]. □

### 4.1.3 Numerical Analysis

In this section, we provide some relevant numerical results that illustrate the dynamics of our auction-based resource management scheme under different scenarios and cost functions. We consider $N_C = 30$ Controllers whose interest factors are randomly generated in the interval $[0, 1]$. Also, unless explicitly stated otherwise, we set the maximum admissible bid of each Controller to $B_i = B = 1$ for all $i \in \mathcal{N}_\mathcal{C}$. The results have been obtained by executing a number of simulations such that all statistical results are stated at a 95% confidence level [159]. Confidence intervals are not shown for the sake of illustration.

#### 4.1.3.1 Flow Table Auction

At the beginning of the auction, we assume that each Controller is provided with $L_i = 200$ flow table entries (and thus, rules) that can be stored. Also, the Zipf distribution parameter is set to the realistic value of $\alpha = 0.7$ [155] and we assume that each flow entry occupies a unitary cache size, i.e., $f = 1$.

In Fig. 4.1 we show the per-user average hit probability as a function of the amount $R$ of resources, i.e., the cache size, sold by the auctioneer when different flow distributions and cost function models are considered. Since we have assumed $f = 1$, $R$ is expressed in number of flow entries. When the amount of the available cache size $R$ is high, Controllers are likely to obtain a large number of flow table entries at the end of the auction, which results in high values of the hit probability. Therefore, upon increasing the cache size, Fig. 4.1 shows that the hit probability increases as well. Also, if the cost parameter $\lambda$ is high ($\lambda = 5$), Controllers that are not interested in the resource experience high costs and low benefits. Thus, only Controllers that are interested in the resource submit their individual bids. Accordingly, interested Controllers improve their hit probability, while non

interested Controllers do not obtain any additional flow entries and their hit probability is the same as the one they had at the beginning of the auction. Therefore, the system's average hit probability is low.

On the contrary, when $\lambda$ is low ($\lambda = 1$) or equal to zero, even non interested Controllers are likely to submit their bids, which allows them to share the available resources among all participating Controllers. All Controllers obtain additional space to store their flow entries. Thus, the average hit probability in the two latter cases, is higher than the one achieved in the case $\lambda = 5$. Furthermore, Fig. 4.1 also shows that the average hit probability in the LC case is higher than (or equal to) that achieved in the NLC case. Finally, a higher average hit probability is achieved when flows are Zipf distributed. Zipf distribution allows to prioritize flows that are expected to arrive with a higher probability. Accordingly, popular flows are likely to have a matching rule in the flow tables. On the contrary, the arrival rate of uniformly distributed flows is hard to be predicted and the only way to achieve high hit probability consists in increasing the cache size put up for auction.

Fig. 4.2 shows the average hit probability as a function of the ratio $L_i/N_i$ when different cost function models are considered and $\lambda = 1$. First we note that the hit probability at the end of the auction is always higher than that obtained when no auction mechanisms are considered and the resource is statically allocated to Controllers. It follows that by putting up for auction unallocated cache resources, it is possible to improve SDNs performance. Recall that $L_i$ is the number of flow table entries that are already assigned at the beginning of the auction, and $N_i$ is the number of flows for each Controller. Accordingly, by increasing the ratio $L_i/N_i$, the hit probability increases as well. As already shown in Fig. 4.1, higher average hit probabilities are achieved when flows are distributed according to a Zipf law.

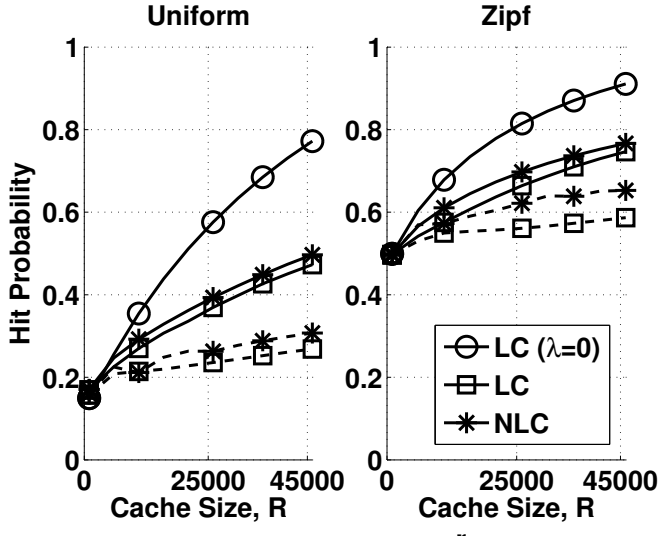



***Figure 4.1:*** *Average hit probability as a function of the cache size $R$ for different cost function models (Solid lines: $\lambda = 1$; Dashed lines: $\lambda = 5$).*

***Table 4.1:*** *Simulation parameter settings.*

| Bidder | $B_i$ | $\theta_i$ |
|---|---|---|
| $C_1$ | 0.5013 | 0.4923 |
| $C_3$ | 0.9976 | 0.9727 |
| $C_6$ | 0.8944 | 0.7391 |
| $C_8$ | 0.39 | 0.0319 |
| $C_{13}$ | 0.3433 | 0.7111 |
| $C_{23}$ | 0.5523 | 0.2815 |
| $C_{24}$ | 0.9791 | 0.7311 |

#### 4.1.3.2 Bandwidth Auction

Fig. 4.3 illustrates the impact of the cost function on the Controllers' bidding strategies. Specifically, we show how the average normalized bid strategy at the NE defined as $1/N_C \cdot \sum_{i \in \mathcal{N}_C} b_i^*/B_i$ varies as a function of the cost parameter $\lambda$ under different cost function models. As expected, when $\lambda = 0$ each Controller submits a bid whose value is

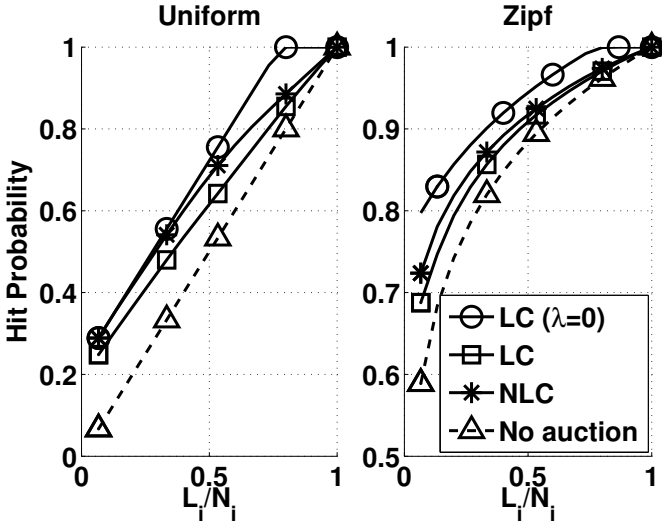



***Figure 4.2:*** *Average hit probability as a function of the ratio $L_i/N_i$ for different cost function models and $\lambda = 1$.*

equal to its own maximum admissible bid, i.e., $b_i = B_i$ for all $i \in \mathcal{N}_\mathcal{C}$. On the other hand, when LC and NLC cost models are considered, an increase in the value of $\lambda$ causes an increase in the costs experienced by each Controller. Accordingly, bids submitted by Controllers decrease as $\lambda$ increases. Also, if the amount $R$ of resources that are leased out is small (solid lines), submitted bids are smaller than those submitted when larger values of $R$ are considered (dashed lines). This latter result is caused by the fact that when $R$ is small, only interested Controllers submit high bids. On the contrary, a larger amount of available resources $R$ leads non interested Controllers to submit non-zero bids to obtain a small fraction of the considered resource.

To illustrate the bidders' behavior and dynamics, in Fig. 4.4 we show the Controllers' bidding strategies for different cost function models. For simplicity, we only show the behavior of a subset of the participating Controllers. More specifically we consider Controllers $C_i$ with $i \in \{1, 3, 6, 8, 13, 23, 24\}$ and we assume $\lambda = 0.2$. For illustration

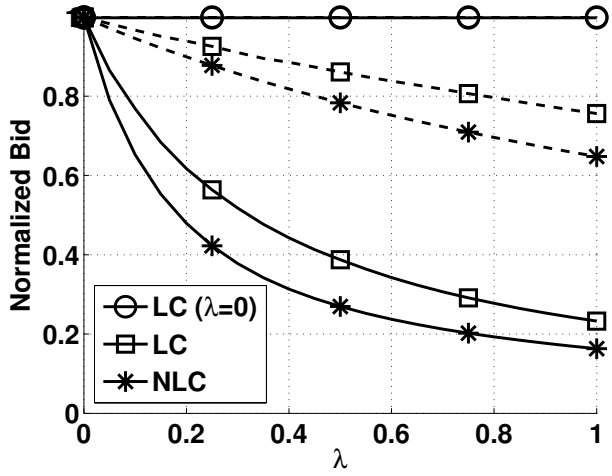



***Figure 4.3:*** *Average normalized bid as a function of the cost parameter $\lambda$ for different cost function models (Solid lines: $R = 10$; Dashed lines: $R = 100$).*

purposes, we assume that the maximum admissible bid $B_i$ and the interest factor $\theta_i$ are randomly generated in the interval $[0, 1]$. The considered simulation setup is reported in Table 4.1. Fig. 4.4 shows that when no cost is charged to bidders (i.e., $\lambda = 0$ case), all bidders submit non-zero bids. On the contrary, no bids are submitted by $C_8$ and $C_{23}$ in the LC and NLC cases. In fact, such controllers are not interested in buying additional space in the flow table, i.e., $\theta_8$ and $\theta_{23}$ are small. Controllers such as $C_3$, $C_6$ and $C_{24}$ are interested in participating in the auction and have high budgets. Therefore, they submit $b_i = B_i$ independently of the actual cost model. On the contrary, even though $C_1$ and $C_{13}$ are interested in buying additional bandwidth, their budget is low. Accordingly, their submitted bids decrease when they experience costs as in the LC and NLC cases.

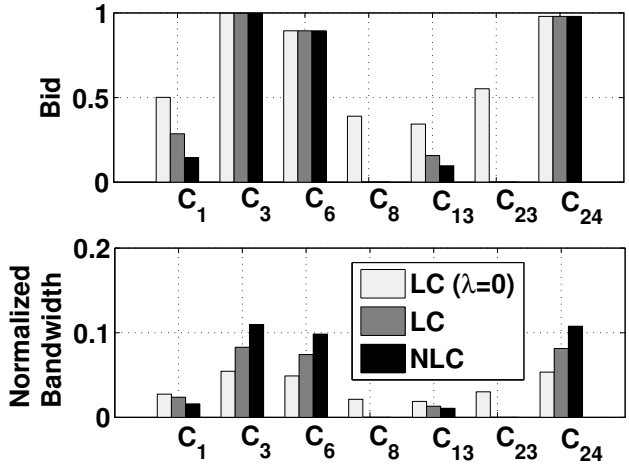



***Figure 4.4:*** *Bid strategies and normalized allocated bandwidth for different cost schemes.*

#### 4.1.3.3 Convergence Evaluation

As stated in Theorem 8, the convergence to the unique NE of Algorithm 2 is guaranteed in the case a variable step-size (e.g., $\gamma_m = 1/m^\beta$ with $\beta \in (0.5, 1]$) is used. However, in the following we show that the learning process still converges to the NE even if a fixed step-size $\gamma_m$ is used. For illustrative purposes, in the following we only consider the bandwidth auction case, but the same results hold for the flow table auction case as well.

Figure 4.5 illustrates how fast the game converges to the unique NE by means of the proposed distributed learning procedure. Specifically, the figure shows the Euclidian distance between the actual bidding profile at each iteration and the NE in case of LC and NLC schemes. As expected, the convergence rate is slow when a variable step-size is considered. Also, by increasing the value of the parameter $\beta$, it is possible to improve the convergence rate of the proposed learning procedure under a varible step-size scheme. Otherwise, if a fixed step-

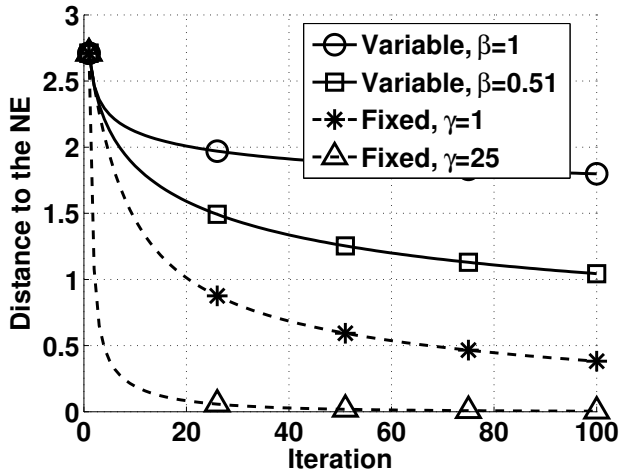



***Figure 4.5:*** *Distance to the NE for different step-size schemes.*

size is considered, the convergence rate increases. More specifically, the higher the step-size, the faster the convergence rate is.

## 4.2 Services Management in the Core Network

In this section, we propose a distributed service management for the CN. Specifically, we aim to provide a framework for service provisioning by exploiting virtualization and softwarization of network functions. Accordingly, we exploit a NFV approach where we aim to distribute orchestration and resource allocation tasks, while limiting the work of the Orchestrator to coordinate and facilitate them.

More specifically, third-parties in the CN and/or in the cloud, in the following referred to as *VNF Servers*, can host and execute VNFs to process network flows, so participating in the *VNF Market* as sellers. VNFs are requested by both tenants and their users. In the following of this section, we refer to the entities which request VNFs

as the *VNF Customers*[4]. VNF Servers decide the price to be applied to the customers, and the bandwidth to request to the network operator to provide the service. VNF Customers, on the other hand, according to the price specified by each server, and the corresponding expected performance in terms of both experienced latency and received bandwidth, choose one server for each VNF. In this way the task of associating each flow to a VNF Server is not decided by the Orchestrator, but autonomously and in a distributed way, as a consequence of the interaction between customers and VNF Servers. In this thesis, we assume that VNF Servers are located in the CN. However, our formulation also holds for the more general case where VNF Servers are in both the CN and the cloud.

To model interactions between VNF Servers and VNF Customers, we exploit analytical tools derived from the game theory. Specifically, we define a two-stage Stackelberg game where VNF Servers act as the leaders of the game, and VNF Customers as the followers. VNF Servers have conflicting interests among themselves, as their objective is to individually and selfishly maximize a utility function. Also, as commonly assumed in multi-player markets, VNF Servers are expected not to cooperate with each other, and do not exchange any information with other competitors. Therefore, their interactions are modeled by non-cooperative game theory. Instead, customers are influenced by social behavior, i.e., they observe other customers' decisions and replicate those decisions if this is expected to improve their benefit. Therefore, interactions among VNF Customers are modeled by using the replicator dynamics from Evolutionary Game Theory (EGT).

In the following of this section, we propose a game-theoretic framework that models interactions between VNF Servers and VNF Customers is introduced. The existence and uniqueness of the SE of the

[4]In the following the terms VNF Customers and customers are used interchangeably.

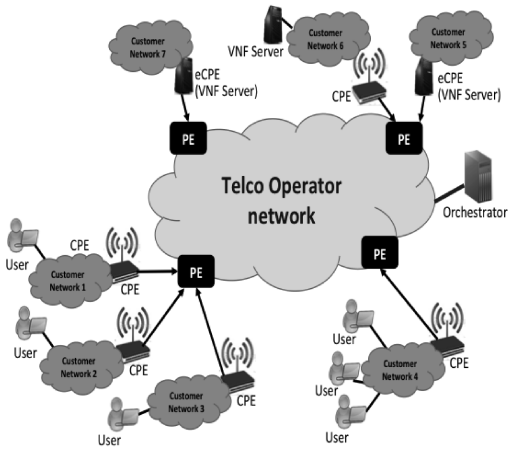



***Figure 4.6:*** *The reference network scenario.*

game are discussed, and a learning procedure that provably converges to the SE is proposed. Also, closed-form results at the SE are provided in the special case when two VNF Servers compete with each other to provide a given VNF.

### 4.2.1 System Model and Problem Formulation

The reference scenario we consider is sketched in Fig. 4.6. It consists of a CN that provides VNF Customers with virtualized network functions (VNFs) according to the NFV paradigm.

The main roles are played by the Orchestrator, the VNF Servers, and the VNF Customers.

*VNF Customers* are the entities that generate flows and requests VNFs. As shown in Fig. 1.1, Customers access the CN through RAN and CN gateways. Specifically, the RAN is connected to the CN through RAN gateways also referred to as *Customer Premise Equipment* (CPE) devices. Instead, access to the CN is provided through CN gateways also referred to as *Provider Edge* (PE) nodes.

*VNF Servers* are NFV-compliant nodes [160, 161] in the CN that are able to run VNFs, to obtain economic benefits. The price that is applied for each VNF is autonomously decided by each VNF Server. Unlike the NFV paradigm where VNFs run on servers owned by the Orchestrator or in public data centers, here we consider the case where private third-parties in the CN are allowed to run VNFs. A VNF Server can be either a stand-alone computer (e.g., a volume server), whose resources are partially or totally dedicated to run VNFs, or an enhanced CPE (eCPE) node. As described in [162], these are CPE nodes that are able to run VNFs in a virtualized environment. Besides the hardware facilities, VNF Servers need an amount of bandwidth that is provided them by the TO.

A very important role in the system is played by the *Orchestrator*, which is in charge of management of the whole system. It runs on a dedicated server and communicates with all the NFV nodes through the CN. The main tasks performed by the Orchestrator are:

- Exposing the list of the supported VNFs;
- Running, migrating and halting VNF instances on the VNF Servers;
- Assigning a slice of bandwidth to the VNF Servers according to their bandwidth requests;
- Providing each customer with the list of those VNF Servers that are running a given VNF, including information regarding the price applied by each VNF Server and the relevant performance parameters, in terms of experienced latency and received bandwidth.
- Setting the flow table of the SDN switches in such a way that VNF Customers flows traverse the chosen VNF Servers.

As regards the latency, it is evident that each VNF Server is characterized by almost the same performance latency parameter for all the customers that enter the CN through the same PE node, or through PE nodes that are very close to each other. So, in the following we will use the term *Customer Group* to indicate the set of VNF Customers that are characterized by the same latency from the same VNF Servers. Concerning the bandwidth provided by a VNF Server to each customer flow, it depends on both the amount of bandwidth the VNF Server requests to the network and the number of customers flows using its VNFs.

Specifically, the system works as follows:

- For each VNF, the relevant VNF Server autonomously decides the price to be applied to the customers of each Customer Group. The amount of bandwidth needed to provide a given VNF to the whole set of customers being served is decided by playing a game with all the other VNF Servers that are available to run that function, by taking into account that it will be shared among the flows of all the VNF Customer connected to that VNF Server.

- For each VNF, each VNF Customer chooses one VNF Server by playing another game with all the customers that belong to its Customer Group.

In the following, for the sake of simplicity, we will focus on one individual VNF, namely $f$. Let $\mathcal{V}_\mathcal{S}$ be the set composed by $M$ VNF Servers that provide $f$, and $\mathcal{V}_\mathcal{U}$ a Customer Group interested in that function. Let $d_i$ be the latency encountered by the flows of the customers belonging to $\mathcal{V}_\mathcal{U}$ to reach the VNF Server $i \in \mathcal{V}_\mathcal{S}$. For each customer flow traversing a given VNF Server, this Server has to allocate a given amount of computing and storage resources, and this represents a cost for the VNF Server.

Considering the generic VNF Server $i$, as in [162] we will refer to the cost of the incremental energy needed by a flow to use the resources as $c_i$. It depends on the price applied by the energy provider and the amount of available renewable energy.

Therefore, the energy cost for a VNF Server to manage all the customer flows, $C_i^{(\mathcal{F})}$, is proportional to the number of flows $n_i$, that is

$$C_i^{(\mathcal{F})} = c_i \cdot n_i \tag{4.19}$$

Another cost for the VNF Servers is due to the bandwidth that they receive from the CN according to the requests issued to the Orchestrator. Let $b_i$ be the bandwidth received by the VNF Server $i$. Also, let $p_i^{(\mathcal{B})}$ be the bandwidth-unit price applied by the Orchestrator to the VNF Server $i$. So the cost of the overall bandwidth used by the VNF Server is:

$$C_i^{(\mathcal{B})} = p_i^{(\mathcal{B})} \cdot b_i \tag{4.20}$$

On the other hand, the revenue for the VNF Server $i$ is proportional to number $n_i$ of VNF Customers that are using its VNF $f$:

$$R_i = p_i^{(\mathcal{F})} \cdot n_i \tag{4.21}$$

where $p_i^{(\mathcal{F})}$ is the price applied by this VNF Server. As said so far, the decision regarding the amount of bandwidth that each VNF Server requests to the Orchestrator is taken after playing a game with the other VNF Servers. The game, which will be described in Section 4.2.2, aims at maximizing an utility function defined as follows:

$$U_i^{(\mathcal{S})}(\mathbf{b}) = \beta_1 R_i - \beta_2 \left[ C_i^{(\mathcal{F})} + C_i^{(\mathcal{B})} \right] \tag{4.22}$$

where $\mathbf{b} = (b_1, b_2, \ldots, b_M)$ is the *bandwidth vector* that contains the bandwidth $b_i$ provided by each VNF server, and $\beta_1$ and $\beta_2$ are appropriate constants weighing the relative relevance of revenues and costs.

On the other hand, customers choose the VNF Server by taking into account the latency experienced to reach it, $d_i$, and the current price it is applying to the VNF $f$. However, the higher the number of customer flows using the same VNF Server, the lower the bandwidth allocated to each of them. With all this in mind, each VNF Customer selects the VNF Server that maximizes the following utility function [163]:

$$U_i^{(\mathcal{U})}(\mathbf{n}) = \ln\left(\alpha_1 \frac{b_i}{n_i}\right) - \alpha_2 p_i^{(\mathcal{F})} - \alpha_3 d_i \tag{4.23}$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_M)$ is the *state vector* that contains the number $n_i$ of flows served by each VNF Server in $\mathcal{V}_\mathcal{S}$; $\alpha_1$, $\alpha_2$ and $\alpha_3$ are appropriate constants that weigh the contributions to the utility function of the bandwidth received, the price applied by the VNF Server, and the latency encountered to reach that Server, respectively. In the following, we will refer to $\alpha_1$, $\alpha_2$, $\alpha_3$, $\beta_1$ and $\beta_2$ as the *weighing parameters*.

### 4.2.2 Game Model

In this section, we illustrate the proposed game-theoretic framework which models the interactions between VNF Servers and VNF Customers.

Decisions taken by VNF Servers and VNF Customers depend on both individualistic interests, e.g., maximize their own utility, and decisions taken by counterparts, e.g., opponents' strategies. For example, customers connect to either one of the available VNF Servers depending on the offered bandwidth and other relevant parameters such as proposed prices and expected communication delays. On the contrary, VNF Servers aim to maximize their revenues and are not likely to cooperate with each other. Also, their actions depend on the number of VNF Customers that are connected to them to use their

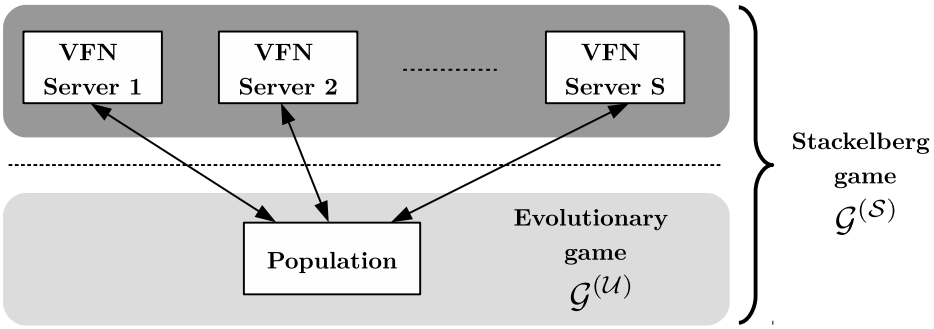



***Figure 4.7:*** *The proposed game-theoretic framework.*

VNFs.

In real scenarios, VNF Servers naturally act and make decisions by anticipating the VNF Customers. Accordingly, interactions among VNF Servers and VNF Customers can be modeled as a two-stage Stackelberg game where VNF Servers act as the *leaders* of the game and VNF Customers as the *followers*. In the addressed problem we should also consider customers that replicate other VNF Customers' decisions. Such replicative behavior naturally arises in those scenarios where multiple entities make decisions by replicating other entities' behavior [164–170].

In Section 4.2.2.1 we first define a game $\mathcal{G}^{(\mathcal{V}_\mathcal{U})}$ where we exploit EGT and replicator dynamics to model the decision-making process of VNF Customers. Then, in Section 4.2.2.2 we use non-cooperative game theory to define the game $\mathcal{G}^{(\mathcal{V}_\mathcal{S})}$ which models competitive interactions among the VNF Servers. The considered game-theoretic model and its hierarchical structure are shown in Fig. 4.7.

#### 4.2.2.1 Evolutionary game $\mathcal{G}^{(\mathcal{V}_\mathcal{U})}$ among VNF Customers

Let $N_\mathcal{V}$ be the number of VNF Customers in the system. We assume that each customer knows how many VNF Customers are connected to each VNF Server at a given time. Also, we assume that each VNF

Customer can connect to only one VNF Server at a time.

Each VNF Customer is intrinsically selfish as it makes decisions with the aim of maximizing its own utility $U_i^{(\mathcal{U})}$, as defined in (4.23). However, the higher the number $n_i$ of VNF Customer connected to the $i$-th VNF Server, the lower the utility $U_i^{(\mathcal{U})}$ of that customer. Therefore, the decision-taking process of each customer is also influenced by decisions taken by other VNF Customers in $\mathcal{V}_\mathcal{U}$. Also, if a VNF Customer is aware that another VNF Customer is achieving a better utility, it can decide to imitate that customer and migrate to the same VNF Server to which that VNF Customer is connected. In the rest of this analysis, we refer to this phenomenon as *imitation behavior*.

Imitation behavior often arises when considering interactions among entities that rationally try to maximize their benefit by imitating other entities' decisions that provide better benefit. For example, imitation is at the basis of a variety of decision making problems in both wired [165–168] and wireless networks [169, 170] that are often modeled by exploiting theoretical tools from evolutionary game theory.

In line with a vast body of literature, we consider the well-known and widely used *replicator dynamics* [171] as the imitation dynamics which describe the interactions among VNF Customers. Accordingly, we define the evolutionary game $\mathcal{G}^{(\mathcal{V}_\mathcal{U})}$ as follows:

- *Population*: it consists of the set of the $N_\mathcal{V}$ VNF Customers.

- *Strategy*: it is defined as the choice of the VNF Server $i \in \mathcal{V}_\mathcal{S}$ to whom each VNF Customer in the population decides to connect; the *strategy set* of each VNF Customer is $\mathcal{V}_\mathcal{S}$.

- *Utility*: the utility, or benefit, achieved by each VNF Customer connected to the VNF Server $i \in \mathcal{V}_\mathcal{S}$ is equal to $U_i^{(\mathcal{U})}$ as defined in (4.23).

We can now define the *replicator equation* that describes how the number of VNF Customers in the population that connects to available VNF Servers varies

$$\dot{n}_i = n_i \left[ U_i^{(\mathcal{U})}(\mathbf{n}) - \frac{1}{N_{\mathcal{V}}} \sum_{j \in \mathcal{V}_{\mathcal{S}}} n_j U_j^{(\mathcal{U})}(\mathbf{n}) \right] \tag{4.24}$$

where $\mathbf{n}$ is the state vector of the replicator dynamics and the generic element $n_i \in \mathbf{n}$ denotes the number of VNF Customers which have chosen as a strategy to connect to the $i$-th VNF Server.

The first term in the right-hand side of (4.24) represents the utility of a VNF Customer that connects to the $i$-th VNF Server, while the second term represents the average utility of the population which depends on the current distribution $\mathbf{n}$ of the population. Therefore, the growth rate $\dot{n}_i/n_i$ of the number of VNF Customers connected to the $i$-th VNF Server is equal to the difference between the benefit when choosing the strategy $i$, and the average benefit of the whole population.

A general result from EGT shows that an equilibrium point for the replicator dynamics is a fixed point of the replicator dynamics such that all VNF Customers experience the same benefit, i.e., $U_i^{(\mathcal{U})} = U_j^{(\mathcal{U})}$ for all $i, j \in \mathcal{V}_{\mathcal{S}}$.

In Proposition 6, we show that the replicator equation (4.24) admits a unique solution for any bandwidth vector $\mathbf{b}$. Furthermore, we characterize the equilibrium point by deriving the resulting state vector $\mathbf{n}^*$ at the equilibrium. To this purpose, let us define $\phi_{i,j}$ as follows:

$$\phi_{i,j} = e^{\left[\alpha_2 \left(p_i^{(\mathcal{F})} - p_j^{(\mathcal{F})}\right) + \alpha_3 (d_i - d_j)\right]} \tag{4.25}$$

From (4.25), it can be easily shown that the following relationships

hold for all $i, j, k \in \mathcal{V}_{\mathcal{S}}$

$$\phi_{i,i} = 1, \quad \phi_{i,j} = 1/\phi_{j,i}, \quad \text{and} \quad \phi_{k,j} = \frac{\phi_{1,j}}{\phi_{1,k}} \tag{4.26}$$

**Proposition 6.** *For any given bandwidth vector* $\mathbf{b}$*, the replicator equation* (4.24) *admits a unique evolutionary equilibrium* $\mathbf{n}^*$*. Also, the number of VNF Customers* $n_i^*$ *connected to the generic VNF Server* $i \in \mathcal{V}_{\mathcal{S}}$ *at the equilibrium point can be derived as follows:*

$$n_i^* = N_{\mathcal{V}} \frac{b_i}{\sum_{j \in \mathcal{V}_{\mathcal{S}}} b_j \phi_{i,j}} \tag{4.27}$$

*where* $b_i \in \mathbf{b}$*.*

*Proof.* The replicator equation can be reduced to an equivalent system of ordinary differential equations (ODEs). Thus, to show that the replicator dynamics admits a unique equilibrium point, it suffices to show that the right-hand side of the mean dynamic in (4.24) is Lipschitz continuous [172]. However, from (4.24), we have that the function in the right-hand side of (4.24) is continuously differentiable. This is a sufficient condition for Lipschitz continuity. Thus, uniqueness of the equilibrium is guaranteed by results contained in [172].

Now, in order to determine the unique equilibrium, it is well known that it is reached when $\dot{n}_i = 0$ [172]. Such condition implies that $U_i^{(\mathcal{U})} = U_j^{(\mathcal{U})}$ for all $i, j \in \mathcal{V}_{\mathcal{S}}$, i.e., all VNF Customers receive the same benefit. Accordingly, we can build a system of equations with $N_{\mathcal{V}}(N_{\mathcal{V}} - 1)/2$ equations that can be solved by exploiting the relationship $N_{\mathcal{V}} = \sum_{i \in \mathcal{V}_{\mathcal{S}}} n_i$. Thus, after some easy analytical derivations, we obtain the result in (4.27).

For the sake of illustration, in the following we show how to derive (4.27) when $N_{\mathcal{V}} = 2$. However, the more general case can be treated in a similar way. From (4.23) and (4.25), and by imposing $U_1^{(\mathcal{U})} = U_2^{(\mathcal{U})}$

we get

$$\ln\left(\frac{b_1}{b_2}\frac{n_2}{n_1}\right) = \ln(\phi_{1,2}) \tag{4.28}$$

Recall that $N_{\mathcal{V}} = n_1 + n_2$. Thus, we get

$$n_1^* = N_{\mathcal{V}}\frac{b_1}{b_1 + b_2\phi_{1,2}} \quad \text{and} \quad n_2^* = N_{\mathcal{V}}\frac{b_2}{b_2 + b_1\phi_{2,1}} \tag{4.29}$$

which is a specific case of (4.27). □

#### 4.2.2.2 Stackelberg game $\mathcal{G}^{(\mathcal{V}_S)}$ between VNF Servers and VNF Customers

As already discussed before, VNF Servers act as leaders of the game between VNF Servers and VNF Customers. Also, in Proposition 6 we have derived the distribution $\mathbf{n}^*$ of the population at the equilibrium of the replicator dynamics.

For the sake of notation, let us first define the two following auxiliary variables

$$\tilde{p}_i = N_{\mathcal{V}}\left(\beta_1 p_i^{(\mathcal{F})} - \beta_2 c_i\right) \tag{4.30}$$

and

$$\pi_i = \beta_2 p_i^{(\mathcal{B})} \tag{4.31}$$

Accordingly, we can incorporate (4.21), (4.27), (4.30) and (4.31) in (4.22) to rewrite the utility function $U_i^{(\mathcal{S})}$ of the generic VNF Server $i \in \mathcal{V}_{\mathcal{S}}$ as follows:

$$U_i^{(\mathcal{S})}(\mathbf{b}) = \tilde{p}_i\frac{b_i}{\sum_{k\in\mathcal{V}_{\mathcal{S}}} b_k\phi_{i,k}} - \pi_i b_i \tag{4.32}$$

We define the non-cooperative game $\mathcal{G}^{(\mathcal{V}_S)}$ as follows:

- *Player set*: it consists of the set $\mathcal{V}_{\mathcal{S}}$ of VNF Servers.
- *Strategy*: it is defined as the amount of bandwidth $b_i$ to be re-

quested to the TO to serve its connected customers. We assume that such amount of bandwidth is bounded by $B_i$; the strategy set is $\mathcal{B} = \prod_{i \in \mathcal{V}_\mathcal{S}} \mathcal{B}_i$, where $\mathcal{B}_i = [0, B_i]$ and $\prod$ identifies the Cartesian product.

- *Utility*: the utility of each VNF Server $i \in \mathcal{V}_\mathcal{S}$ is equal to $U_i^{(\mathcal{S})}$ as defined in (4.32).

By calculating the first-order derivative of (4.32), it can be easily shown that $\tilde{p}_i \leq 0$ leads to a non-positive first-order derivative of the utility function $U_i^{(\mathcal{S})}$. In other words, the best strategy for the $i$-th VNF Server is not to participate in the game by choosing $b_i = 0$. Thus, those VNF Servers with $\tilde{p}_i \leq 0$ exit the game and they can be removed from the player set $\mathcal{V}_\mathcal{S}$. Accordingly, without loss of generality, in our model we assume that the player set $\mathcal{V}_\mathcal{S}$ is composed by only those VNF Servers such that $\tilde{p}_i > 0$.

In the following, we analyze the Stackelberg game $\mathcal{G}^{(\mathcal{V}_\mathcal{S})}$ and provide useful results about its equilibrium points, referred to as SEs.

**Definition 4.** *Let* $\mathbf{b}^* \in \mathcal{B}$. *The strategy profile* $(\mathbf{b}^*, \mathbf{n}^*)$ *is a SE for the game* $\mathcal{G}^{(\mathcal{V}_\mathcal{S})}$ *if for all* $\mathbf{b} \in \mathcal{B}$ *and* $i \in \mathcal{V}_\mathcal{S}$, *we have*

$$U_i^{(\mathcal{S})}(\mathbf{b}^*, \mathbf{n}^*) \geq U_i^{(\mathcal{S})}(\mathbf{b}, \mathbf{n}^*)$$

*where* $\mathbf{n}^*$ *is defined as in* (4.27).

In Proposition 7, we prove that the game $\mathcal{G}^{(\mathcal{V}_\mathcal{S})}$ admits a unique SE.

**Proposition 7.** *The game* $\mathcal{G}^{(\mathcal{V}_\mathcal{S})}$ *admits a unique SE.*

*Proof.* The main steps of the proof are as follows. First, we prove the existence of the equilibrium by exploiting concavity properties of VNF Servers' utility functions in (4.32). Then, we show that the

DSC property holds. The DSC property implies that VNF Servers experience diminishing returns along any direction, i.e., along all $b_i \in \mathbf{b}$. Finally, we exploit results contained in [146, 157] to prove that a unique equilibrium exists.

Let the *marginal utility* $v_i(\mathbf{b})$ of each player $i \in \mathcal{V}_\mathcal{S}$ be defined as $v_i(\mathbf{b}) = \frac{\partial U_i^{(\mathcal{S})}(\mathbf{b})}{\partial b_i}$. Therefore, from (4.32) it follows that the marginal utility of the generic VNF Server $i$ is

$$v_i(\mathbf{b}) = \tilde{p}_i \frac{\sum_{k \in \mathcal{V}_\mathcal{S}, k \neq i} b_k \phi_{i,k}}{\left(\sum_{k \in \mathcal{V}_\mathcal{S}} b_k \phi_{i,k}\right)^2} - \pi_i \tag{4.33}$$

where $\tilde{p}_i$ is defined in (4.30). Also, let $\mathbf{b}_{-i}$ be the bandwidth vector of all players except $i$, i.e., $\mathbf{b}_{-i} = (b_j)_{j \in \mathcal{V}_\mathcal{S},\ j \neq i}$ with $b_j \in \mathbf{b}$.

To show that the DSC property holds, it must be shown that: i) $U_i^{(\mathcal{S})}(\mathbf{b})$ is strictly concave in $b_i$; ii) $U_i^{(\mathcal{S})}(\mathbf{b})$ is convex in $\mathbf{b}_{-i}$; and iii) the function $\rho(\mathbf{b}, \mathbf{r})$ defined as

$$\rho(\mathbf{b}, \mathbf{r}) = \sum_{i \in \mathcal{V}_\mathcal{S}} r_i U_i^{(\mathcal{S})}(\mathbf{b}) \tag{4.34}$$

is concave in $\mathbf{b}$ for some $\mathbf{r} = (r_1, r_2, \ldots, r_M)$ such that $r_i > 0\ \forall i \in \mathcal{V}_\mathcal{S}$.

From (4.32), it can be shown that property i) holds as $U_i^{(\mathcal{S})}(\mathbf{b})$ is defined as the difference between a strictly concave function and a concave function. To prove ii), it suffices to note that the Hessian matrix of $U_i^{(\mathcal{S})}(\mathbf{b})$ has all non-negative eigenvalues, i.e., the Hessian matrix is positive semidefinite.

Let $r_i = 1/\tilde{p}_i$ for all $i \in \mathcal{V}_\mathcal{S}$. Accordingly, (4.34) can be rewritten

as follows:

$$\rho(\mathbf{b},\mathbf{r}) = \sum_{i\in\mathcal{V}_S} \frac{b_i}{b_i + \sum_{j\neq i} b_j\phi_{i,j}} - \sum_{i\in\mathcal{V}_S} r_i\pi_i b_i = \\ = \frac{b_1}{b_1 + \sum_{j\neq 1} b_j\phi_{1,j}} + \sum_{k\neq 1} \frac{b_k}{b_k + \sum_{j\neq k} b_j\phi_{k,j}} - \sum_{i\in\mathcal{V}_S} r_i\pi_i b_i \qquad (4.35)$$

From (4.26), we have that

$$\rho(\mathbf{b},\mathbf{r}) = \\ = \frac{b_1}{b_1 + \sum_{k\neq 1} b_k\phi_{1,k}} + \sum_{k\neq 1} \frac{b_k\phi_{1,k}}{b_1 + \sum_{j\neq 1} b_j\phi_{1,j}} - \sum_{i\in\mathcal{V}_S} r_i\pi_i b_i = \\ = 1 - \sum_{i\in\mathcal{V}_S} r_i\pi_i b_i \qquad (4.36)$$

Observe that $\rho(\mathbf{b},\mathbf{r})$ is a concave function in $\mathbf{b}$ as required in iii). Therefore, we have that DSC property holds and the general theory in [146, 157] ensures the uniqueness of the equilibrium. □

In (4.32), we use the equilibrium condition in (4.27). Therefore, interactions between VNF Customers (i.e., the followers) and VNF Servers (i.e., the leaders) modeled through the game $\mathcal{G}^{(\mathcal{V}_S)}$ produce a unique SE $(\mathbf{b}^*, \mathbf{n}^*)$. However, recall that VNF Servers compete with each other in the Stackelberg game. Accordingly, the strategy profile $\mathbf{b}^*$ discussed above also represents a NE [146] for the competitive game among VNF Servers.

In Proposition 7, we have shown that the game $\mathcal{G}^{(\mathcal{V}_S)}$ admits a unique equilibrium. As will be shown in Section 4.2.3, we are able to fully characterize and derive closed-form expressions for the equilibrium of the game where $M = 2$ VNF Servers compete with each other. Unfortunately, in the general case where $M > 2$ we are not able to find a proper characterization of the equilibrium point and provide closed-form expressions. Therefore, we need to provide a robust mech-

anism to allow VNF Servers to individually reach the equilibrium of the game. Accordingly, in the following we propose an *exponential reinforcing learning* [173] procedure, which provably converges to the unique equilibrium of the game.

For each VNF Server $i \in \mathcal{V}_{\mathcal{S}}$, we define the following learning procedure

$$\begin{cases} z_i(m+1) = z_i(m) + \gamma_m v_i\left(\mathbf{b}(m)\right) \\ b_i(m+1) = B_i \frac{e^{z_i(m+1)}}{1+e^{z_i(m+1)}} \end{cases} \tag{4.37}$$

where $m$ represents the iteration index, $\mathbf{b}(m)$ is the bandwidth vector at iteration $m$, and $\gamma_m$ is the step-size of the learning procedure whose importance will be explained later.

In the following Proposition 8, we show that the proposed exponential reinforcing learning procedure converges to the equilibrium of the game.

**Proposition 8.** *Let $\gamma_m$ be the step-size of the learning procedure. If $\sum_m \gamma_m^2 < \sum_m \gamma_m = +\infty$. For any feasible initial condition in $\mathcal{B}$, Algorithm 2 always converges to the unique SE of $\mathcal{G}^{(\mathcal{V}_{\mathcal{S}})}$.*

The proof consists in showing that i) the *mean dynamic* of (4.37), i.e., its continuous-time version, converges to the equilibrium of the game as time goes to infinity, and ii) (4.37) is an *asymptotic pseudo-trajectory* (APT) [174] for the continuous-time version of (4.37). For a detailed and rigorous proof, we refer the reader to Appendix C.

From Proposition 8, we have that any variable step-size rule in the form $\gamma_m = 1/m^{\beta}$ with $\beta \in (0.5, 1]$ will converge to the unique SE of the game $\mathcal{G}^{(\mathcal{V}_{\mathcal{S}})}$.

To compute $b_i(m+1)$ in (4.37), each $i \in \mathcal{V}_{\mathcal{S}}$ is required to evaluate $v_i\left(\mathbf{b}(m)\right)$ in (4.33). Note that $v_i\left(\mathbf{b}(m)\right)$ depends on the term $\sum_{k \in \mathcal{V}_{\mathcal{S}}} b_k(m)\phi_{i,k}$ which can be computed by each VNF Server only if full information on other VNF Servers' parameters and requests is available. Unfortunately, such an assumption is unrealistic and un-

feasible in several application scenarios where VNF Servers compete with each other and do not exchange any information. However, the Orchestrator collects all VNF Servers' requests and has full access to their parameters (e.g., $d_i$, $p_i^{(\mathcal{F})}$, etc...) by assumption. Accordingly, at each iteration and for each VNF Server $i \in \mathcal{V}_\mathcal{S}$, the Orchestrator is able to compute the overall sum $\sum_{k\in\mathcal{V}_\mathcal{S}} b_k(m)\phi_{i,k}$ and send it to the corresponding $i$-th VNF Server. Also, note that the $i$-th VNF Server cannot extract any private information on other VNF Servers from the sum $\sum_{k\in\mathcal{V}_\mathcal{S}} b_k(m)\phi_{i,k}$.

Thus, under the above assumptions the learning procedure (4.37) can be implemented in a privacy-preserving and distributed fashion.

### 4.2.3 The two VNF Server case

In this section, we consider the special case where two VNF Servers provide the same function $f$ to VNF Customers in the system, i.e., $\mathcal{V}_\mathcal{S} = \{S_1, S_2\}$.

First, we provide theoretical results that show the stability of the equilibrium point $\mathbf{n}^*$ in (4.29). In particular, in Proposition 9 we have shown that the equilibrium point $\mathbf{n}^*$ is an evolutionary stable strategy (ESS). ESS is a classical concept in evolutionary game theory that expresses the robustness of a given equilibrium strategy against mutation in the population. Specifically, a population in which all members play an ESS strategy is resistant to invasion by a small group of mutants who play an alternative strategy [172].

**Proposition 9.** *The population distribution* $\mathbf{n}^*$ *is an ESS.*

*Proof.* Let $s_i^* = \frac{n_i^*}{N_\mathcal{V}}$ be the fraction of the population that is connected to the $i$-th VNF Server at the equilibrium and $\mathbf{s}^* = (s_1^*, s_2^*)$. Let $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$ be a population distribution such that $\sum_{i\in\mathcal{V}_\mathcal{S}} \sigma_i = 1$ and $\boldsymbol{\sigma} \neq \mathbf{s}^*$. To show that $\mathbf{n}^*$ is ESS, according to [172] it suffices to show

that

$$(\sigma_1 - s_1^*)U_1^{(\mathcal{S})}(\boldsymbol{\sigma}) + (\sigma_2 - s_2^*)U_2^{(\mathcal{S})}(\boldsymbol{\sigma}) < 0 \tag{4.38}$$

where $U_1^{(\mathcal{S})}(\boldsymbol{\sigma})$ and $U_2^{(\mathcal{S})}(\boldsymbol{\sigma})$ are the utility functions of VNF Servers $S_1$ and $S_2$, respectively.

By exploiting the two relationships $\sigma_2 = 1 - \sigma_1$ and $s_2^* = 1 - s_1^*$, we get

$$(\sigma_1 - s_1^*) \ln \left( \frac{s_1^*(1-\sigma_1)}{\sigma_1(1-s_1^*)} \right) < 0 \tag{4.39}$$

By assumption, $s_1^* \neq \sigma_1$. Thus, the relationship (4.39) always hold. Accordingly, condition (4.38) is satisfied and the equilibrium point $\mathbf{n}^* = (n_1^*, n_2^*) = (s_1^* N_{\mathcal{V}}, s_2^* N_{\mathcal{V}})$ is ESS. □

In the following, we derive closed forms for the SE $(\mathbf{b}^*, \mathbf{n}^*)$ of the game $\mathcal{G}^{(\mathcal{V}_{\mathcal{U}})}$. More specifically, we have already derived closed forms for $\mathbf{n}^*$ in (4.29). Therefore, we still need to find a closed-form solution for $\mathbf{b}^*$.

Let $k_i$ be defined as

$$k_i = \frac{\tilde{p}_i}{\pi_i} \tag{4.40}$$

Let $\mathrm{BR}_j(b')$ denote the *best response* of VNF Server $j \in \mathcal{V}_{\mathcal{S}}$ to strategy $b_i$ of VNF Server $i \in \mathcal{V}_{\mathcal{S}}$. Formally, we have

$$\mathrm{BR}_j(b') = \arg\max_{b_j \in \mathcal{B}_j} U_j^{(\mathcal{S})}(b', b_j) \tag{4.41}$$

Therefore, since we assumed $S = 2$, we get

$$\mathrm{BR}_1(b_2) = \sqrt{b_2 \phi_{1,2}} \left( \sqrt{k_1} - \sqrt{b_2 \phi_{1,2}} \right) \tag{4.42}$$

and

$$\mathrm{BR}_2(b_1) = \sqrt{\frac{b_1}{\phi_{1,2}}} \left( \sqrt{k_2} - \sqrt{\frac{b_1}{\phi_{1,2}}} \right) \tag{4.43}$$

By substituting (4.42) in (4.43), we obtain

$$b_2^* = \max\left\{\frac{k_1 k_2^2 \phi_{1,2}}{(k_1 + k_2\phi_{1,2})^2}, B_2\right\} \tag{4.44}$$

$$b_1^* = \max\left\{\frac{k_1^2 k_2 \phi_{1,2}}{(k_1 + k_2\phi_{1,2})^2}, B_1\right\} \tag{4.45}$$

Therefore the equilibrium of the two VNF Servers is $\mathbf{b}^* = (b_1^*, b_2^*)$, where $b_1^*$ and $b_2^*$ are defined in (4.45) and (4.44), respectively. Accordingly, the unique SE equilibrium is $(\mathbf{b}^*, \mathbf{n}^*)$, where $\mathbf{n}^* = (n_1^*, n_2^*)$, and $n_1^*$ and $n_2^*$ are defined in (4.29).

### 4.2.4 Numerical Analysis

In this section we present a numerical analysis of the proposed distributed orchestration and resource allocation scheme. First we investigate its impact on the distribution of VNF Customers and on the VNF provisioning, showing results obtained through extensive simulations.

For illustrative purposes, we primarily focus on the two VNF Servers case (i.e., $M = 2$) as it allows us to highlight the dynamics of the interactions among VNF Customers and VNF Servers together with the impact of the various system parameters on the outcome of the game $\mathcal{G}^{(\mathcal{V}_S)}$; though, we also provide results also for the case $M > 2$, which makes possible to show the feasibility of the proposed learning procedure and to analyze the impact of latency when multiple VNF Servers are willing to provide the considered VNF.

In our simulations, we assume a population size of $N_{\mathcal{V}} = 3000$ VNF Customers. Unless otherwise stated, weight parameters are assumed as follows: $\alpha_1 = 1$, $\alpha_2 = 0.015$, $\alpha_3 = 0.035$, $\beta_1 = 1$ and $\beta_2 = 1$. Finally, and unless explicitly mentioned otherwise, we assume that the bandwidth-unit price $p_i^{(\mathcal{B})}$ and the cost $c_i$ are equal for both the

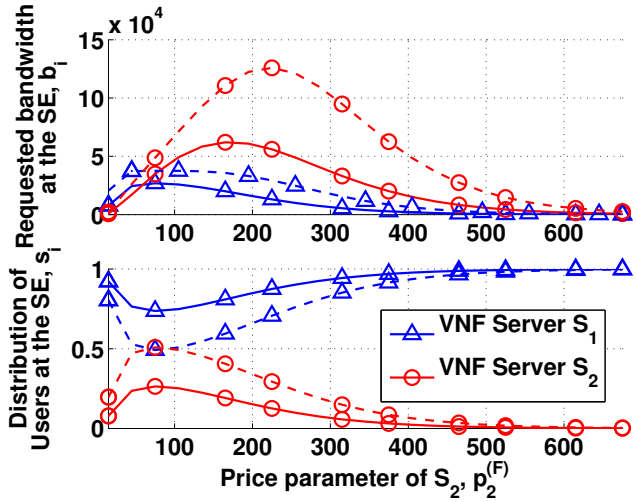



***Figure 4.8:*** *Requested bandwidth and population distribution at the equilibrium as a function of the price $p_2^{(\mathcal{F})}$ charged by $S_2$ (Solid lines: $d_1 = 5$ DUs and $d_2 = 40$ DUs; Dashed lines: $d_1 = d_2 = 40$ DUs).*

two VNF Servers and are set to $p_i^{(\mathcal{B})} = 1$ PUs and $c_i = 10$ PUs, respectively.

#### 4.2.4.1 Impact of pricing on the SE

In this section, we preliminarily study the impact of the pricing applied by the VNF Servers. To this purpose, in Fig. 4.8 we show the outcome of the game as a function of the price $p_2^{(\mathcal{F})}$ charged by VNF Server $S_2$ to its VNF Customers, when the price applied by VNF Server $S_1$ is assumed constant and equal to $p_1^{(\mathcal{F})} = 60$ PUs. Specifically, we show the amount of bandwidth $b_i$ that each VNF Server requests to the TO and the number $n_i$ of VNF Customers that connect to each VNF Server at the SE. Also, we consider two different configurations of the latencies $d_i$ experienced by VNF Customers connected to the $i$-th VNF Server. For the sake of generality, we will express latency in terms of DUs. More in detail, solid lines illustrate the outcome of the game when $d_1 = 5$ DUs and $d_2 = 40$ DUs, respectively. Instead, dashed lines refer

to the case when $d_1 = d_2 = 40$ DUs. As expected, when $p_2^{(\mathcal{F})}$ is high, VNF Customers are likely to connect to VNF Server $S_1$ because it applies a lower price (i.e., $p_1^{(\mathcal{F})} = 60$ PUs). Accordingly, customers get higher payoffs when they connect to VNF Server $S_1$ independently of the experienced connection latencies $d_i$. On the contrary, when $p_2^{(\mathcal{F})}$ is low, in order to attract more VNF Customers, the strategy of VNF Server $S_2$ consists in requesting a high amount of bandwidth to the TO. In this way, as evident from (4.23), the utility of the customers increases as a consequence of the increase in the shared bandwidth. Such behavior holds for values of $p_2^{(\mathcal{F})}$ that are below a given threshold, above which requesting more bandwidth is no more the optimal choice. For values of $p_2^{(\mathcal{F})}$ higher than this threshold[5], the optimal strategy of the VNF Servers consists in reducing the amount of requested bandwidth. Such behavior is motivated by the fact that an increase in the requested bandwidth causes an increase in the costs, which also leads to a reduction in the utilities achieved by the VNF Servers. Accordingly, when the cost to provide more resources to the VNF Customers is higher than the expected revenues, VNF Servers prefer to reduce the amount of shared resource to reduce costs and keep high revenues. Finally, it is worth noting that when $d_1 = 5$ DUs and $d_2 = 40$ DUs (solid lines), both VNF Servers request to the TO a lower amount of bandwidth than in the case when $d_1 = d_2 = 40$ DUs. This is due to the fact that, when latencies are equal (or similar), there is no monopolistic behavior and VNF Servers have to compete to attract more VNF Customers, which results in higher requested bandwidth.

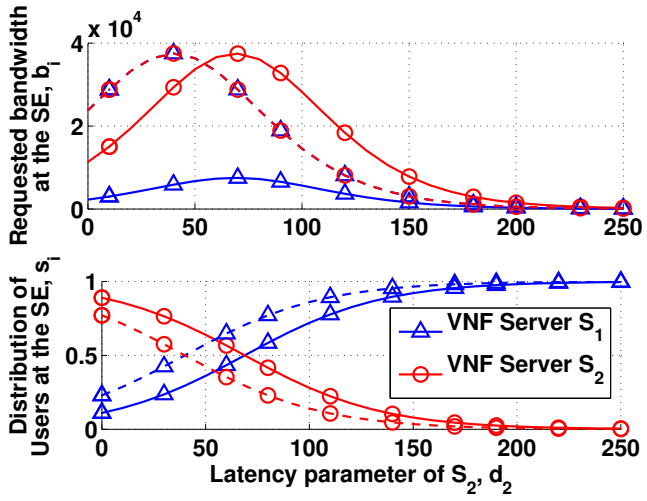



***Figure 4.9:*** *Requested bandwidth and distribution of VNF Customers at the equilibrium as a function of the latency $d_2$ of $S_2$ (Solid lines: $p_1^{(\mathcal{F})} = 20$ PUs and $p_1^{(\mathcal{F})} = 60$ PUs; Dashed lines: $p_1^{(\mathcal{F})} = p_2^{(\mathcal{F})} = 60$ PUs ).*

#### 4.2.4.2 Impact of latency on the SE

In this section, we investigate the impact of the latency experienced by VNF Customers when accessing VNF Servers on the outcome of the game $\mathcal{G}^{(\mathcal{V}_S)}$. In Fig. 4.9, we illustrate the requested bandwidth and the population distribution at the SE as a function of the latency $d_2$ from the VNF Server $S_2$. We consider two different scenarios: i) $p_1^{(\mathcal{F})} = 20$ PUs and $p_2^{(\mathcal{F})} = 60$ PUs (solid lines); and ii) $p_1^{(\mathcal{F})} = p_2^{(\mathcal{F})} = 60$ PUs (dashed lines). We assume that the latency from the VNF Server $S_1$ is fixed and equal to $d_1 = 40$ DUs. Instead, we let the latency $d_2$ from the VNF Server $S_2$ vary. Similarly to what we have observed in Fig. 4.8, in Fig. 4.9 we can identify two regions depending on the value of the parameter $d_2$: a first region for low values of $d_2$ where the bandwidth requested by both VNF Servers increases

[5]Note that in general the threshold values are different for the two VNF Servers.

as $d_2$ increases as well, and a second region for higher values of $d_2$, where the bandwidth requested by both VNF Servers decreases as the value of $d_2$ increases. From (4.23) we have that an increase in the latency $d_i$ causes a decrease in the utility achieved by each VNF Customer. Therefore, upon increasing the latency $d_2$, the number $n_2$ of VNF Customers connected to VNF Server $S_2$ decreases. On the contrary, the number $n_1$ of VNF Customers connected to VNF Server $S_1$ increases as the latency $d_2$ increases as well. It is worth noting that the bandwidth requested by both VNF Servers is the same in the case where $p_1^{(\mathcal{F})} = p_2^{(\mathcal{F})} = 60$ PUs. However, the fraction of VNF Customers that connect to each VNF Server is equal to 0.5 only when $d_1 = d_2 = 40$ DUs. So, when both VNF Servers charge VNF Customers with the same price, we have that $n_2 > n_1$ if $d_2 < d_1$. On the contrary, we have that $n_2 < n_1$ if $d_2 > d_1$, while $n_1 = n_2$ iff $d_2 = d_1$.

#### 4.2.4.3 Time-varying analysis

In this section, we discuss the impact of the population size $N_{\mathcal{V}}$ and the cost $c_i$ to process each flow on the outcome of the game $\mathcal{G}^{(\mathcal{V}_S)}$. To this purpose, we simulated a NFV scenario where the number of VNF Customers requesting a NFV and the cost $c_i$ vary in time according to realistic night/day usage patterns. More specifically, let the number of VNF Customers $N_{\mathcal{V}}$ and the cost $c_i$ vary in a 48-hours long temporal window as shown in Fig. 4.10.

Fig. 4.11 illustrates both the strategy of each VNF Server, i.e., the bandwidth requested to the TO, and the distribution of the population at the equilibrium when $p_1^{(\mathcal{F})} = p_2^{(\mathcal{F})} = 80$ PUs, $d_1 = 5$ DUs and $d_2 = 40$ DUs. Observe that when the cost to process flows is low, VNF Servers can support more customer connections. Accordingly, VNF Servers request more bandwidth to the network to attract a higher number of VNF Customers. However, when $d_1 = 5$ DUs and

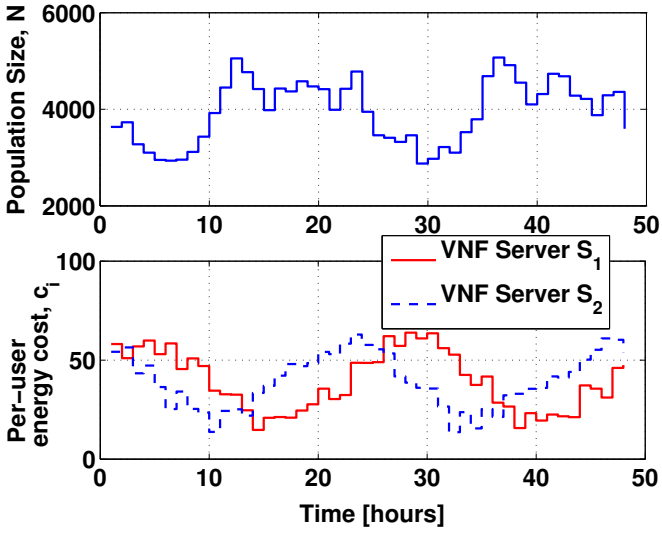



***Figure 4.10:*** *Population size $N_{\mathcal{V}}$ and price parameter $c_i$ as a function of time.*

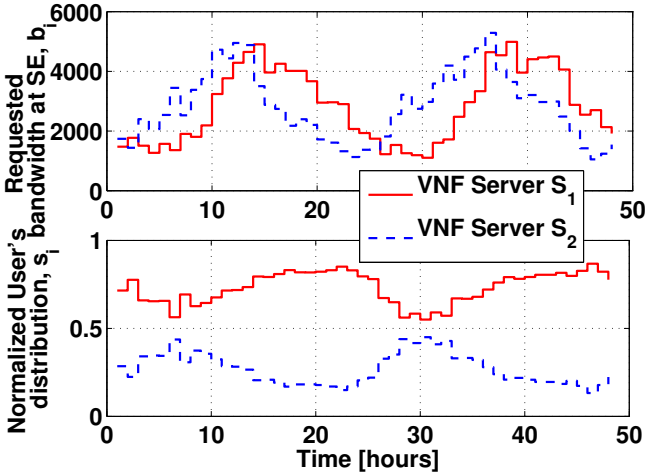



***Figure 4.11:*** *Requested bandwidth and distribution of VNF Customers at the equilibrium as a function of time when $d_1 = 5$ DUs and $d_2 = 40$ DUs.*

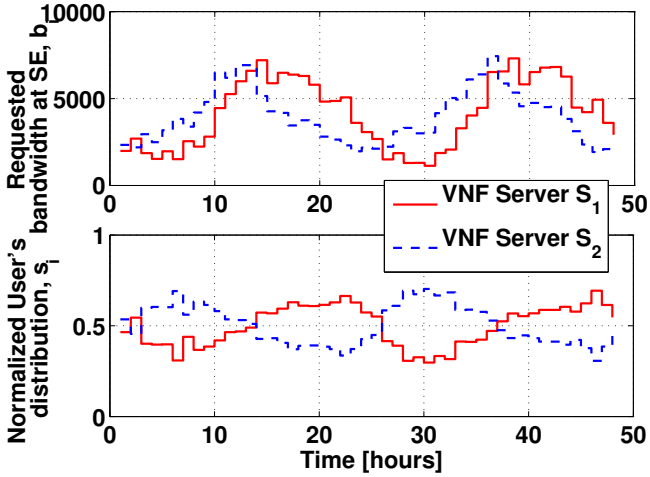



***Figure 4.12:*** *Requested bandwidth and distribution of VNF Customers at the equilibrium as a function of time when* $d_1 = d_2 = 40$ *DUs.*

$d_2 = 40$ DUs, even though the VNF Server $S_2$ provides customers with a higher amount of bandwidth, it also has a high latency. Therefore, to reduce the experienced latency, VNF Customers connect to the VNF Server $S_1$ and thus $n_1 > n_2$. Instead, when both VNF Servers have equal latency, i.e., $d_1 = d_2 = 40$ DUs, Fig. 4.12 shows that the majority of the population chooses the VNF Server which provides the highest amount of bandwidth.

#### 4.2.4.4 Impact of the PE position on the SE

In this section, we study the impact of the position of the PE node on the outcome of the game $\mathcal{G}^{(\mathcal{V}_S)}$. We consider five VNF Servers, i.e., $\mathcal{V}_S = \{S_1, S_2, S_3, S_4, S_5\}$, and five possible positions of the access PE, here denoted as $\text{PE}_k$ with $k = 1, 2, \ldots, 5$. We assume $p_i^{(\mathcal{F})} = 60$ PUs for all $i \in \mathcal{V}_S$. Each access PE position corresponds to a different latency configuration. The considered latency configurations are shown in Fig. 4.13 and, for the sake of readability, are also reported in Table 4.2.

***Table 4.2:*** *Latency Configurations in DUs*

| | $PE_1$ | $PE_2$ | $PE_3$ | $PE_4$ | $PE_5$ |
|---|---|---|---|---|---|
| $S_1$ | 31.6228 | 31.6228 | 42.4264 | 58.3095 | 76.1577 |
| $S_2$ | 36.0555 | 30 | 36.0555 | 50 | 67.082 |
| $S_3$ | 41.2311 | 22.3607 | 10 | 22.3607 | 41.2311 |
| $S_4$ | 80.6226 | 60.8276 | 41.2311 | 22.3607 | 10 |
| $S_5$ | 80.6226 | 60.8276 | 41.2311 | 22.3607 | 10 |

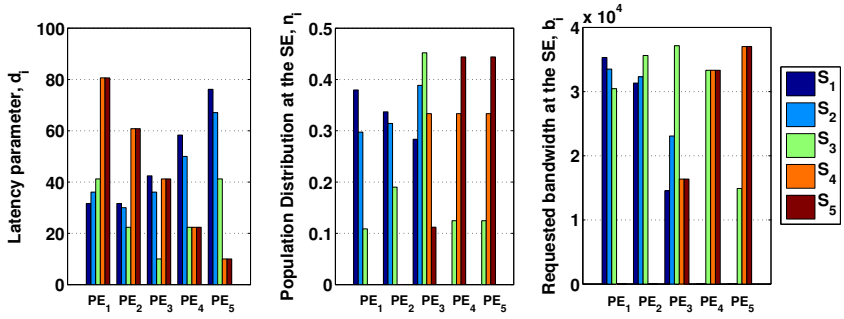



***Figure 4.13:*** *Latencies, requested bandwidth and distribution of VNF Customers at the SE for different access PE positions.*

In Fig. 4.13, it is shown that VNF Servers $S_1$ and $S_2$ provide low latencies when VNF Customers access the network through provider edges $PE_1$ and $PE_2$. On the contrary, the same VNF Servers provide high latencies when VNF Customers access through $PE_4$ and $PE_5$. The converse holds for VNF Servers $S_4$ and $S_5$, while $S_3$ provides low latencies to VNF Customers independently of the position of the access PE. Accordingly, when VNF Customers access through $PE_1$ and $PE_2$ the majority of them decides to connect to $S_1$ and $S_2$. Thus, to serve such a high amount of VNF Customers, $S_1$ and $S_2$ request a high amount of bandwidth to the TO. As expected, the contrary holds in the case when VNF Customers access through $PE_4$ and $PE_5$. That is, few VNF Customers connects to $S_1$ and $S_2$ while the majority of them decides to connect to VNF Servers $S_4$ and $S_5$. Since $S_3$ provides

low latency in almost all cases, it is worth noting that it is able to attract VNF Customers and almost always requests high amount of bandwidth to the TO.

#### 4.2.4.5 Convergence Analysis

In this section, we investigate the convergence of the proposed learning procedure in (3.15). Specifically, we are interested in analyzing the convergence speed of (3.15) and its scalability w.r.t. the number $M$ of VNF Servers. Results shown in this section are averaged over 100 simulation runs where we have assumed $p_i^{(\mathcal{F})} = 60$ PUs for all $i \in \mathcal{V}_{\mathcal{S}}$, while the latency parameters $d_i$ have been randomly generated.

To measure the convergence speed of the proposed learning procedure, at each iteration we consider the normalized Euclidean distance between the bandwidth vector $\mathbf{b}(m)$ computed in (3.15) and the SE $\mathbf{b}^*$ as follows:

$$d(\mathbf{b}(m), \mathbf{b}^*) = \sqrt{\sum_{i \in \mathcal{V}_{\mathcal{S}}} \left( \frac{|b_i(m) - b_i^*|}{B_i} \right)^2} \tag{4.46}$$

In Fig. 4.14, we show how fast the proposed learning procedure converges to the unique SE of the game $\mathcal{G}^{(\mathcal{V}_S)}$ when $M = 10$, for different step-size rules. Specifically, we consider both variable step-size (i.e., $\gamma_m = 1/m^\beta$) with $\beta \in \{0.51, 1\}$ and fixed step-size rules. It is shown that fixed step-size rules converge faster than variable step-size rules. Also, the convergence speed is faster when high values of the fixed step-size are considered, i.e., $\gamma_m = 2$. Recall that convergence of the learning procedure under variable step-size rules is ensured by Theorem 8. Unfortunately, the same is not true for fixed step-size rules, as in this case convergence to the SE cannot be proven analytically. Therefore, to guarantee convergence to the SE while achieving a fast convergence speed, a variable step-size $\gamma_m = 1/m^\beta$ with $\beta = 0.51$

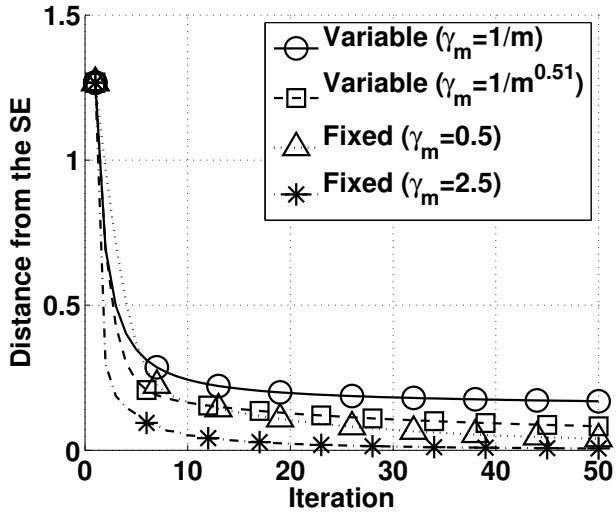



***Figure 4.14:*** *Distance from the SE for different step-size rules.*

should be considered.

CHAPTER

# FIVE

# IMPLEMENTATION ASPECTS

In this section, we discuss implementation aspects of the resource allocation schemes that we have presented in Chapters 3 and 4. Specifically, we provide algorithms and flowcharts which illustrate main procedures and message exchanges between main actors of the considered problem. Also, we analyze the proposed algorithmic implementations and we study their computational complexity and scalability.

## 5.1 Power-efficient Jamming-proof RAN

In the following, we propose and analyze algorithmic implementations for the power-efficient and jamming-proof RAN's resource allocation schemes that we have proposed in Section 3.

### 5.1.1 Joint Power-efficient and Jamming-proof Approach

In this section, we consider the centralized resource allocation scheme to provide power-efficient and jamming-proof RAN as described in

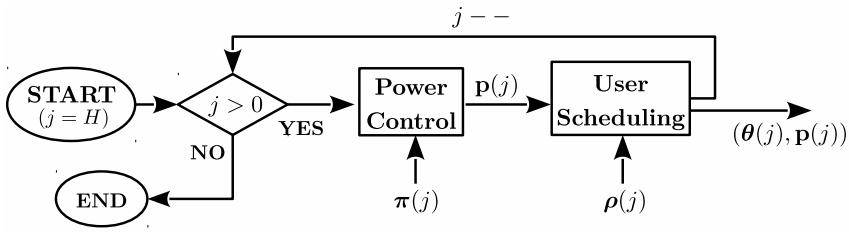



***Figure 5.1:*** *Structure of the dynamic programming problem.*

Section 3.1 and we propose an algorithmic implementation for the latter problem.

The building blocks of our proposed solution are shown in Fig. 5.1. In Section 3.1.3.1, we have exploited Bellman's equation which we solve by exploiting backward induction. Backward induction implies that we start from slot $j = H$ and go backwards in time.

Therefore, at each stage we first solve the power control problem by exploiting the system state parameter $\boldsymbol{\pi}(j)$ in (3.14). The solution of the power control problem consists in the power control vector $\mathbf{p}(j)$ whose elements $p_{nk}^{OPT}(j) \in \mathbf{p}(j)$ are calculated in (3.22). To solve the scheduling problem, the system requires $\mathbf{p}(j)$ and the residual performance vector $\boldsymbol{\rho}(j)$ in (3.22). The scheduler solves the single-stage reward maximization problem and provides the optimal scheduling policy $\boldsymbol{\theta}(j)$. Finally, each element in $\mathbf{p}(j)$ is modified such that each $p_{nk}(j) \in \mathbf{p}(j)$ is set to $p_{nk}(j) = 0$ if $\theta_{nk}(j) = 0$ and the optimal joint power control and user scheduling policy $(\boldsymbol{\theta}(j), \mathbf{p}(j))$ is found.

At each stage $j$, the number of possible combinations of $\boldsymbol{\rho}(j)$ and $\boldsymbol{\pi}(j)$ are ${N_r}^N$ and $(\frac{N_p(N_p-1)}{2})^{NK}$, respectively. Finding the maximum of the single-stage reward maximization problem has complexity $\mathcal{O}(\omega)$, where $\omega = (\max\{K, N\})^{min\{K,N\}}$. Finally, the number of possible combinations in (3.17) is $\mathcal{O}(KNH)$. Therefore, the overall complexity of the Bellman's equation is $\mathcal{O}\left(\omega KN\left(N_r \frac{N_p^K (N_p-1)^K}{2^K}\right)^N H\right)$. However, if $K < N$, by exploiting

constraints (3.19) and (3.20) we can reduce the complexity of the single-stage maximization problem to $\mathcal{O}\left(\frac{N!}{(N-K)!}\right)$. Thus, the complexity becomes $\mathcal{O}\left(KN\frac{N!}{(N-K)!}\left(N_r\frac{N_p^K(N_p-1)^K}{2^K}\right)^N H\right)$. In the special case where $N = \mathcal{O}(\log H)$, the proposed algorithm has polynomial complexity. That is, to solve the problem in polynomial time, we should consider either a low number of users that scales as the logarithm of the horizon $H$, or a large horizon. In all other cases, the complexity of the algorithm exponentially increases with the number of users in the system.

### 5.1.2 Power-efficient Approach

In Section 3.2, we have discussed the problem of power-efficient resource allocation for the RAN when no jamming attacks are performed. Accordingly, in Algorithm 1 we propose an algorithmic implementation of the learning scheme (3.43) we have described in Section 3.2.

**Algorithm 1** Learning Scheme for Power-efficient RAN

Parameter: step size $\gamma_m$ (default: $\gamma_m = 1/m$).
Initialize: $m \leftarrow 0$; scores $z_{nk} \leftarrow 0$ for all $n \in \mathcal{N}$, $k \in \mathcal{K}$.
**Repeat**
- $m \leftarrow m + 1$;
- **for each** user $n \in \mathcal{N}$ **do simultaneously**
  - set transmit power $p_{nk} \leftarrow P_n \dfrac{e^{z_{nk}}}{1 + \sum_\nu e^{z_{n\nu}}}$;
  - measure $\mathsf{SINR}_{nk}$;
  - update marginal utilities: $v_{nk} \leftarrow \dfrac{1}{p_{nk}} \dfrac{\mathsf{SINR}_{nk}}{1 + \mathsf{SINR}_{nk}}$ from (3.38);
  - update marginal power cost: $v_{n,0} \leftarrow \Gamma_n'(P_n - p_n)$ from (3.40);
  - update scores: $z_{nk} \leftarrow z_{nk} + \gamma_m \big[v_{nk} - v_{n,0}\big]$;
  - **until** termination criterion is reached.

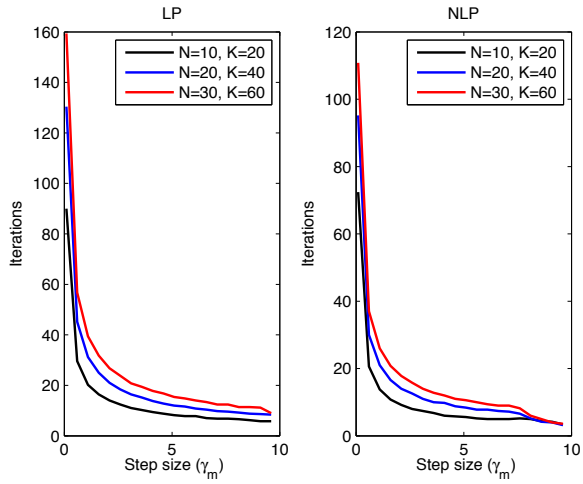



***Figure 5.2:*** *Evolution of the equilibration rate for different pricing models and network configurations as a function of the step-size $\gamma_m$.*

where $\gamma_m$ is the step-size parameter; $P_n$ is the maximum transmission power for user $n \in \mathcal{N}$ and $\text{SINR}_{nk}$ is defined in (3.27).

It can be easily shown that the computational complexity of Algorithm 1 at each iteration is $\mathcal{O}(NK)$. That is, the complexity of the proposed Algorithm 1 for power-efficient resource allocation in RAN is linear with both the number of users ($N$) and frequencies ($K$).

To investigate the algorithm's scalability, we plotted in Fig. 5.2 the number of iterations needed to reach the equilibrium for different network configurations as a function of the step-size parameter $\gamma_m$. Specifically, we considered $N/K = 0.5$ and varied $K$ to study the impact of the number of users on the algorithm's convergence rate: Fig. 5.2 shows that the proposed distributed algorithm scales well with the number of users, especially if higher values of the step-size $\gamma_m$ are used in the learning process. More precisely, even for a large number of users, the algorithm converges to the game's equilibrium within a few iterations, and this convergence rate is approximately independent of the exact number of users.

### 5.1.3 Jamming-proof Approach

In Section 3.3, we have proposed a resource allocation scheme that exploits timing channels to provide jamming-proof access to the RAN. Also, we have shown that an optimal resource allocation solution exists in both the Nash and Stackelberg games. We have derived a closed-form solution for the NE and a closed-form approximation for the SE in (3.56) and (3.66), respectively. Accordingly, it can be shown that the computational complexity of the timing channel-based jamming-proof resource allocation scheme in Section 3.3 has complexity $\mathcal{O}(K)$.

## 5.2 Network Management

In this section, we propose an algorithmic implementation for the SDN-based multi-tenant backhaul management we have proposed in Section 4.1. Specifically, we first propose an algorithmic implementation of the learning procedure in (4.12) and investigate its computational complexity. Then, we focus on the bandwidth auction and we show how OpenFlow's procedures can be exploited to implement (4.12) and to efficiently allocate resource among several tenants through network slicing.

To implement the learning procedure in (4.12) we propose the following Algorithm 2.
where $\gamma_m$ is the step-size parameter and $B_i$ is the maximum admissible bid for tenant $i \in \mathcal{N}_{\mathcal{C}}$.

The computational complexity of Algorithm 2 is $\mathcal{O}(N_C)$. That is, the proposed learning scheme has linear complexity w.r.t. the number of tenants.

OpenFlow specifications provide the `fvctl add-slice [options] <slicename> <controller-url> <admin-email>` command that can be easily exploited to create/modify slices. Specif-

**Algorithm 2** Learning Scheme for SDN-based multi-tenant Network Management

Parameter: step-size sequence $\gamma_m$ (default: $\gamma_m = 1/m$).
Initialize: $m \leftarrow 0$; $z_i \leftarrow 0$ for all $i \in \mathcal{N}_\mathcal{C}$.
**Repeat**
  $m \leftarrow m + 1$;
  **for each** bidder $i \in \mathcal{N}_\mathcal{C}$ **do simultaneously**
    bid $b_i \leftarrow B_i \left[1 + \exp(-z_i)\right]^{-1}$;
    measure marginal utility $v_i$ from (4.11);
    update scores: $z_i \leftarrow z_i + \gamma_m v_i$;
    **until** termination criterion is reached.

ically, it is used at the high-level by the FlowVisor during the start-up phase to create a new slice whose name is `<slicename>`, and assign it to a given Controller at a specific URL (i.e., `<controller-url>` could be `tcp:hostname:port`) [111]. It is also possible to specify some additional options and the email address of the tenant.

In order to limit the maximum bandwidth that each tenant can use in its slice, we exploit the OpenFlow `Rate Limiter` [175]. `Rate Limiter` is a software element in the switch that continuously monitors and measures the rate of packets on a given link. When the rate of the traffic generated by a given Controller exceeds the maximum threshold, `Rate Limiter` executes a `drop` action on those packets that are not allowed to flow through the network.

In the following, we show the basic operations performed by both the FlowVisor and the Controllers before, during and after the auction. The communication link between each Controller and the FlowVisor can exploit either TCP or UDP protocols. For the sake of simplicity, let us focus on the simple case when the FlowVisor monitors only a single link connecting two different OpenFlow-compliant switches. Let us denote as $l$ the monitored link. In our model we assume that $R = \xi \hat{R}$, where $\xi \in [0, 1]$ is the fraction of the available bandwidth

that the FlowVisor is willing to sell[1]. As discussed in Section 4.1.3, the FlowVisor also fixes a cost $\lambda$ that each Controller has to pay.

In Figure 5.3 we provide a flowchart of the relevant operations executed by the both the FlowVisor and the $i$-th Controller during the auction.

A `notifyAuction()` message is sent to each Controller by indicating the link $l$, the amount of bandwidth $R$ to be sold and the cost $\lambda$. Each Controller evaluates its interest $\theta_i$ in buying some additional bandwidth on the link $l$, determines its bid according to Algorithm 2, and sends a `sendBid()` message including the bid $b_i$ to the FlowVisor.

By simply using a timeout, the FlowVisor is able to collect the bidding vector $b$ containing all the bids submitted by the interested Controllers and modify the bandwidth allocation policy by modifying the `Rate Limiter`'s settings on each switch connected to the corresponding link. Specifically, the FlowVisor assigns an additional bandwidth $\varrho_i$ to each Controller according to (4.1).

Hence, each Controller re-executes Algorithm 2, the bidding profile is updated and sent to the FlowVisor until the convergence criterion is reached and a `notifyAuctionEnd()` message is sent by the FlowVisor to the Controllers.

When the auction is over, each Controller that has successfully submitted a bid, sends the payment notification that depends on the cost parameter $\lambda$.

Finally, in Figure 5.4 we investigate the scalability of the proposed learning procedure as the number of bidders participating in the auction increases, and we also show how fast the learning process converges to the unique NE as a function of the fixed step-size $\gamma_m$ for different cost schemes. Figure 5.4 shows that an increase in the step-

[1]For example, the FlowVisor could be interested in keeping unused some bandwidth to provide additional best-effort services or avoid congestion caused by a sudden increase in the traffic rate on the considered link.

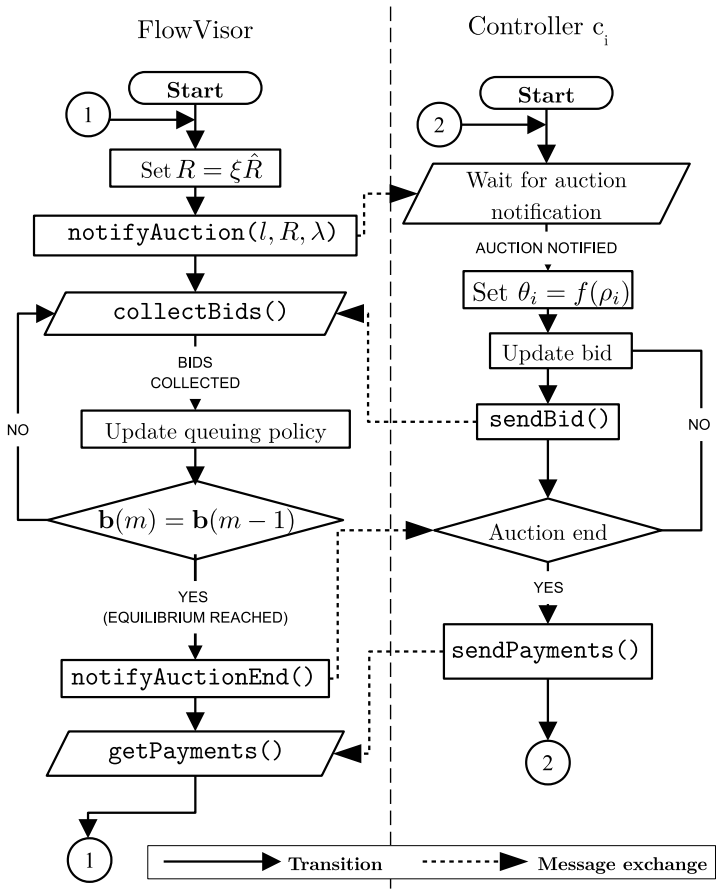



***Figure 5.3:*** *Flowchart of the operations performed by the FlowVisor and the generic Controller $C_i$.*

size causes an improvement in the convergence rate of the learning process in both the LC and NLC scheme as defined in Section 4.1.1.2. For low values of the step-size $\gamma_m$, if a small number of bidders is considered ($N_C = 10$), the Algorithm 2 converges fast to the equilibrium point, while the convergence rate is slower when the number of bidders increases. By increasing the value of $\gamma_m$, the convergence rate of the learning process is fast even in highly populated auctions with a large number of participants. Therefore, an increase in the step-size helps

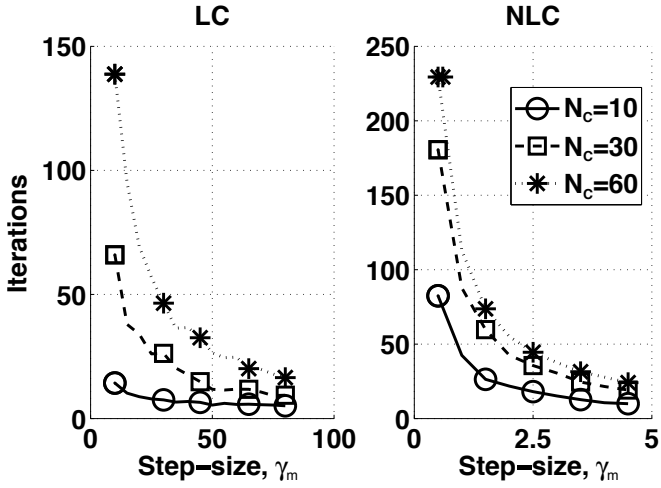



***Figure 5.4:*** *Number of iterations needed to reach the NE as a function of the step-size $\gamma_m$ for different network configurations and cost functions.*

in improving the scalability of the proposed approach; in fact, with large values of $\gamma_m$ the number of iterations needed to reach the equilibrium in scarcely populated auctions ($N_C = 10$) and highly populated auctions ($N_C = 60$) are comparable and anyway small.

## 5.3 Service Management

In this section, we propose an algorithmic implementation of the learning scheme in Section 4.2 for NFV-based service management in the CN. Specifically, the algorithmic implementation of (4.37) is presented in Algorithm 3.

where $\gamma_m$ is the step-size parameter and $B_i$ is the maximum admissible requested bandwidth for NFV Server $i \in \mathcal{V}_S$. Also, the computational complexity of Algorithm 3 is $\mathcal{O}(|\mathcal{V}_S|)$, i.e., it is linear w.r.t. the number $|\mathcal{V}_S|$ of NFV Servers.

For the sake of illustration, in Fig. 5.5 we show exchanged mes-

**Algorithm 3** Learning Scheme for NVF-based Service Management

Parameter: step-size sequence $\gamma_m$ (default: $\gamma_m = 1/m$).

Initialize: $m \leftarrow 0$; $z_i \leftarrow 0$ for all $i \in \mathcal{V}_S$.

**Repeat**

$m \leftarrow m + 1$;

**for each** VNF Server $i \in \mathcal{V}_S$ **do simultaneously**

requested bandwidth $b_i \leftarrow B_i \left[1 + \exp(-z_i)\right]^{-1}$;

measure marginal utility $v_i$ from (4.33);

update scores: $z_i \leftarrow z_i + \gamma_m v_i$;

**until** termination criterion is reached.

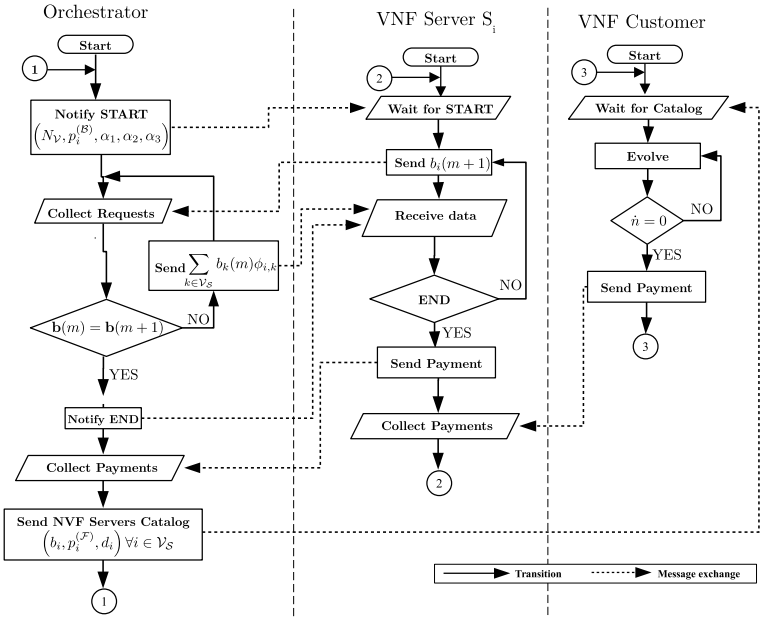



***Figure 5.5:*** *Flowchart of main operations executed by the Orchestrator, VNF Servers and VNF Customers under the proposed resource allocation and orchestration scheme.*

sages and main procedures executed by the Orchestrator, VNF Servers and VNF Customers. Specifically, the Orchestrator notify the start of the resource allocation scheme to all VNF Servers in $\mathcal{V}_S$. Each

VNF Server $i \in \mathcal{V}_{\mathcal{S}}$ executes Algorithm 3 until convergence criterion is reached. When the equilibrium is reached, the Orchestrator collects payments from VNF Servers. Thus, a VNF Servers Catalog that, for each $i \in \mathcal{V}_{\mathcal{S}}$, contains the offered bandwidth $b_i$, the price $p_i^{(\mathcal{F})}$ to access the considered function and the latency parameter $d_i$ is sent to all VNF Customers in $\mathcal{V}_{\mathcal{U}}$. VNF Customers evolve according to the replicator dynamics in (4.24) from VNF Servers. Finally, when the evolutionary equilibrium is reached, VNF Customers send payments to the Orchestrator and the resource allocation scheme is stopped until the allocation period expires and it is started again.

In Fig. 5.6, we show how many iterations the proposed learning procedure needs to reach the equilibrium as a function of the step-size $\gamma_m$ for different values of the number $M$ of VNF Servers. More in detail, we let the learning procedure run until the stopping condition is reached, i.e., $d(b_i(m), b_i^*) \leq 0.01$ for all $i \in \mathcal{V}_{\mathcal{S}}$. As expected, an increase in the value of the step-size improves the convergence speed of the learning procedure. Furthermore, in Fig. 5.6 we show the scalability of the proposed learning procedure w.r.t. the number $M$ of VNF Servers. It is important to note that an increase in the value of the step-size $\gamma_m$ allows to improve the convergence speed of the learning procedure even when high number of VNF Servers are considered, e.g., $M = 50$. Thus, by properly increasing the value of the step-size, it is also possible to improve the scalability of the learning procedure.

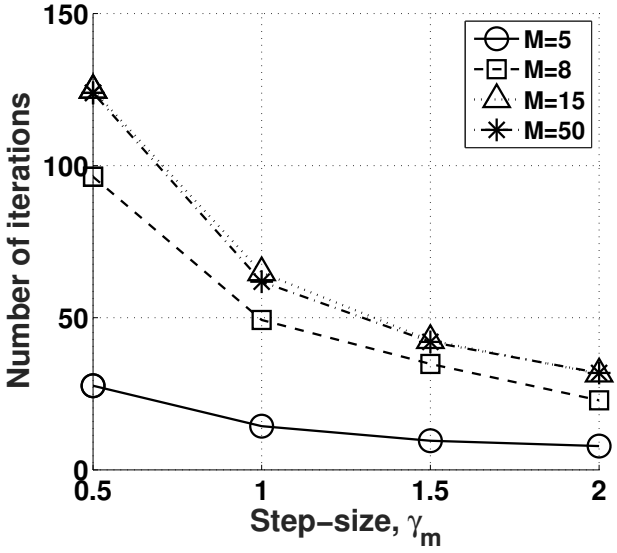



***Figure 5.6:*** *Number of iterations needed to reach the equilibrium for different number $M$ of VNF Servers.*

CHAPTER

# SIX

# CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we have studied the problem of user-centric resource allocation for power-efficient and jamming-proof RAN on top of a multi-tenant backhaul. We have surveyed previous work on the considered topic and we have identified crucial issues and challenges to design a holistic approach for efficient resource management in the considered scenario.

We have tackled energy consumption and security issues by providing centralized and distributed algorithms for power-efficient and jamming-proof access to the RAN. By exploiting SDNs and network slicing, an auction-based resource allocation scheme for flexible and dynamic network management of the multi-tenant backhaul has been proposed. Also, a NFV-based approach for service and network function management in the CN has been investigated by exploiting hierarchical and evolutionary game theory.

Our work focused on the analysis of the existence and uniqueness of optimal resource allocation policies and particular interest has been given to the computational complexity and the scalability of the proposed solutions. As a result, we have shown that the proposed

resource allocation schemes have very low computational complexity and well-scales with the number of users and tenants in the network.

Even though we have devised a holistic approach for resource management in modern communication networks, there are still open issues and challenges that must be addressed. A promising approach that is expected to revolutionize next-generation communication paradigms consists in the exploitation of heterogeneous networks together with the softwarization of network control and management in the cloud. Future works will be focused on the extension of our work to the case where heterogeneous RANs composed by GSM/UMTS, wi-fi and WiMax radios are shared among several network operators and resource are dynamically allocated according to time-varying user and traffic patterns. Our approach will be focused on the implementation aspects of a cloud-based resource management and control by exploiting the resource allocation solutions which have been proposed in this thesis.

Another future direction consists in the exploitation of human social behavior to design optimal resource allocation schemes. It is well-known that users often move in groups. Also, it is most likely that users in the same group have similar interests and behavior. Furthermore, users in the same group are expected to be in proximity and D2D communications can be exploited to offload the RAN. Accordingly, it would be useful to provide dynamic network resource management and control for efficient information delivery in social groups in heterogeneous networks.

# BIBLIOGRAPHY


[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] T. Velte, A. Velte, and R. Elsenpeter, *Cloud computing, a practical approach*. McGraw-Hill, Inc., 2009.

[3] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: principles and paradigms*. John Wiley & Sons, 2010, vol. 87.

[4] M. Rumney *et al.*, *LTE and the evolution to 4G wireless: Design and measurement challenges*. John Wiley & Sons, 2013.

[5] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 86–92, 2014.

[6] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 74–80, 2014.

[7] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "On the trade-off between delivery delay and power consumption in opportunis-

tic scenarios," in *Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on*, July 2013, pp. 34–38.

[8] I. F. Akyildiz and J. M. Jornet, "The internet of nano-things," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 58–63, 2010.

[9] ——, "Electromagnetic wireless nanosensor networks," *Nano Communication Networks*, vol. 1, no. 1, pp. 3–19, 2010.

[10] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "A timing channel-based MAC protocol for energy-efficient nanonetworks," *Nano Communication Networks*, vol. 6, no. 2, pp. 39 – 50, 2015, pervasive and Ubiquitous Environment Interactions with Nano Things.

[11] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "Exploiting object group localization in the internet of things: Performance analysis," *Vehicular Technology, IEEE Transactions on*, vol. 64, no. 8, pp. 3645–3656, Aug 2015.

[12] S. D'Oro, P. Mertikopoulos, A. Moustakas, and S. Palazzo, "Interference-based pricing for opportunistic multi-carrier cognitive radio systems," *Wireless Communications, IEEE Transactions on*, vol. 14, no. 12, pp. 6536–6549, 2015.

[13] ——, "Adaptive transmit policies for cost-efficient power allocation in multi-carrier systems," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, May 2014, pp. 1–7.

[14] *FCC online table of frequency allocations*, available at https://transition.fcc.gov/oet/spectrum/table/fcctable.pdf.

[15] J. Y. Hui, "Resource allocation for broadband networks," *Selected Areas in Communications, IEEE Journal on*, vol. 6, no. 9, pp. 1598–1608, 1988.

[16] J. Huang, V. G. Subramanian, R. Agrawal, R. Berry *et al.*, "Downlink scheduling and resource allocation for ofdm systems," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 1, pp. 288–296, 2009.

[17] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Now Publishers Inc, 2006.

[18] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1452–1463, 2006.

[19] S. D'Oro, S. Grancagnolo, E. Marano, and C. Rametta, "Enhancing tetra with capabilities exploiting mipv6 and ieee 802.21 mih services," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, July 2013, pp. 976–981.

[20] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "SatCache: a profile-aware caching strategy for information-centric satellite networks," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 4, pp. 436–444, 2014. [Online]. Available: http://dx.doi.org/10.1002/ett.2723

[21] M. A. Marotta, N. Kaminski, I. Gomez-Miguelez, L. Z. Granville, J. Rochol, L. DaSilva, and C. B. Both, "Resource sharing in heterogeneous cloud radio access networks," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 74–82, 2015.

[22] J. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 136–144, March 2013.

[23] A. Rath, S. Hua, and S. Panwar, "Femtohaul: Using femtocells with relays to increase macrocell backhaul bandwidth," in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, March 2010, pp. 1–5.

[24] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "Netlord: a scalable multi-tenant network architecture for virtualized datacenters," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 62–73.

[25] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, N. Gude, P. Ingram *et al.*, "Network virtualization in multi-tenant datacenters," in *USENIX NSDI*, 2014.

[26] M. Hasan and E. Hossain, "Distributed resource allocation for relay-aided device-to-device communication: A message passing approach," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 11, pp. 6326–6341, Nov 2014.

[27] K. Yang, N. Prasad, and X. Wang, "A message-passing approach to distributed resource allocation in uplink dft-spread-ofdma systems," *Communications, IEEE Transactions on*, vol. 59, no. 4, pp. 1099–1113, April 2011.

[28] I. Sohn, S. H. Lee, and J. G. Andrews, "Belief propagation for distributed downlink beamforming in cooperative mimo cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 12, pp. 4140–4149, 2011.

[29] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 809–819, 2005.

[30] *Auction of Advanced Wireless Services (AWS-3) Licenses Closes; Winning Bidders Announced for Auction 97*, available at https://apps.fcc.gov/edocs_public/attachmatch/DA-15-131A2.pdf.

[31] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri, "A general framework for wireless spectrum auctions," in *New Frontiers in Dy-*

*namic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on.* IEEE, 2007, pp. 22–33.

[32] P. Cramton and J. A. Schwartz, "Collusive bidding: Lessons from the fcc spectrum auctions," *Journal of regulatory Economics*, vol. 17, no. 3, pp. 229–252, 2000.

[33] S. D'Oro, P. Mertikopoulos, A. L. Moustakas, and S. Palazzo, "Cost-efficient power allocation in OFDMA cognitive radio networks," in *EUCNC '15: Proceedings of the 2015 European Conference on Networks and Communications*.

[34] S. D'Oro, E. Ekici, and P. Palazzo, "Optimal power allocation and scheduling under jamming attacks," *submitted for publication*.

[35] S. D'Oro, L. Galluccio, G. Morabito, S. Palazzo, L. Chen, and F. Martignon, "Defeating jamming with the power of silence: A game-theoretic analysis," *Wireless Communications, IEEE Transactions on*, vol. 14, no. 5, pp. 2337–2352, May 2015.

[36] S. D'Oro, L. Galluccio, P. Mertikopoulos, G. Morabito, and S. Palazzo, "Auction-based resource allocation in openflow multi-tenant networks," *submitted for publication*.

[37] S. D'Oro, L. Galluccio, G. Morabito, S. Palazzo, and G. Schembra, "A distributed framework for resource allocation and orchestration of nfv networks: a game theoretic approach," *submitted for publication*.

[38] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 641–646, November 1993.

[39] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 59–67, 2008.

[40] G. Scutari, S. Barbarossa, and D. Palomar, "Potential games: A framework for vector power control problems with coupled constraints," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 4, May 2006, pp. IV–IV.

[41] T. Alpcan, T. Başar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," *Wireless Networks*, vol. 8, pp. 659–670, 2002.

[42] C.-W. Sung and W. S. Wong, "A noncooperative power control game for multirate CDMA data networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 1, pp. 186–194, January 2003.

[43] F. Meshkati, M. Chiang, H. V. Poor, and S. C. Schwartz, "A game-theoretic approach to energy-efficient power control in multicarrier CDMA systems," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1115–1129, 2006.

[44] S. T. Chung, S. J. Kim, J. Lee, and J. M. Cioffi, "A game-theoretic approach to power allocation in frequency-selective Gaussian interference channels," in *ISIT '03: Proceedings of the 2003 International Symposium on Information Theory*, 2003.

[45] G. A. Gupta and S. Toumpis, "Power allocation over parallel Gaussian multiple access and broadcast channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3274–3283, July 2006.

[46] F. Meshkati, H. V. Poor, and S. C. Schwartz, "Energy-efficient resource allocation in wireless networks: an overview of game-theoretic approaches," *IEEE Signal Process. Mag.*, vol. Special Issue on Resource-Constrained Signal Processing, Communications and Networking, May 2007.

[47] C. U. Saraydar and D. G. Narayan B. Mandayam, "Efficient power

control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, February 2002.

[48] P. Mertikopoulos, E. V. Belmega, A. L. Moustakas, and S. Lasaulce, "Distributed learning policies for power allocation in multiple access channels," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 96–106, January 2012.

[49] G. Bacci, E. V. Belmega, and L. Sanguinetti, "Distributed energy-efficient power optimization in cellular relay networks with minimum rate constraints," in *ICASSP '14: Proceedings of the 2014 International Conference on Acoustics, Speech and Signal Processing*, to appear.

[50] FCC Spectrum Policy Task Force, "Report of the spectrum efficiency working group," Federal Communications Comission, Tech. Rep., November 2002.

[51] K. V. Schinasi, "Spectrum management: Better knowledge needed to take advantage of technologies that may improve spectrum efficiency," United States General Accounting Office, Tech. Rep., May 2004.

[52] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Commun. Mag.*, vol. 6, no. 4, pp. 13–18, August 1999.

[53] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[54] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, February 2005.

[55] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proc. IEEE*, vol. 97, no. 5, pp. 894–914, 2009.

[56] C. U. Saraydar, N. B. Mandayam, and D. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, February 2002.

[57] D. Niyato and E. Hossain, "Spectrum trading in cognitive radio networks: A market-equilibrium-based approach," *IEEE Wireless Commun. Mag.*, vol. 15, no. 6, pp. 71–80, December 2008.

[58] B. Wang, Y. Wu, and K. Liu, "Game theory for cognitive radio networks: An overview," *Computer networks*, vol. 54, no. 14, pp. 2537–2561, 2010.

[59] J.-S. Pang, G. Scutari, D. P. Palomar, and F. Facchinei, "Design of cognitive radio systems under temperature-interference constraints: A variational inequality approach," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3251–3271, June 2010.

[60] Y. Yang, G. Scutari, P. Song, and D. P. Palomar, "Robust MIMO cognitive radio under interference temperature constraints," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2465–2483, November 2013.

[61] Y. Xu and X. Zhao, "Robust power control for multiuser underlay cognitive radio networks under QoS constraints and interference temperature constraints," *Wireless Personal Communications*, vol. 75, no. 4, pp. 2383–2397, 2014.

[62] W. Wang, Y. Cui, T. Peng, and W. Wang, "Noncooperative power control game with exponential pricing for cognitive radio network," in *VTC '07: Proceedings of the 2007 IEEE Vehicular Technology Conference*, April 2007, pp. 3125–3129.

[63] P. Mertikopoulos and E. V. Belmega, "Transmit without regrets: online optimization in MIMO–OFDM cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 1987–1999, November 2014.

[64] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, IEEE*, vol. 20, no. 3, pp. 41–47, 2006.

[65] R. Saranyadevi, M. Shobana, and D. Prabakar, "A survey on preventing jamming attacks in wireless communication," *International Journal of Computer Applications*, vol. 57, no. 23, pp. 1–3, November 2012, published by Foundation of Computer Science, New York, USA.

[66] R.-T. Chinta, T. F. Wong, and J. M. Shea, "Energy-efficient jamming attack in IEEE 802.11 MAC," in *Military Communications Conference, 2009. MILCOM 2009. IEEE.* IEEE, 2009, pp. 1–7.

[67] Y. W. Law, L. Van Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks.* ACM, 2005, pp. 76–88.

[68] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2005, pp. 46–57.

[69] M. Strasser, B. Danev, and S. Čapkun, "Detection of reactive jamming in sensor networks," *ACM Trans. Sen. Netw.*, vol. 7, no. 2, pp. 16:1–16:29, Sep. 2010.

[70] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: reactive jamming in wireless networks: how realistic is the threat?" in *Proceedings of the fourth ACM conference on Wireless network security.* ACM, 2011, pp. 47–52.

[71] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "Efficiency analysis of jamming-based countermeasures against malicious tim-

ing channel in tactical communications," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4020–4024.

[72] M. Pajic and R. Mangharam, "Spatio-temporal techniques for anti-jamming in embedded wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 51, 2010.

[73] A. Wood, J. Stankovic, and G. Zhou, "Deejam: Defeating energy-efficient jamming in ieee 802.15.4-based wireless networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, June 2007, pp. 60–69.

[74] D. Nguyen, C. Sahin, B. Shishkin, N. Kandasamy, and K. R. Dandekar, "A real-time and protocol-aware reactive jamming framework built on software-defined radios," in *Proceedings of the 2014 ACM Workshop on Software Radio Implementation Forum*, ser. SRIF '14. ACM, 2014, pp. 15–22.

[75] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: reactive jamming in wireless networks: how realistic is the threat?" in *Proceedings of the fourth ACM conference on Wireless network security*. ACM, 2011, pp. 47–52.

[76] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa, "On the performance of ieee 802.11 under jamming," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, pp. –.

[77] A. Marttinen, A. Wyglinski, and R. Jantti, "Statistics-based jamming detection algorithm for jamming attacks against tactical manets," in *Military Communications Conference (MILCOM), 2014 IEEE*, Oct 2014, pp. 501–506.

[78] C. Popper, M. Strasser, and S. Capkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques,"

*Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 5, pp. 703–715, June 2010.

[79] E.-K. Lee, S. Oh, and M. Gerla, "Randomized channel hopping scheme for anti-jamming communication," in *Wireless Days (WD), 2010 IFIP*, Oct 2010, pp. 1–5.

[80] B. DeBruhl, C. Kroer, A. Datta, T. Sandholm, and P. Tague, "Power napping with loud neighbors: optimal energy-constrained jamming and anti-jamming," in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*. ACM, 2014, pp. 117–128.

[81] S. Khattab, D. Mosse, and R. Melhem, "Jamming mitigation in multi-radio wireless networks: Reactive or proactive?" in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*. ACM, 2008, p. 27.

[82] I. Shin, Y. Shen, Y. Xuan, M. T. Thai, and T. Znati, "A novel approach against reactive jamming attacks." *Ad Hoc & Sensor Wireless Networks*, vol. 12, no. 1-2, pp. 125–149, 2011.

[83] Y. Liu and P. Ning, "Bittrickle: Defending against broadband and high-power reactive jamming attacks," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 909–917.

[84] Y. Xuan, Y. Shen, N. Nguyen, and M. Thai, "A trigger identification service for defending reactive jammers in wsn," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 793–806, May 2012.

[85] I. Shin, Y. Shen, Y. Xuan, M. T. Thai, and T. Znati, "Reactive jamming attacks in multi-radio wireless sensor networks: an efficient mitigating measure by identifying trigger nodes," in *Proceedings of the 2nd ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing*. ACM, 2009, pp. 87–96.

[86] V. Anantharam and S. Verdu, "Bits through queues," *Information Theory, IEEE Transactions on*, vol. 42, no. 1, pp. 4–18, 1996.

[87] G. Morabito, "Exploiting the timing channel to increase energy efficiency in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 8, pp. 1711–1720, 2011.

[88] L. Galluccio, G. Morabito, and S. Palazzo, "TC-Aloha: A novel access scheme for wireless networks with transmit-only nodes," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 8, pp. 3696–3709, August 2013.

[89] Y. Liu, D. Ghosal, F. Armknecht, A.-R. Sadeghi, S. Schulz, and S. Katzenbeisser, "Hide and seek in time—robust covert timing channels," in *Computer Security–ESORICS 2009*. Springer, 2009, pp. 120–135.

[90] W. Xu, W. Trappe, and Y. Zhang, "Anti-jamming timing channels for wireless networks," in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 203–213.

[91] R. D. Pietro and G. Oligeri, "Silence is golden: Exploiting jamming and radio silence to communicate," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 3, pp. 9:1–9:24, Mar. 2015.

[92] S. Anand, S. Sengupta, and R. Chandramouli, "An attack-defense game theoretic analysis of multi-band wireless covert timing networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[93] K. Xu, Q. Wang, and K. Ren, "Joint ufh and power control for effective wireless anti-jamming communication," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 738–746.

[94] W. Xu, "On adjusting power to defend wireless networks from jamming," in *Mobile and Ubiquitous Systems: Networking Services, 2007.*

*MobiQuitous 2007. Fourth Annual International Conference on*, Aug 2007, pp. 1–6.

[95] Open Networking Foundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, 2012.

[96] M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with OpenSketch." in *Proc. of NSDI*, vol. 13, 2013, pp. 29–42.

[97] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, "Composing software defined networks." in *Proc. of NSDI*, 2013, pp. 1–13.

[98] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, 2008.

[99] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC*, Aug. 2009.

[100] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "NetLord: A scalable multi-tenant network architecture for virtualized datacenters," in *Proceedings of the ACM SIGCOMM 2011 Conference*, Aug. 2011.

[101] C. S. Mower, M. A. Palmer, and S. C. Mayhew, "Networking as a service: delivering network services using remote appliances controlled via a hosted, multi-tenant management system," Jan. 1 2013, US Patent 8,347,355.

[102] R. Yu, G. Xue, V. Kilari, and X. Zhang, "Network function virtualization in the multi-tenant cloud," *IEEE Network*, vol. 29, no. 3, May 2015.

[103] S. Shenker, M. Casado, T. Koponen, and N. McKeown, "The future of networking, and the past of protocols," *Open Networking Summit*, 2011.

[104] H. Jin, D. Pan, J. Liu, and N. Pissinou, "OpenFlow-based flow-level bandwidth provisioning for CICQ switches," *IEEE Transactions on Computers*, vol. 62, no. 9, Sep. 2013.

[105] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-Serve: Load-balancing web traffic using OpenFlow," *ACM SIGCOMM Demo*, Aug. 2009.

[106] L. Sun, K. Suzuki, C. Yasunobu, Y. Hatano, and H. Shimonishi, "A network management solution based on OpenFlow towards new challenges of multitenant data center," in *Proceedings of the 9th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT)*, Nov. 2012.

[107] A. TaheriMonfared and C. Rong, "Multi-tenant network monitoring based on software defined networking," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, ser. Lecture Notes in Computer Science, vol. 8185, 2013.

[108] Y. Shin, S. Kang, J. Kwak, B. Lee, and S. Hyang, "The study on configuration of multi-tenant networks in SDN controller," in *Proceedings of the 16th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2014.

[109] Z. Bozakov and P. Papadimitriou, "AutoSlice: Automated and scalable slicing for software-defined networks," in *Proceedings of the 2012 ACM CoNEXT Student Workshop*, Dec. 2012.

[110] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang, "Meridian: an SDN platform for cloud network services," *IEEE Communications Magazine*, vol. 51, no. 2, 2013.

[111] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "FlowVisor: A network virtualization layer," *OpenFlow Switch Consortium, Tech. Rep*, 2009.

[112] S. Gutz, A. Story, C. Schlesinger, and N. Foster, "Splendid isolation: A slice abstraction for software-defined networks," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 79–84.

[113] D. Shue, M. J. Freedman, and A. Shaikh, "Performance isolation and fairness for multi-tenant cloud storage." in *OSDI*, vol. 12, 2012, pp. 349–362.

[114] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 242–253.

[115] A. Rai, R. Bhagwan, and S. Guha, "Generalized resource allocation for the cloud," in *Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 15.

[116] J. Huang, R. A. Berry, and M. L. Honig, "Auction-based spectrum sharing," *Mobile Networks and Applications*, vol. 11, no. 3, 2006.

[117] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction protocols for decentralized scheduling," *Games and Economic Behavior*, vol. 35, no. 1, 2001.

[118] F. Martignon, S. Paris, I. Filippini, L. Chen, and A. Capone, "Efficient and truthful bandwidth allocation in wireless mesh community networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, Feb. 2015.

[119] P. Maillé and B. Tuffin, "Pricing the internet with multibid auctions," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, 2006.

[120] T. V. Ganesh. (2012, Jan.) Towards an auction-based internet.

[121] ——. (2013, Feb.) A method for optimal bandwidth usage by auctioning available bandwidth using the OpenFlow protocol.

[122] A. M. et al. Software-defined networks for future networks and services, white paper based on the IEEE Workshop SDN4FNS, http://sites.ieee.org/sdn4fns/whitepaper.

[123] (http://portal.etsi.org/NFV/NFV_White_Paper.pdf) White paper on Ònetwork functions virtualizationÓ.

[124] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The journal of supercomputing*, vol. 54, no. 2, pp. 252–269, 2010.

[125] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 1107–1117, 2013.

[126] L. Wu, S. K. Garg, and R. Buyya, "Sla-based resource allocation for software as a service provider (saas) in cloud computing environments," in *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on.* IEEE, 2011, pp. 195–204.

[127] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[128] H. Moens and F. D. Turck, "Vnf-p: A model for efficient placement of virtualized network functions," in *Proc. of CNSM 2014, Rio de Janeiro, Brazil*, 2014.

[129] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Communications Magazine, Vol. 51, Issue 7, pp. 63-70, 2013.*

[130] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene, "An open framework to enable netfate (network functions at the edge)," *Proc. of IEEE Workshop on Management Issues in SDN, SDI and NFV (Mission 2015), London, UK*, 2015.

[131] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying nfv and sdn to lte mobile core gateways, the functions placement problem," 2014.

[132] A. Lombardo, A. Manzalini, V. Riccobene, and G. Schembra, "An analytical tool for performance evaluation of software defined networking services," 2014.

[133] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc)," *Network, IEEE*, vol. 28, no. 6, pp. 18–26, 2014.

[134] J. Hwang, K. Ramakrishnan, and T. Wood, "Netvm: high performance and flexible networking using virtualization on commodity platforms," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 1, pp. 34–47, 2015.

[135] K. Giotis, Y. Kryftis, and V. Maglaris, "Policy-based orchestration of nfv services in software-defined networks," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[136] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[137] J. G. March, "Exploration and exploitation in organizational learning," *Organization science*, vol. 2, no. 1, pp. 71–87, 1991.

[138] H. J. Kushner and G. G. Yin, *Stochastic approximation algorithms and applications*. Springer, 1997.

[139] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific, 1995, no. 2.

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

[141] A. Neyman, "Correlated equilibrium and potential games," *International Journal of Game Theory*, vol. 26, pp. 223–227, 1997.

[142] W. H. Sandholm, *Population Games and Evolutionary Dynamics*, ser. Economic learning and social evolution. Cambridge, MA: MIT Press, 2010.

[143] P. Mertikopoulos, E. V. Belmega, A. L. Moustakas, and S. Lasaulce, "Dynamic power allocation games in parallel multiple access channels," in *ValueTools '11: Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools*, 2011.

[144] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical Biosciences*, vol. 40, no. 1-2, pp. 145–156, 1978.

[145] A. S. Nemirovski, A. Juditsky, G. G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[146] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, 1965.

[147] D. G. Luenberger, "Complete stability of noncooperative games," *Journal of Optimization Theory and Applications*, vol. 25, no. 4, pp. 485–505, 1978.

[148] A. Sahasrabudhe and K. Kar, "Bandwidth allocation games under budget and access constraints," in *Information Sciences and Systems,*

*2008. CISS 2008. 42nd Annual Conference on*. IEEE, 2008, pp. 761–769.

[149] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[150] P. Cramton, Y. Shoham, and R. Steinberg, "Combinatorial auctions," 2006.

[151] R. Jain and J. Walrand, "An efficient mechanism for network bandwidth auction," in *Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE*. IEEE, 2008, pp. 227–234.

[152] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory*. Oxford University Press, 1995.

[153] K. Phemius and M. Bouet, "Openflow: Why latency does matter," in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management*, May 2013.

[154] G. K. Zipf, "Human behavior and the principle of least effort." Addison-Wesley, 1949.

[155] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of IEEE INFOCOM*, Mar. 1999.

[156] A. Lazaris, D. Tahara, X. Huang, E. Li, A. Voellmy, Y. R. Yang, and M. Yu, "Tango: Simplifying SDN control with automatic switch property inference, abstraction, and optimization," in *Proceedings of the 10th ACM International Conference on emerging Networking Experiments and Technologies*, Dec. 2014.

[157] C. Goodman, "Note on existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, 1980.

[158] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Séminaire de Probabilités XXXIII*, ser. Lecture Notes in Mathematics, J. Azéma, M. Émery, M. Ledoux, and M. Yor, Eds. Springer Berlin Heidelberg, 1999, vol. 1709, pp. 1–68.

[159] J. Rice, *Mathematical statistics and data analysis*, 2006.

[160] ETSI. Network functions virtualization (NFV): Infrastructure overview.

[161] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene, "An open framework to enable NetFATE (network functions at the edge)," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–6.

[162] G. Faraci and G. Schembra, "An analytical model to design and manage a green SDN/NFV CPE node," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 3, pp. 435–450, Sept 2015.

[163] A. Mas-Colell, M. D. Whinston, J. R. Green *et al.*, *Microeconomic theory*. Oxford university press New York, 1995, vol. 1.

[164] A. Bosch-Domènech and N. J. Vriend, "Imitation of successful behaviour in cournot markets*," *The Economic Journal*, vol. 113, no. 487, pp. 495–524, 2003.

[165] G. Nan, Z. Mao, M. Yu, M. Li, H. Wang, and Y. Zhang, "Stackelberg game for bandwidth allocation in cloud-based wireless live-streaming social networks," *Systems Journal, IEEE*, vol. 8, no. 1, pp. 256–267, March 2014.

[166] G. Nan, Z. Mao, M. Li, Y. Zhang, S. Gjessing, H. Wang, and M. Guizani, "Distributed resource allocation in cloud-based wireless multimedia social networks," *Network, IEEE*, vol. 28, no. 4, pp. 74–80, July 2014.

[167] H. Tembine, E. Altman, R. El-Azouzi, and Y. Hayel, "Multiple access game in ad-hoc network," in *Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, 2007, p. 6.

[168] H. Tembine, E. Altman, and R. El-Azouzi, "Delayed evolutionary game dynamics applied to medium access control," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*. IEEE, 2007, pp. 1–6.

[169] J. Elias, F. Martignon, A. Capone, and E. Altman, "Non-cooperative spectrum access in cognitive radio networks: a game theoretical model," *Computer Networks*, vol. 55, no. 17, pp. 3832–3846, 2011.

[170] H. Tembine, E. Altman, R. El-Azouzi, and Y. Hayel, "Evolutionary games in wireless networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 3, pp. 634–646, 2010.

[171] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical biosciences*, vol. 40, no. 1, pp. 145–156, 1978.

[172] W. H. Sandholm, "Evolutionary game theory," in *Encyclopedia of Complexity and Systems Science*. Springer, 2009, pp. 3176–3205.

[173] P. Mertikopoulos and A. L. Moustakas, "The emergence of rational behavior in the presence of stochastic perturbations," *The Annals of Applied Probability*, vol. 20, no. 4, pp. 1359–1388, 2010.

[174] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Seminaire de probabilites XXXIII*. Springer, 1999, pp. 1–68.

[175] OpenFlow switch specification version 1.3.1. *Available at:* https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf.

CHAPTER

# SEVEN

# APPENDICES

## A Proof of Theorem 3

*Proof.* In order to prove the theorem we have to solve (3.55), that is, find a pair $(x, y)$ which solves the following system of equations:

$$\begin{cases} y = \chi(\Delta e^{\psi(y)+1}) \\ x = \Delta e^{\psi(y)+1} \end{cases} \tag{7.1}$$

By exploiting the Lambert W-function definition and the relationship $z/W(z) = e^{W(z)}$, where $z = \left[\frac{2(T_{AJ}+y)}{e\Delta}\right]$, it can be proven that the above system leads to

$$(y + T_{AJ})^2 = \frac{1}{\eta} \cdot \frac{\psi^2(y)}{\psi(y)+1} \tag{7.2}$$

Given that the first derivative of the Lambert W-function is defined as

$$W'(z) = \frac{W(z)}{z(W(z)+1)} \tag{7.3}$$

(7.2) can also be rewritten as

$$e^{W\left(\frac{2(T_{AJ}+y)}{e\Delta}\right)} = \frac{1}{\eta} \cdot \frac{2}{\Delta e} \cdot W'\left(\frac{2(T_{AJ}+y)}{e\Delta}\right) \tag{7.4}$$

Note that the function on the left-hand side is strictly increasing, while the one on the right-hand side is strictly decreasing. These structural properties imply that the two functions have no more than one intersection point. Therefore, the game admits a unique NE.

Now we focus on finding a closed form for the unique NE.

To this purpose, (7.4) can be reformulated as

$$e^{2W\left(\frac{2(T_{AJ}+y)}{e\Delta}\right)}(W\left(\frac{2(T_{AJ}+y)}{e\Delta}\right)+1)=\frac{1}{\eta}\left(\frac{2}{e\Delta}\right)^2$$

which, by exploiting the relation $z = W(z)e^{W(z)}$, can be rewritten as follows:

$$W\left(\frac{2(T_{AJ}+y)}{e\Delta}\right)=\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right)-1 \tag{7.5}$$

It is easy to prove that (7.5) has the following solution

$$y^*=\frac{\Delta}{2}\left(\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right)-1\right)e^{\frac{1}{2}W\left(\frac{8}{\eta\Delta^2}\right)}-T_{AJ} \tag{7.6}$$

By substituting (7.6) in (3.52) we obtain $x^* = \Delta e^{\frac{1}{2}W(\frac{8}{\eta\Delta^2})}$. As the point $(x^*, y^*)$ has been obtained as the intersection between the best response functions in (3.52) and (3.53), it follows that $(x_{\mathrm{NE}}, y_{\mathrm{NE}}) = (x^*, y^*)$ is the unique NE.

Finally, we prove that the NE $(x_{\mathrm{NE}}, y_{\mathrm{NE}})$ is an *interior* NE. An interior NE happens when it is not on the border of the strategy set; therefore, we aim at proving that $x_{\mathrm{NE}} > 2\Delta$ and $y_{\mathrm{NE}} > 0$. As $x_{\mathrm{NE}} = \Delta e^{\frac{1}{2}W(\frac{8}{\eta\Delta^2})}$, proving that $x_{\mathrm{NE}}$ is not on the border is trivial; from (7.6) it can also be easily proven that the condition $y_{\mathrm{NE}} > 0$ implies $0 < c_T < \tilde{c_T}$, where $\tilde{c_T}$ is given in (3.57); therefore, an interior NE exists only if $0 < c_T < \tilde{c_T}$. Theorem 2 states that an NE must exist for any given weight parameter $c_T$. Since we already proved that an interior NE exists only if $0 < c_T < \tilde{c_T}$, we can deduce that the NE is on the border if $c_T \geqslant \tilde{c_T}$ .

From (3.53) we know that for $c_T \geqslant \tilde{c_T}$ the best response function of the jammer, $b_J(x)$, is continuous, and it is upper-bounded by $b_J(\hat{x})$ where $\hat{x} = \Delta e^{\frac{1}{2}W(\frac{2}{\eta\Delta^2})}$, and lower-bounded by 0; thus, as the NE has to be at the

border, it follows that the only feasible solution is $y_{\mathrm{NE}} = 0$. Hence, from (3.52) and (3.53), it is easy to derive closed form solutions on the border NE, $(x_{\mathrm{NE}}, y_{\mathrm{NE}}) = \left(\Delta e^{W(\frac{2T_{AJ}}{e\Delta})+1}, 0\right)$, which concludes the proof. □

# B Proof of Lemma 3

*Proof.* To prove the Lemma, it will be shown that the condition in (3.60) implies that the Jacobian matrix norm $||J_{\mathbf{b}}||_\infty$ in (3.58) is lower than 1. In fact, the condition $||J_{\mathbf{b}}||_\infty < 1$ leads to:

$$\max\left(\left|\frac{\partial}{\partial y}b_T(y)\right|, \left|\frac{\partial}{\partial x}b_J(x)\right|\right) < 1$$

Note that $\left|\frac{\partial}{\partial y}b_T(y)\right|$ can be calculated as

$$\left|\frac{\partial}{\partial y}b_T(y)\right| \quad = \quad \frac{2}{W(\frac{2(T_{AJ}+y)}{e\Delta})+1}$$

The above function is non-negative and strictly decreasing, thus it achieves its maximum value when $y = 0$. Accordingly, it is sufficient to show that

$$\max_{y\in\mathcal{S}_\mathcal{J}}\left(\frac{2}{W(\frac{2(T_{AJ}+y)}{e\Delta})+1}\right) < 1 \quad , \quad \forall y \geq 0$$

or, equivalently, that

$$\max_{y\in\mathcal{S}_\mathcal{J}}\left(\frac{2}{W(\frac{2(T_{AJ}+y)}{e\Delta})+1}\right) = \frac{2}{W(\frac{2T_{AJ}}{e\Delta})+1} < 1 \quad , \quad \forall y \geq 0$$

which is indeed satisfied for all values of $y$ in the strategy set; therefore, $\left|\frac{\partial}{\partial y}b_T(y)\right| < 1, \forall y \in \mathcal{S}_\mathcal{J}$.

Concerning the condition $\left|\frac{\partial}{\partial x}b_J(x)\right| < 1$, by deriving $b_J(x)$, it follows that

$$\left|\frac{1}{2}\left[\frac{1}{x\sqrt{\eta\log\frac{x}{\Delta}}} - 1\right]\right| < 1 \tag{7.7}$$

The expression on the right-hand side of (7.7) is a non-negative strictly

decreasing function, so again (7.7) results in

$$\max_{x\in\mathcal{S}_{\mathcal{T}}}\left(\left|\frac{1}{2}\left[\frac{1}{x\sqrt{\eta\log\left(\frac{x}{\Delta}\right)}}-1\right]\right|\right)<1 \tag{7.8}$$

Note that (7.8) can be rewritten in the form given in (3.60) and $||J_{\mathbf{b}}||_\infty = ||J_{\mathbf{b}}||$ as $J_{\mathbf{b}}$ is diagonal. Let $s^i=(x^i,y^i)$, it then follows that

$$||s^{i+1}-s^i||\leq||J_{\mathbf{b}}^{max}||\cdot||s^i-s^{i-1}||\leq\cdots\leq||J_{\mathbf{b}}^{max}||^i||s^1-s^0||$$

where $||J_{\mathbf{b}}^{max}|| = \max J_{\mathbf{b}}$. The above equation indicates that given any $\epsilon>0$, after at most $\log_{J_{\mathbf{b}}^{max}}\frac{\epsilon}{||s^1-s^0||}$ iterations, the game converges to the NE as $||s^{i+1}-s^i||\leq\epsilon$ which thus concludes the proof. □

# C Proof of Proposition 8

*Proof.* The mean dynamic of (4.37) is

$$\begin{cases}\dot{z}_i = v_i(\mathbf{b})\\ b_i = B_i\frac{e^{z_i}}{1+e^{z_i}}\end{cases} \tag{7.9}$$

At any given time $t$, let $\mathbf{b}(t)$ be a solution for (7.9). In system theory, such solution is often referred to as *solution orbit* or *trajectory* of the system. In the following, we show that i) $\mathbf{b}(t)$ converges to $\mathbf{b}^*$ as $t\to+\infty$, and ii) (4.37) is an *asymptotic pseudo-trajectory* (APT) [174] for the mean dynamic (7.9) and converges to $\mathbf{b}^*$ if some mild conditions on the step-size are satisfied.

From Proposition 7, we have that $U_i^{(\mathcal{S})}(\mathbf{b})$ is a strictly concave function in $b_i$. Therefore, $v_i(\mathbf{b})(b_i-b_i^*)<0$ for all $b_i\in[0,B_i]$ by definition. By exploiting this latter result, it can be shown that the function $V(\mathbf{b})$ defined as

$$V(\mathbf{b})=\sum_{i\in\mathcal{V}_{\mathcal{S}}}B_i\ln\left(\frac{B_i-b_i^*}{B_i-b_i}\right)+b_i^*\ln\left(\frac{b_i^*}{b_i}\cdot\frac{B_i-b_i}{B_i-b_i^*}\right) \tag{7.10}$$

is a *strict Lyapunov function* for (7.9). In fact, we have that $\dot{V} = dV(\mathbf{b})/dt = \sum_{i\in\mathcal{V}_S} v_i(\mathbf{b})(b_i - b_i^*) < 0$, $V(\mathbf{b}^*) = 0$ and $V(\mathbf{b}) > 0$ for all $\mathbf{b} \neq \mathbf{b}^*$. It can be shown that $V(\mathbf{b})$ is *radially unbounded*. A function $V(\mathbf{b})$ is said to be radially unbounded if $V(\mathbf{b}) \to \infty$ when $\|\mathbf{b}\| \to \infty$. Therefore, the equilibrium point $\mathbf{b}^*$ is also *globally asymptotically stable* (GAS), which leads to the conclusion that $\mathbf{b}(t)$ converges to $\mathbf{b}^*$ as $t \to +\infty$. Now, we prove the second part of the proposition which consists in showing that also the discrete-time algorithm asymptotically converges to the equilibrium. By decoupling (7.9), we get

$$\dot{b}_i = \frac{db_i}{dt} = b_i \left(1 - \frac{b_i}{B_i}\right) v_i(b) \tag{7.11}$$

The latter result will be useful to show that the discrete-time algorithm tracks the continuous-time system up to a bounded error that asymptotically tends to 0 as $i$ increases.

A second-order Taylor expansion of (4.37) leads to

$$b_i(m+1) = b_i(m) + \gamma_m b(m) \left(1 - \frac{b_i(m)}{B_i}\right) v_i(\mathbf{b}(m)) + \frac{1}{2}\mu\gamma_m^2 \tag{7.12}$$

for some bounded $\mu$ . Note that $\mu$ is bounded as $\frac{\partial}{\partial b_i} v_i(\mathbf{b})$ is bounded by definition. Intuitively, (7.12) is the discrete version of (7.11) up to a bounded error. Since, by assumption, $\sum_m \gamma_m^2 < \sum_m \gamma_m = +\infty$, theoretical results in [174] show that $b_i(m)$ is an APT for (7.9).

It still remains to prove that $b_i(m) \to b_i^*$. By decoupling $z_i$ and $b_i$, we obtain $z_i = \ln\left(\frac{b_i}{B_i - b_i}\right)$. By rewriting $V(\mathbf{b})$ in terms of $\mathbf{z}$, we obtain $V(\mathbf{z})$. By considering a Taylor expansion of $V(\mathbf{z})$, we obtain

$$V(\mathbf{z}(m+1)) = V(\mathbf{z}(m)) + \gamma_m \sum_{i\in\mathcal{V}_S} (b_i(m) - b_i^*)\, v_i\,(b_i(m)) + \frac{1}{2}\mu'\gamma_m^2 \tag{7.13}$$

for some bounded $\mu' > 0$.

Since $\mathbf{b}^*$ is GAS, it follows that $\mathcal{B}$ is a basin of attraction for $\mathbf{b}^*$. Therefore, there must exist a compact set $\mathcal{L} \subset \mathcal{B}$ containing $\mathbf{b}^*$. If we prove that there also exists a large enough $m'$ such that $\mathbf{b}(m') \in \mathcal{L}$, then, the proof is

concluded. Assume ad absurdum that such $m'$ does not exists. Recall that $v_i(\mathbf{b})(b_i(m) - b_i^*) < 0$ by definition. Therefore, it must exist some $\beta > 0$ such that $\sum_{i \in \mathcal{S}} v_i(\mathbf{b})(b_i(m) - b_i^*) \leq -\beta$ for a large enough $m$. It follows that

$$V(\mathbf{z}(m+1)) \leq V(\mathbf{z}(m)) - \gamma_m \beta + \frac{1}{2}\mu' \gamma_m^2 \tag{7.14}$$

which yields to

$$V(\mathbf{z}(m+1)) \leq V(\mathbf{z}(0)) - \beta \sum_m \gamma_m + \frac{1}{2}\mu' \sum_m \gamma_m^2 \tag{7.15}$$

By assumption $\sum_m \gamma_m^2 < \sum_m \gamma_m = +\infty$. Thus, (7.15) leads to $V(\mathbf{z}(m+1)) \leq -\infty$, which is a contradiction as $V(\mathbf{z})$ is lower bounded by construction. Therefore, [174] ensures that there must exist $m'$ such that $\mathbf{b}(m') \in \mathcal{L}$ and $\lim_{m \to +\infty} \mathbf{b}(m) = \mathbf{b}^*$, which concludes the proof.

□