UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

ENHANCEMENTS IN SPECTRUM MANAGEMENT TECHNIQUES FOR HETEROGENOUS 5G FUTURE NETWORKS

Autor: Vincenzo Sciancalepore,
Universidad Carlos III de Madrid, IMDEA Networks Institute, Madrid, Spain
DEIB - Politecnico di Milano, Italy
Directores: prof. Albert Banchs, Universidad Carlos III de Madrid
prof. Antonio Capone, DEIB - Politecnico di Milano

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés (Madrid), septiembre de 2015

UNIVERSIDAD CARLOS III DE MADRID

PH.D. THESIS

ENHANCEMENTS IN SPECTRUM MANAGEMENT TECHNIQUES FOR HETEROGENOUS 5G FUTURE NETWORKS

Author: Vincenzo Sciancalepore,
Universidad Carlos III de Madrid, IMDEA Networks Institute, Madrid, Spain
DEIB - Politecnico di Milano, Italy
Directors: prof. Albert Banchs, Universidad Carlos III de Madrid
prof. Antonio Capone, DEIB - Politecnico di Milano

DEPARTMENT OF TELEMATIC ENGINEERING

Leganés (Madrid), September 2015

Enhancements in Spectrum Management Techniques for Heterogenous 5G Future Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Prepared by
Vincenzo Sciancalepore,
Universidad Carlos III de Madrid, IMDEA Networks Institute, Madrid, Spain
DEIB - Politecnico di Milano, Italy

Under the advice of
prof. Albert Banchs, Universidad Carlos III de Madrid
prof. Antonio Capone, DEIB - Politecnico di Milano

Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid

---

Date: septiembre, 2015

Contact: vincenzo.sciancalepore@imdea.org

This work has been supported by IMDEA Networks Institute.

TESIS DOCTORAL

ENHANCEMENTS IN SPECTRUM MANAGEMENT TECHNIQUES FOR HETEROGENOUS 5G FUTURE NETWORKS

Autor: Vincenzo Sciancalepore,
Directores: prof. Albert Banchs, Universidad Carlos III de Madrid
prof. Antonio Capone, DEIB - Politecnico di Milano

Firma del tribunal calificador:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Abstract

In the last decade, cellular networks are undergoing with a radical change in their basic design foundations. The huge increase in traffic demand requires a novel design of future cellular networks. Driven by this increase, a network densification phenomena is occurring thereby, which in turns requires to devise efficient and reliable mechanisms to deal with the interference problems resulting from such densification. The architecture and mechanisms resulting from such drastic re-design of the network are commonly referred under the term '5G network'.

In this context, this work unveils that current networking solutions are no longer sufficient to (i) provide the required network spectral efficiency, and (ii) guarantee the desired level of quality of experience from the user side. In order to address this problem, in this thesis we propose a novel SDN-like framework that incorporates the needed mechanisms to improve spectral efficiency while delivering the desired quality of experience to users. In particular, our architecture includes the following two approaches:

- Our first approach addresses the intercell interference issues resulting from high network densification. To this end, we propose novel mechanisms to mitigate the inter-cell interference problem. We address the design of such schemes from two angles: (i) a controller-aided mechanism, which gathers all the information of the network at a centralized point and, based on this information, optimally schedules the transmission from different users, and (ii) a semi-distributed mechanism, which limits the signaling overhead involved in sending the information to a centralized point while providing close to optimal performance. One of the key novelties of our scheduling algorithms is that they are based on the Almost Blank SubFrame (ABSF) scheme; indeed, this scheme has been standardized only recently and very little work has addressed the design of algorithm to use it.

- Our second approach addresses spectral efficiency from a complementary angle: cellular traffic offloading for content update applications. This approach leverages high user mobility to offload the cellular downlink traffic through a device-to-device communication. In this context, we propose an adaptive algorithm to decide how to optimally transmit content to base stations in order to maximize traffic offload. By relying on control theory techniques, our approach delivers near optimally performance.

- A third key contribution of this thesis is the design of a solution that combines the

above two approaches. In particular, our solution takes into account that traffic offload is taking place in the network and addresses the design of an optimal scheduling algorithm that leverages on the Almost Blank SubFrame (ABSF) scheme. Indeed, the combination of these kind of approaches has received little attention from the literature.

The feasibility and performance of the approaches described above are thoroughly evaluated and compared against state-of-the-art solutions through an exhaustive simulation campaign. Our results show that the proposed approaches outperform conventional eICIC techniques as well as standard offloading mechanisms, respectively, and confirm their feasibility in terms of overhead and computational complexity.

To the best of our knowledge, this thesis is the first attempt to design an unified framework which is able to optimally perform offloading for content-update distribution applications while boosting the network performance in terms of spectral efficiency

List of Publications

[1] H. Ali-Ahmad, C. Cicconetti, A De La Oliva, M. Draxler, R. Gupta, V. Mancuso, L. Roullet, and V. SCIANCALEPORE. Crowd: An sdn approach for densenets. In *Software Defined Networks (EWSDN), 2013 Second European Workshop on*, pages 25–31, Oct 2013.

The material of this publication can be found in Chapter 1.

[2] V. SCIANCALEPORE, V. Mancuso, and A. Banchs. BASICS: Scheduling base stations to mitigate interferences in cellular networks. In *IEEE WOWMOM*, 2013. The material of this publication can be found in Chapter 3.

[3] V. SCIANCALEPORE, I. Filippini, V. Mancuso, A. Capone, and A. Banchs. A semi-distributed mechanism for inter-cell interference coordination exploiting the absf paradigm. In *IEEE SECON*, 2015.

The material of this publication can be found in Chapter 4.

[4] V. SCIANCALEPORE, D. Giustiniano, A. Banchs, and A. Picu. Offloading cellular traffic through opportunistic communications: Analysis and optimization. *Accepted for publication at IEEE Journal on Selected Areas in Communications*, 2015.

The material of this publication can be found in Chapter 5.

[5] V. SCIANCALEPORE, V. Mancuso, A. Banchs, S. Zaks, and A. Capone. Interference coordination strategies for content update dissemination in LTE-A. In *IEEE INFOCOM*, 2014.

The material of this publication can be found in Chapter 6.

Table of Contents

List of Tables

List of Figures

Part I

Background

Chapter 1

Introduction

The rapid growth of mobile data traffic in conjunction with the increasing expectation for high network performance is dramatically pushing network operators to speed-up the introduction of future cellular technologies, termed as 5G future networks. In particular, the steady increase of traffic demand in current cellular networks requires new technologies and network architectures able to handle such demands [6]. The growth foreseen cannot be only sustained by increasing the spectrum assigned to mobile radio networks. In fact, spectrum availability is already scarce in the ranges of practical interest, and spectral efficiency achieved by today's technologies, such as Long Term Evolution (LTE and LTE-A), is already close to Shannon's capacity limit. This leads to a huge densification of wireless networks, which seriously degrades the network performance due to the high interference and the low efficiency in the utilization of spectrum in Radio Access Networks (RANs) [6]. Therefore, support for densification of interference control is the key-enabler for future 5G networks.

In this context, it has been shown that frequency-reuse-1 can provide substantial improvements in terms of efficient utilization of the scarce and expensive wireless resources. This implies that neighboring base stations (BSs) should be allowed to transmit on all available time-frequency resource blocks simultaneously, thus causing strong interference to each other's users. To this aim, several techniques have been proposed to independently cope with interference or low spectral efficiency in RANs, such as beamforming, MIMO or many others, as shown in [18]. None of them assume that these issues are highly intertwined because reducing the impact of either one strongly affects the others, thus seriously impairing the cellular network performance. This contrasts with interference mitigation and/or cancellation techniques that have been used for many years in the past, which basically exploited orthogonality of frequency and/or spatial resources [19]. More recently, advanced solutions have been designed which actively reduce or cancel interference when orthogonality cannot be guaranteed [15, 87].

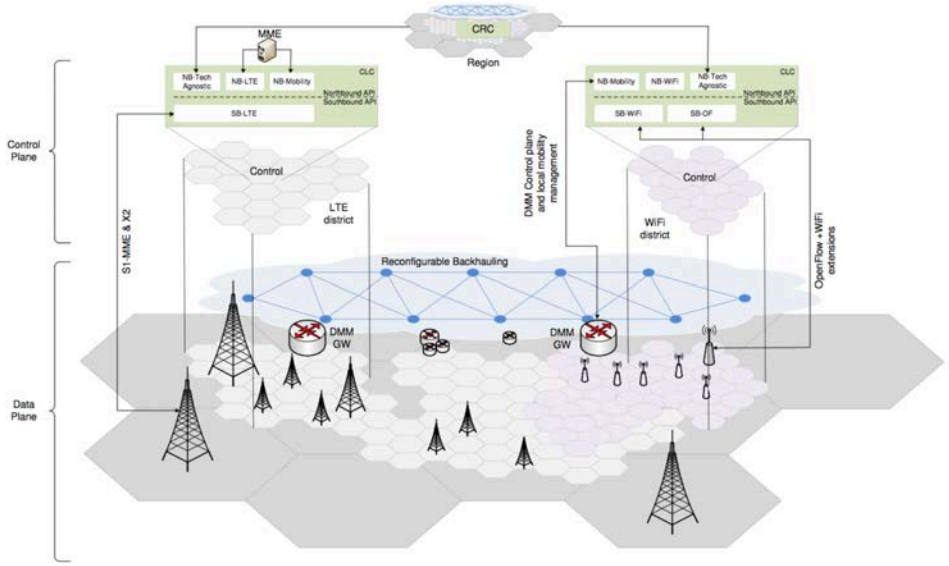


Figure 1.1: Future cellular network architecture based on CROWD Project [33].

1.1. New Software Defined Wireless Network Architectures

To cope with all the above mentioned issues, a novel framework is required to provide optimization mechanisms, dynamic and *high-density proof* provisioning of resources. The recent Software Defined Network (SDN) paradigm [57] is therefore a natural candidate to design an architecture able to provide and manage the required network solutions. To this aim, we rely on a novel architecture, presented in the FP7 CROWD Project [33] [1]. In particular, due to density of the network and the expected computational overhead, a dynamic two-tier SDN controller hierarchy is proposed, as envisaged in Fig. 1.1. The architecture is structured into two logical tiers: districts with a limited, but fine grain scope for short time scales, and regions with a broader but more coarse grain scope for long time scales. Thus, two types of network controller are defined, as follows:

- CROWD Local Controller (CLC or LC), which can take fast, short-time scale decisions on a limited but fine grain scope;
- CROWD Regional Controller (CRC), which can take slower, long time scale decisions with a broader but more coarse grain scope.

The architectural proposal works with a network where LTE (macro/pico/femto) and WiFi cells are deployed. LTE and WiFi cells are assumed, without loss of generality, to be reconfigurable

[1] We use the CROWD proposal as reference architecture in the PhD dissertation since I was involved in the European Project FP7 CROWD (agreement 318115) along the duration of the PhD programme.

via some open protocol, e.g., OpenFlow (OF) [68]. Operations within a district are optimized by applications connected to the CLC via a set of APIs, called North Bound (NB) interface in the SDN terminology. Finally, all the network elements belong to the same administrative domain and we neglect the security measures which must be implemented in practice to prevent malicious access of the control functions and to avoid unauthorised disclosure of sensitive information from customers. Our aim is to present novel optimization mechanisms as CLC applications, aimed at enhanced wireless MAC operations in a district, which perfectly comply with the CROWD future network architecture requirements.

1.2. SDN applications

SDN approach represents a powerful solution to implement CLC applications defined in the CROWD architecture to take care of radio and MAC operation control. We structure our proposals into two main categories: (i) Enhanced Inter-Cell Interference Coordination (eICIC) for conventional cellular networks and (ii) Advanced Offloading Techniques for future heterogenous networks.

The former application focuses on a set of advanced solutions which aim at abating the intercell interference while boosting the spectral efficiency of the system. Our idea is to build a SDN application which aims at orchestrating base stations transmission activities within a local area covered by multiple base stations and pico/femto/small cells. After requiring monitoring and filtering of interference statistics, scheduling decisions are taken to coordinate transmissions. As a guideline, 3GPP recently proposed the *Almost Blank Sub-Frame* (ABSF or ABS [88]) scheme to implement efficient inter-cell interference coordination (ICIC). Initially suggested to enable small-cell transmissions within heterogenous environments, it has been smartly extended to provide acceptable results even when applied to homogenous scenarios, where only macro base station are involved. The scheme temporarily inhibits transmissions at a particular macro cells, as shown in Fig. 1.2. Specifically, macro base stations may silent a specific set of subframes where vulnerable users could be suffered from high neighbouring interference. However, for those blanked subframes, cell reference symbols (CRS) must be active to provide channel measurements. Most interestingly, the ABSF paradigm is simple enough to be accepted for inclusion into technically entangled 3GPP specifications and at the same time it has found rather wide acceptance among the standardization institutions [26].

Cellular offloading mechanisms are also required to enhance the overall spectral efficiency through a Device-to-Device (D2D) opportunistic communication. In particular, the second application is in charge of deciding how conventional cellular transmissions should be offloaded to users adopting the D2D paradigm. Moreover, the D2D offloading application could also decide whether users can form clusters whose cluster leader relays the cellular traffic for all cluster members using WLAN connectivity (e.g., WiFi Direct).

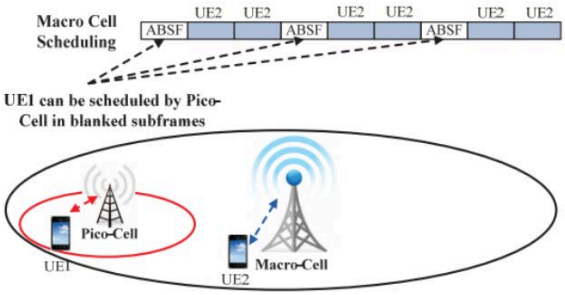


Figure 1.2: Almost Blank SubFrame (ABSF) explanation scheme [88].

1.3. Enhanced Inter-cell Interference Coordination: centralized vs distributed

A high-speed backhauling enables the CLC to correctly gather all cellular user information, such as channel status (CSI), in real-time and to properly take scheduling decisions, which are promptly translated into ABSF patterns which each base station must apply for the next transmission period. As extensively proved by the literature, such centralized solutions achieve near-optimal solutions in terms of system spectral efficiency and thus, energy efficiency. However, this condition holds only when simplistic assumptions are taken into consideration. The huge explosion of mobile applications we are witnessing requires that network operators must introduce traffic guarantees for their customer contracts, burdening the network capacity. The compound effect of including traffic constraints with user and base station network densifications could be even further dramatic when considering complexity and delays. That is the reason why we are forced to consider a distributed approach, which must be enabled when network conditions are no longer under the CLC control. A distributed eICIC proposal with local decisions would not only be aligned with the well accepted self-organizing network concepts [78], but also allows to make ABSF and user scheduling decisions jointly—rather than assuming worst case conditions for the user scheduling process—which allows for further improving performance. Note that one critical aspect in the design of the distributed scheme is to limit the amount of information exchanged between base stations as well.

Therefore, we design a semi-distributed mechanism, which reduces the computational burden from a centralized controller (CLC) while drastically abating the signaling overhead. In particular, the ABSF coordination of local schedulers (base stations) is aided by the local controller, which supervises the ABSF decisions and drives the system to the best possible performance without requiring overall statistics and imposing centralized decisions. This makes our approach a first step towards a practical and effective solution of ABSF that can be implemented in real networks.

1.4. How to offload conventional cellular network: Device-to-Device Communication

A huge volume of traffic for mobile devices seems to be strictly required by the appearance of a great number of web and smartphone applications. This popularization of smartphones and the ensuing explosion of mobile data traffic [7] lead to conventional cellular networks which are currently overloaded, and even worse in the near future [6]. A large portion of that traffic consists in the distribution of content updates such as social network updates, road traffic updates, map updates, and news feeds (e.g., *waze*, an *app* for a social network for navigation, includes all the above mentioned features).

Along with the appearance of such applications, some schemes have been recently proposed to offload the traffic generated by them in the cellular network. In particular, the device-to-device (D2D) paradigm has been proposed to assist the base station in the content distribution [20, 96, 104]: with D2D communications enabled, the base station delegates a few interested mobile users (*content injection*) to carry and opportunistically spread the content updates to others interested upon meeting them (*content dissemination*). Indeed, opportunistic communication exploits the daily mobility of users, which enables intermittent *contacts* whenever two mobile devices are in each other's proximity. These contacts are used to transport data through the opportunistic network, which may introduce substantial delays. However, the type of content concerned by cellular offloading may not always be entirely delay-tolerant. In many applications, it is indeed critical that the content reach all users before a given deadline, lest it lose its relevance or its usability. Therefore, the design of opportunistic-based cellular offloading techniques faces serious challenges from the intermittent availability of transmission opportunities and the high dynamics of the mobile contacts. In order to find the best trade-off between the *load* of the cellular network and the *delay* until the content reaches the interested users, any opportunistic-based offloading design must answer crucial questions such as, *how many* copies of the content to inject, to *which users* and *when*.

While most of the currently available offloading proposals focus on the characterization of content dissemination and the design of content injection strategies, they largely neglect the optimization of radio resources in the *injection phase*, i.e., the process of injecting a content in a subset of the mobile user population, which produces bursty and periodic traffic. Some existing work partially considered the impact of opportunistic resource utilization in the content injection strategies but their analysis is restricted to a single cell and does not consider the interference caused by other cells, which is a key limiting factor for the deployment of dense and heterogeneous networks that are expected to appear in 5G cellular systems [79]. In line with the 5G networks view, we leverage the heterogeneity of technologies in the network to implement a novel D2D-based offloading mechanisms, which also tackles the cellular traffic offloading issue from a different and unexplored perspective: the intercell interference coordination problem. The rationale behind our approach is twofold: (i) interference is a key factor in future networks, where

the single cell study case is not representative of a real network; (ii) content injection operations are impacted by network speed, which, in turn, strongly depends on intercell interference.

1.5. Outline of the thesis

In this work we provide in details a set of novel optimization solutions for enhanced MAC operations, which aim at analyzing and proposing enhanced Intercell Interference coordination mechanisms (eICIC) for drastically reducing the inter-cell interference problem while improving the spectral efficiency through a D2D opportunistic offloading of conventional cellular traffic. Our research thesis starts with a survey of the current technological features applied to cellular networks, such as LTE or LTE-Advanced networks. Then, we study in details how centralized solution for eICIC may severally impact on the network performance in terms of aggregate system throughput. When network densification phenomena is no longer negligible, we present our findings on a distributed architecture where the number of exchanged messages between different base stations is kept very low. Later, we mathematically analyze the cellular offloading procedure through D2D opportunistic transmissions for content dissemination applications, providing a smart tool to properly reduce the cellular burden while keeping reasonable the content delivery delays. Lastly, we focus on the injection phase during the content dissemination process, where inter-cell interference issue may impair the system performance, if no accurately managed.

The remainder of this thesis is structured as follows:

- In Chapter 2, we provide a detailed study of the state of the art. We structured the chapter to show existing solutions for each of the topics addressed by this thesis. In addition, we point out a set of significant works which are considered as benchmark in our performance evaluation sections.

- In Chapter 3, the inter-cell interference problem is properly presented from a centralized perspective. The CLC collects user channel statistics and makes decisions, which rely on the ABSF patterns, as explained above. A low-complexity algorithm is devised to generate ABSF patterns based on collected user data. Simulation results are shown by means of a commercial network simulator, such as Opnet Modeller 17.1.

- In Chapter 4, we face the densification problem by showing that centralized solutions are no longer reliable in terms of complexity and signalling overhead. In addition, we introduce a new class of downlink guaranteed traffic characterized by bit-rate constraints. Therefore, we propose a novel lightweight semi-distributed solution, which optimally scales with the number of users placed in the entire network. We propose a convergence study based on game theory notions. Finally, we benchmark our proposal against other existing solutions, ranging from very complex power control schemes to simple and affordable ICIC techniques.

- In Chapter 5, we introduce the downlink cellular traffic offloading principles when content update applications are in place. Based on control theory paradigm, we propose an adaptive algorithm to minimize the cellular transmissions in order to deliver the required contents to all intended users within the content lifetime. We compare our approach with other state-of-the-art solutions, showing the validity and the feasibility of our novel mechanism.

- In Chapter 6, we mainly target the injection phase during the content dissemination process by shedding the light on how to optimally perform the content updates distribution when inter-cell interferences are caused by neighbouring base station transmissions. Specifically, we analytically and empirically prove that minimizing injection phase through an efficient eICIC mechanism leads to outstanding dissemination results and in turn, to near-optimal cellular downlink traffic offloading performance.

- In Chapter 7, final remarks and conclusions are drawn and some open issues are discussed.

Chapter 2

Related Works

We deeply investigated the state-of-the-art works, previously suggested in the literature, to provide a valid benchmark for our proposal. We structure this chapter by following the main topics addressed in this thesis.

2.1. Intercell Interference Coordination

Intercell interference issues have been addressed by several researchers and still play a key role in the future network requirements definition. For instance, the authors of [19] provide an overview of techniques that can be exploited to mitigate inter-cell interference in OFDM-based networks. Interestingly, they do not identify base station scheduling as a possible tool to reduce interference, and limit their discussion to beamforming, coding and decoding techniques, opportunistic spectrum access, interference cancellation, power control and (fractional) frequency reuse. CoMP shows a very good gain by mitigating interference exploiting cooperation between sector transmitters or different base stations [46]. CoMP proposals range from complex distributed MIMO solutions to advanced beamforming mechanisms. E.g., Multi-Cell Joint Transmission [65] proposes to share the same data to transmit across multiple base stations, while Coordinated Beamforming Scheduling [56] proposes a method to choose transmission beam patterns in coordination between base stations. In both cases, a large backhaul capacity is required for inter-base station communication, over both data and control planes, which realistically prevents implementation in real systems. Most of the work available in the literature focuses on *user scheduling*, in terms of beamforming, CoMP, and power allocation. For instance, random beamforming has been proposed to reduce the need for BS-to-BS CQI information exchange [87]. It enhances the per-cell throughput at limited cost, at least in small networks. Fast distributed beamforming in multi-cell environments was also proposed in [15], in which scheduling is performed in two steps: first each base station chooses the proper beamforming that minimizes intercell interference and then a particular user is scheduled in each cell. However, none of these two works address fairness. The authors of [55] propose to leverage power control mechanisms to

achieve frequency reuse 1 in multicellular environments. Joint scheduling, beamforming, power allocation with proportional fairness is the objective of [101]. In that work, the three problems are disjointly and iteratively addressed, so that the proposed algorithm requires several optimization iterations, for which the authors do not provide complexity analysis. The authors of [17] use dynamic programming to optimize the transmission probability of base stations in a TDMA network. Their analysis only yields a probabilistic scheduling model for base stations and cannot be realistically applied to existing cellular technologies. Additionally, their solution can only be used in case of networks consisting of two base stations, or with homogeneous topologies. Although originally proposed for WiMAX systems, RADION represents another valid approach to intercell interference mitigation [98]. RADION is a distributed resource management framework that manages interference across femtocells and enables femtocells to opportunistically find the available resources in a distributed manner, by performing three different actions: client categorization, resource decoupling and two-phase adaptation and allocation. A similar and interesting approach is presented in [8] where a management system, called FERMI, is introduced for OFDMA-based femtocell networks. It performs a clients classification in order to identify which clients need resource isolation and those that require just link adaptation, then it incorporates a frame zoning structure that supports the coexistence of clients from both categories. Afterwards, the system allocates orthogonal sub-channels of the OFDMA spectrum in a fair manner. The authors of [75] describe a vertex coloring technique to schedule frequencies among femto-base stations while mitigating inter-femtocell interference. The schema proposed in [75] is based on user rather than base station scheduling, and does not provide quality guarantees to any of the users. As for the work focusing on *base stations* instead of users, proposed solutions are available for frequency reuse and fractional frequency reuse schemes where the entire available set of frequencies is divided a priori and assigned to adjacent cells, or portions of cells, to avoid strong interference between neighbouring cells. The disadvantage of such schemes lies in their scarce flexibility to adapt to changing cell load conditions. A scheme proposed in [83] suggests to divide the cell into two zones, namely edge and center, and assigns users to each zone dynamically based on their channel state information; however, the gain is limited due to the low number of available frequency bandwidths. The authors of [32] present a game theoretical approach to ICIC. Their approach addresses the coordination among base stations over a set of finite resources as a non-cooperative game. However, they only target the minimization of the perceived interference, and do not take into account user scheduling. All in all, none of the above proposals embodies the set of features that characterize our approaches: in Chapter 3 a centralized solution is envisioned to boost the system network performance while in Chapter 4 a distributed scheme exhibits outstanding results when compared to some of the above-mentioned approaches. At best of our knowledge, there is no literature on the use of ABSF techniques for properly serving inelastic traffic.

2.2. Almost Blank SubFrame

Recently, 3GPP has standardized the ABSF (almost blank subframe) technique [69], which allows base stations to blank data transmissions over one or several subframes. However, the standard does not specify the algorithm to decide which base stations remain silent and which ones do not in a given subframe. The algorithm that we propose here addresses precisely this question, and can be used to drive the ABSF mechanism. The ABSF technique is becoming popular because it is suitable for eICIC in LTE, it requires minimal changes in the operation of base stations and offers flexible tools to trade-off between performance improvement and implementation complexity ([36,95]). However, designing a mechanism to drive ABSF decisions has turned out to be challenging and multifaceted. For instance, the authors of [26] have studied quantitative approaches aiming to determine the best *density* of blanked subframes as a function of the traffic distribution. Other studies focus on heterogeneous scenarios where a macro base station and several small base stations have to coordinate their activities using ABSF patterns ([47,50,51]). Other proposals include access selection in the loop and introduce the concept of Cell Selection Bias ([31]), which improves network spectral efficiency ([48, 84]). However, existing ABSF solutions either require a central entity to gather per-user CSI or need additional and continuously updated information on, e.g., topology and propagation environment, which goes well beyond current base station's features and capabilities. As a result, existing ABSF solutions are not scalable and do not adapt quickly when network conditions change. Some other solutions for resource management behave similarly to ABSF. For instance, a recent proposal for ODFMA femtocells has been presented in [99]. Although the authors do not explicitly use the ABSF paradigm, their work is based on detecting the best region of the time-frequency space where base stations can transmit, like in ABSF. However, they propose a *probe-and-adapt* algorithm to decide whether to use or blank resources. Moreover, they do not require coordination between base stations, which would yield performance limitations. Similarly, the authors of [85] propose the concept of *reuse patterns* for base station activities, which clearly mimics ABSF operations. However, their work focuses on finding the best temporal duration of reuse patterns (in order to maximize the total user throughput) but it does not explain how to generate reuse patterns.

2.3. Content updates Dissemination

While the unsustainable increase in cellular network traffic negatively affects the conventional cellular system in terms of spectral efficiency, several works have been already proposed to cope with this issue. For instance, [61,94] are aiming to exploit the relatively large number of existing WLAN access points, as well as cellular diversity. A different approach, based on new infrastructure, is introduced in [66], in the context of vehicular networks. In that work, the authors advocate the deployment of fixed roadside infrastructure units and study the performance of the system in offloading traffic information from the cellular network. Slightly different are the proposals inves-

tigating the use of infrastructure-free opportunistic networking as a complement for the cellular infrastructure. In particular, the studies in [40,45,54,63,96] propose solutions based on this idea. In [45], the authors propose to push updates of dynamic content from the infrastructure to subscribers, which then disseminate the content epidemically. The distribution of content updates over a mobile social network is shown to be scalable, and different rate allocation schemes are investigated to maximize the data dissemination speed. Han et al. investigate, in [40], which initial subset of users (who receive the content through the cellular) will lead to the greatest infection ratio. A heuristic algorithm is proposed, that uses the history of user mobility of the previous day to identify a target set of users for the cellular deliveries. In [54], an architecture is implemented to stream video content to a group of smartphones users within proximity of each other, using both the cellular infrastructure and WLAN ad-hoc communication. The decision of who will download the content from the cellular network is based on the phones' download rates. However, the focus of [54] is on the implementation rather than the model and the algorithm. Indeed, the algorithm proposed is a simple heuristic, which does not guarantee optimal performance. Another study where opportunistic networking is used to offload the mobile infrastructure is [63]. Here, some chosen users, named "helpers", participate in the offloading, and incentives for these users are provided by using a micro-payment scheme. Alternatively, the operator can offer the participants a reduced cost for the service or better quality of service. Thus, the focus of [63] is on incentives, which is out of the scope of our work. Most interesting is the Push-and-Track solution, presented in [96]. There, a subset of users initially receive content from a content provider and subsequently propagate it epidemically. Upon reception of the content, every node sends an acknowledgment to the provider, which may decide to re-inject extra copies to other users. Upon reaching the content deadline, the system enters into a "panic zone" and pushes the content to all nodes that have not yet received it. Specifically, Push-and-Track relies on a heuristic to choose *when* to feed more content copies into the opportunistic network, which does not guarantee that the load on the cellular network is minimized. Other works focus on content dissemination solutions in purely opportunistic networks [42, 44]. However, most of these studies focus on finding the best ways to collaborate or contribute to the dissemination, under various constraints (e.g., limited "public" buffer space). Evaluation is usually based on the delay incurred to obtain desired content or the equivalent metric of average content freshness over time. In contrast, we build our analytical model and propose a dynamic and adaptive algorithm to guarantee near-optimal performance in the content distribution process, as explained in Chapter 5.

All the previous works on offloading cellular networks through opportunistic communications assume that all the transmissions over the cellular network are unicast. There are several key reasons that limit the usage of multicast messages in a cellular network. First, multicast cannot be easily combined with opportunistic transmissions, as this would require that the Content Server is aware of the cell of each node and can dynamically select the subset of nodes at each cell that receives the multicast message, which is not possible with current cellular multicast approaches. Second, in urban scenarios users will likely be associated to different base stations (there are

hundreds/thousands of them in the city, each covering some sector, and in dense urban areas femtocells have started to be deployed). Thus, there is a low probability that users subscribed to a specific content are associated to the same cells at the same time, and hence multicast may collapse to unicast. Finally, transmissions with multicast would occur at the lowest rate to preserve users in the edge of the cell, which degrades the resulting performance.

Finally, none of the above works tackle the impact of interference in dense scenarios, in presence of offloading traffic strategies. The authors of [80, 81] design a heuristic to allocate resource blocks when adjacent cells interfere with each other. Their approach allows the reuse of resource blocks in cell centers, while users at the cell edge, which suffer higher interference, cannot be allocated specific resource blocks, as figured out by the proposed heuristic. Indeed, that work only considers avoiding the interference of the two most interfering base stations. Similarly, the proposal in [62] assigns resource blocks via a central entity while [102] solves the problem in a distributed manner. However, they allocate resources not only to base stations but also to users, based on backlog and channel conditions. The author of [71] uses graph theory to model network interference. That work proposes a graph coloring technique to cope with interference coordination, based on two interference graphs: one *outer* graph using global per-user interference information, and an *inner* graph using local information, available at the base station, and global constraints derived from the global graph. To reduce complexity, [71] uses genetic algorithms to seek a suboptimal resource block allocation. Conversely, we provide an efficient and low-complexity solution to boost the cellular traffic offloading by significantly reducing the inter-cell interference during the content injection phase, as explained in Chapter 6.

Part II

Spectrum Management for Conventional Cellular Networks: Design of Smart ICIC

Chapter 3

Central Coordinator-aided eICIC Mechanism

In this chapter, we tackle the problem of inter-cell interference mitigation from a centralized viewpoint. In particular, a central network controller (CLC) is in charge of scheduling *base stations* rather than *users*. We propose to coordinate base station downlink activities in order to mitigate the interference caused to neighboring cells. To do this, we propose a method to map base stations' activities onto subframes with regular patterns. We aim to mitigate interference by limiting the activity of a given base station to some subframes while forcing it to remain silent in the other subframes. Noticeably, our work is suitable for driving subframe blanking decisions in self organising networks, e.g., as per the ABSF (almost blank subframe) technique recently defined by 3GPP [69]. A key feature of our proposal is that it incurs a very reduced signaling load between base stations. Indeed, we propose to coordinate base station downlink activities in order to limit the interference caused to *any* possible user in the system under *any* possible user scheduling decision taken by the base stations. Therefore, base stations do not need to exchange information with a per-user granularity but rather on a much coarser basis. Another key feature of our design is that we decouple the problem of mitigating downlink inter-cell interference from the problem of optimizing the user scheduling. While the focus of this chapter is on the first problem, our proposal can easily be combined with existing user scheduling schemes to further improve spectral efficiency (as we show in the performance evaluation section). By means of numerical and packet-level simulations, we prove the effectiveness and superiority of our solution as compared to the state of the art of inter-cell interference mitigation schemes.
The key features can be summarized as follows:

- we formulate a novel base station scheduling problem and show that it is NP-hard in strong sense;

- we design an algorithm, called BASICS (BAse Station Inter-Cell Scheduling), that runs in polynomial time and scales with the number of users;

- we show that BASICS not only achieves better throughput performance with respect to state of the art schedulers, but also significantly improves fairness among users.

3.1. Scheduling Problem

In this section, we describe in details the considered system, and formulate our base station scheduling problem for this system. We then prove that this problem is NP-hard by mapping it onto a well-known NP-hard optimization problem, namely the Multidimensional Vector Bin-Packing Problem. Finally, we derive some bounds for the problem's solution.

3.1.1. System Model

We consider a multicellular LTE-like environment with $N = |\mathcal{N}|$ base stations and $U = |\mathcal{U}|$ mobile users. We address only downlink transmissions, for which no power control is adopted, as in the majority of state of the art proposals. Each base station schedules its users across subframes, as specified in LTE systems [93], each subframe lasting 1 *ms*. The duplexing scheme adopted is FDD, with 20MHz bandwidth for each transmission direction. Unless otherwise specified, all base stations use the same frequencies. Users associate to the base station from which they receive the strongest signal, and transmission rates are selected, in each subframe, according to the Signal-plus-Noise Interference Ratio (SINR), see Table 3.1. The SINR for a certain user $u \in \mathcal{U}$ is defined as follows:

$$\text{SINR} := \frac{S_b^u}{N_0 + \sum_{j \neq b} I_j^u},$$

where N_0 is the background noise, S_b^u is the useful signal received by the current user u from the serving base station b (hereafter defined as S^u) and I_j^u is the interference sensed by the user u from any other base station j in the system, when that base station is scheduled. Signal and interferences received by each user are affected by Rayleigh fading, and user channels are assumed to be independent.

In our system, we focus on mitigating interference by deciding whether a base station can transmit during a given subframe. We refer to this decision as *base station scheduling*, to be distinguished from legacy *user scheduling* which occurs at each base station when it is allowed to transmit.

3.1.2. Base Station Scheduling Optimization Problem

We aim at guaranteeing a minimum SINR to every user in the system by allocating in each subframe a subset of the available base stations while minimizing the number of subframes needed in order to schedule the complete set of base stations. We will discuss in Section 3.4.1 how to select that minimum SINR. Meanwhile, here we focus on the first part of the problem: decide whether a base station can transmit in a given subframe.

The goal of our proposal is to schedule base station transmissions in each subframe so that the SINR is greater than a threshold Th for every user u in the network (i.e., for any user that can receive a transmission from a scheduled base station):

$$\frac{S_b^u}{N_0 + \sum_{j \neq b} I_j^u} \geq \text{Th}. \tag{3.1}$$

For convenience of notation, we now call I_b^u the signal received by user u from its BS, i.e., $I_b^u = S_b^u = S^u$, therefore Eq. (3.1) can be rewritten as follows:

$$\sum_{j=1}^{n} I_j^u \leq \frac{S^u}{\text{Th}} - N_0 + S^u \triangleq \text{Th}^u. \tag{3.2}$$

With the above, the problem of minimizing the total number Z of subframes used to schedule once all N base stations in the system[1], for a given minimum SINR (or threshold Th), is formulated as follows:

Problem `BS-Scheduling`:

$$\begin{aligned}
\text{minimize} \quad & Z = \text{number of subframes needed to} \\
& \qquad \text{allocate all base stations once,} \\
\text{subject to} \quad & \textstyle\sum_j I_j^u \, x_{ij} \leq \text{Th}^u, u \in \mathcal{U}, i \in \{1, \ldots, Z\}, \\
& \textstyle\sum_i x_{ij} = 1, \quad j \in \mathcal{N}, \\
& x_{ij} \in \{0,1\}, \quad i, j \in \mathcal{N};
\end{aligned}$$

where

$$x_{ij} = \begin{cases} 1, & \text{if base station } j \text{ is scheduled into subframe } i, \\ 0, & \text{otherwise.} \end{cases}$$

Note that Z is the analogous in the time domain of the frequency reuse factor. However, our approach has two main advantages over frequency reuse schemes. First, we need only one frequency allocated to the system, which is less expensive than using frequency reuse. Second, differently from the frequency reuse factor, which is static, Z can vary from time to time when network conditions change (i.e., with users' arrival or departure).

Theorem 3.1.1. *Minimizing Z as defined in Problem* `BS-Scheduling` *is NP-hard in strong sense.*

Proof: The number of subframes needed to allocate all base stations at least once is upper bounded by the number of base stations, i.e., $Z \leq N$. Consider now the problem of scheduling

[1]For sake of simplicity, we assume that each base station has to be scheduled only once per scheduling cycle. However, our approach can be easily extended to schedule base station B_i n_i times in a scheduling cycle by increasing the set of the candidate base stations with N_i replicas of base station B_i, where replicas have an infinite interference between each other.

each base station exactly once in Z consecutive subframes. If $Z < N$, then $N - Z$ subframes are left empty. Let us define a set of N binary variables y_i, which indicate whether a subframe $i \in \mathcal{N}$ is used or empty:

$$y_i = \begin{cases} 1, & \text{if subframe } i = 1, \ldots, N \text{ is used,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that the concept of *empty subframe* is only an abstraction to simplify the description of the problem. In fact, $Z < N$ means that the scheduling of base stations over Z subframes is repeated cyclically, with period Z subframes. With the above notation, we can re-write Problem `BS-Scheduling` as follows:

Problem `kD-VBP`:

$$\begin{array}{ll} \text{minimize} & Z = \sum_i y_i, \\ \text{subject to} & \sum_j I_j^u \, x_{ij} \leq \text{Th}^u \, y_i, u \in \mathcal{U}, i \in \mathcal{N}, \\ & \sum_i x_{ij} = 1, \quad j \in \mathcal{N}, \\ & y_i \in \{0,1\}, \quad i \in \mathcal{N}, \\ & x_{ij} \in \{0,1\}, \quad i, j \in \mathcal{N}. \end{array} \tag{3.3}$$

The above is the formulation of a k-dimensional vector bin-packing problem (kD-VBP) with $k = U$ being the number of mobile users in the system [67].

Our problem is therefore equivalent to a k-dimensional vector bin-packing problem, in which knapsacks represent subframes, items to be allocated are base stations, and the number of dimensions is given by the number of users, each imposing a constraint on its SINR as expressed in Eq. (3.2). Since our problem has been mapped onto kD-VBP, it can thus be classified as an NP-hard problem in strong sense. ■

3.1.3. Lower Bound

The value Z in Problem `BS-Scheduling` determines the portion of time during which a base station is prevented from transmitting (i.e., each base station is scheduled with frequency $1/Z$). Therefore, the lower bound for Z in our problem represents the highest scheduling frequency that can be associated to base stations in the system with SINR not lower than Th for any of the users. Thus, in order to have high efficiency in the utilization of resources, our goal is to design an algorithm that finds the smallest possible value for Z.

In the following, we obtain a lower bound for Z, which bounds the best possible performance that we can achieve. This bound provides a benchmark against which we can evaluate the performance of our solution, as we do in the performance evaluation section.

Theorem 3.1.2. *The lower bound $L \leq Z$ for Problem `BS-Scheduling` with k users distributed over N base stations, and a guaranteed SINR $\geq$ Th for all users, is given by the following equa-*

tion:

$$L = \max_{u} \left(\left\lceil \sum_{j=1}^{N} \frac{I_j^u}{Th^u} \right\rceil \right). \tag{3.4}$$

Proof: Our proof follows the same approach used in [52]. First, we recall that in vector bin-packing problems items cannot be *rotated*, i.e., the constraints are defined on a per-dimension basis, and dimensions cannot be rearranged. In our case, a dimension represents the interference caused to a given user, which justifies why dimensions cannot be rearranged. In particular, the minimum number of subframes (bins) needed to accommodate all the base stations (items) in such a way that the max interference (constraint) on the uth dimension is not violated is given by the ratio between the sum of all interferences caused by N base stations in the uth dimension, i.e., for the uth user, divided by the threshold Th^u, which represents the capacity of the bin in the uth dimension. Of course, only integer numbers are allowed, hence we need $\left\lceil \sum_{j=1}^{N} I_j^u / \text{Th}^u \right\rceil$ bins to satisfy the constraint in the uth dimension. Since all dimensions are independent, the result follows. ■

3.2. Algorithm Design

To solve the problem formulated in Section 3.1.2, we next propose a heuristic consisting in a greedy algorithm for the mapping of base stations to subframes. The algorithm is designed to dynamically mitigate inter-cell interference caused to any possible user in the system under any possible user scheduling decision taken by the base stations. As a result, our algorithm is *user-scheduling-agnostic* and does not require coordinated scheduling among base stations.

We propose a new heuristic rather than using existing heuristics for two main reasons. First, existing heuristics for multidimensional vector bin-packing problems are simple extensions of solutions designed for the one-dimension problem. Second, existing heuristics do not take into account the nature of the dimensions that describe the items to be allocated. In particular, they assume that the size of an object is the same in any of the possible combinations of items in a bin. In contrast, in our case, the size of an object is the interference caused to mobile users *belonging to the scheduled base stations only*. Therefore, the weight associated to a base station (i.e., its *size*) changes any time a base station is removed from the list of candidate transmitters (e.g., since it is allotted to a subframe).

In the following, we first briefly discuss existing algorithms for solving multidimensional vector bin-packing problems, then we present our novel solution and highlight the difference with existing proposals. Later, in Section 3.4, we prove, by means of empirical results, that our approach outperforms existing algorithms.

3.2.1. State of the art algorithms for bin-packing problems

The most commonly adopted algorithms for solving the bin-packing problem belongs to the family of FFD-based algorithms. The First-Fit Decreasing Algorithm (FFD) was proposed to solve the one-dimensional bin-packing problem [67]. With FFD, items are sorted by size, in decreasing order, and a number of empty bins–equal to the total number of items–is set. Then, items are inserted sequentially from the largest to the smallest in the first available bin with enough capacity left.

To cope with the case of multidimensional vector bin-packing problems, various greedy FFD extensions have been proposed in the literature [28]. Available heuristics collapse all dimensions into one, and then apply the FFD algorithm proposed for the resulting one-dimensional version of the problem. The name of each algorithm depends on how the dimensions are collapsed. When all dimensions are multiplied in order to get one unique *monodimensional size* for each item, the algorithm is called *FFDProd*, whereas the algorithm *FFDSum* uses a weighted sum of the original dimensions. Other algorithms such as *FFDAvgSum* or *FFDExpSum* use similar approaches to *FFDSum* [28].

As described in [73], the above algorithms can be classified as FFD item-centric since all items are allocated until there are no items left to be placed. Another group of algorithms are classified as FFD bin-centric. The latter are algorithms which start with a single bin, and a new bin is initialized when there are no more items which can fit the previously used bins. As proved by empirical evaluations in [73], bin-centric algorithms (such as Dot-Product and Norm-based Greedy) outperform item-centric algorithms and they can sometimes reduce the number of required bins by up to 10%.

A common assumption of FFD-based algorithms is that the dimensions of an item do not change. In contrast, in Problem `BS-Scheduling`, dimensions (i.e., interferences caused by a base station transmission) *do* change with the set of items (i.e., base stations) that are included (i.e., scheduled) in the same bin (i.e., subframe). Indeed, the set of base stations scheduled in a subframe affects the set of mobile users that can receive interference, and therefore affects the number of dimensions of the problem in a given algorithm iteration. Therefore, legacy FFD-based approaches are not suitable for solving our problem, so we propose a novel approach, as described in the reminder of this Section.

3.2.2. BASICS

Interferences sensed by users play the role of dimensions in the Problem `BS-Scheduling`. Thus, we propose BASICS (BAse Station Scheduling Inter-Cell Scheduling), a sum-based algorithm which solves kD-VBP problems by collapsing all problem dimensions (i.e., the interferences to different users) into one unique value. This value is computed for each base station, and consists in the total interference caused by the base station to users belonging to other scheduled base stations.

However, in our problem, the size of the items (base stations) to be allocated into bins (subframes) *changes* at any iteration of the algorithm. In particular, BASICS represents a modification of the FFDSum algorithm in which (i) the size of each item to be accommodated changes at each iteration, and (ii) items are accommodated into bins in order, beginning with the smallest one. As in bin-centric approaches, BASICS allocates a new bin only when there is no more room left in the old bins to accommodate the remaining items. Note that existing algorithms for kD-VBP would rather sort items from the largest to the smallest.

The rationale behind our approach is as follows. First, when we start allocating base stations from the least interfering one, we have a chance to schedule together the highest number of not-previously-allocated base stations in the same subframe. This eliminates the highest number of base station candidates for the next subframe allocation. In turn, considering a uniform distribution of users, this procedure eliminates the highest number of users from the set of interfered users in the next iteration of the algorithm. As a result, the cumulative interference over the remaining users, due to the remaining candidate base stations in the next iteration, is likely to be much lower than in the previous iteration. In contrast, if we removed a base station generating less interference, we would have a high probability that that base station interfered fewer users. Thus, removing the least interfering base station would not only bring less benefit to the current subframe, but also we would not reduce much the impact of that base station in the next subframe allocation (since the set of potential interfered users did not change much). Interestingly, our interference sorting approach is similar to the one presented in [70], which focuses on groups of interfering users.

The details of the BASICS algorithm are presented in Algorithm 1, and described in the following. Initially, the algorithm computes the interference generated by any base station to any user in the system (lines 1 to 6). This computation is performed by means of a simple free space propagation model accounting for the transmission power of the base stations as well as the position of base stations and mobile users. Then, the algorithm checks whether the entire set of base stations can be active in the same subframe, i.e., the entire set of base stations forms the initial base station candidate set. This check is performed by comparing users' SINR thresholds against the SINR experienced when all base stations are active (line 10). At this point, if all SINR constraints are met, then all base stations are allocated into the current subframe, and the algorithm ends. Otherwise, the algorithm computes the overall interference figure due to each base station, and sorts the base stations in decreasing order. The overall interference figure of a base station is computed by summing up all interferences caused by that base station (line 11). Once base stations are sorted, the algorithm removes the most interfering base station from the set of candidate base stations (lines 12-13). The algorithm then re-checks SINR constraints for the subnetwork obtained by removing the most interfering base station and all its users from the original network. The procedure is repeated by removing the most interfering base station (and its users) at each iteration, until all SINR constraints are met. The resulting set of candidate base stations is allocated to the first subframe. Next, the algorithm has to run again for the subnet

Algorithm 1: BASICS: heuristic to allocate base stations into subframes guaranteeing a minimum SINR for each user.

Input and variables
$\mathcal{W}$: set of all base stations in the system
$\mathcal{A}$: set of base stations not yet allocated
$\mathcal{T}_i$: candidate set for subframe i
$\mathcal{U}$: set of all users
N_0: background noise
Th: minimum SINR
i: subframe index
Initialization
$\mathcal{A} \leftarrow \mathcal{W}$
Procedure

1: **for** each $u \in \mathcal{U}$ **do**
2: Compute Th^u from N_0, Th, and S_u
3: **for** each $j \in \mathcal{W}$ **do**
4: Compute the signal strength I_j^u at user u from base station j
5: **end for**
6: **end for**
7: $i \leftarrow 0$
8: **while** $|\mathcal{A}| > 0$ **do**
9: $\mathcal{T}_i \leftarrow \mathcal{A}$
10: **while** $\exists u \mid \sum_{j \in \mathcal{T}_i} I_j^u > \text{Th}^u$ **do**
11: $\forall j \in \mathcal{T}_i, I_j \leftarrow \sum_u I_j^u$
12: $k \leftarrow \arg\max\{I_j\}$
13: $\mathcal{T}_i \leftarrow \mathcal{T} \setminus \{k\}$
14: **end while**
15: $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{T}_i$
16: $i \leftarrow i + 1$
17: **end while**

consisting of the base stations not previously allocated, i.e., the set of base stations that were removed during the first algorithm loop (line 15). The output of the ith algorithm loop is the list of base stations to be scheduled in the ith subframe. When all base stations are allocated, the algorithm ends returning the complete base station scheduling plan.

3.2.3. Optimal Setting of the Threshold

One of the key parameters upon which BASICS relies is the SINR threshold Th, i.e., the minimum SINR guaranteed to each scheduled user in the system. If we set this threshold to a very low value, this means that we do not impose minimum SINR requirements, which corresponds to the normal network operation without BASICS. Conversely, if we set a very high value, this implies that the constraint on minimum SINR can be fulfilled only by scheduling no interfering base stations at all, which corresponds to a pure TDM system in which each base station transmits in isolation. In the following, we address the issue of finding the optimal threshold which lies in between these two extremes. Specifically, we provide a method to efficiently compute an approximation to the optimal threshold Th and we later empirically prove our finding in Section 3.4.1.

In order to find the threshold setting, we look for the threshold value that maximizes the average downlink throughput in the network (fairness is already ensured by allocating the same

number of subframes to all base stations). The average downlink throughput over all users in the system depends on Th through the following relation:

$$R_{avg}(\text{Th}) = \frac{1}{Z(\text{Th})} \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \frac{1}{|U_b|} R_u(\text{Th}), \tag{3.5}$$

where $Z(\text{Th})$ is the total number of subframes needed to allocate all base stations, $\mathcal{B}$ is the set of all base stations, $\mathcal{U}_b$ is the set of users of base station b, and $R_u(\text{Th})$ is the average transmission rate to user u.

In order to obtain the total number of subframes needed to allocate all base stations as a function of the threshold, $Z(\text{Th})$, we assume that the bin packing algorithm executed by BASICS works perfectly and is able to completely fill all bins. In this case, the number of bins required is proportional to the size of the items, which in its run is proportional to Th (see Theorem 3.1.2 and the relation between Th and Th^u expressed in Eq. (3.2)). Thus,

$$Z(\text{Th}) = K\,\text{Th}, \tag{3.6}$$

where K is a constant term.

Similarly, in order to obtain the transmission rates as a function of Th, we assume that all the users of a given cell suffer a similar level of interference ($\sum_{j \neq b} I_j$). In this way, if user v is the user of base station b with the smallest S_u value, we can compute $\sum_{j \neq b} I_j$ by imposing $\frac{S_v}{N_0 + \sum_{j \neq b} I_j} = \text{Th}$.

Once we obtain $\sum_{j \neq b} I_j$ from the weaker user of base station b, we can then compute the SINR of all the users as a function of Th, and from these one can further obtain the transmission rates $R_u(\text{Th})$.

With the above, we have characterized all the terms of Eq. (3.5) as a function of the threshold Th. The optimal threshold value can then be obtained simply by finding the Th value that maximizes $R_{avg}(\text{Th})$, which can be easily done by running a numerical search. Note that this value does not depend on the constant term K of Eq. (3.6).

3.2.4. Computational complexity of BASICS

Next, we evaluate the computational complexity of the proposed algorithm. Let N be the number of base stations in the system and U the number of users (i.e., the number of dimensions for a multi-dimensional bin packing problem). Before the first round of checking for SINR constraints, the algorithm computes all signal strengths from N base stations to U users. This operation has computational cost $O(N \cdot U)$ (see lines 1-6 of the Algorithm 1). Then, each subframe is inspected in order to check if the candidate base stations meet the SINR constraints for each of their users. For the first subframe allocation, the algorithm will perform, for each user, N multiplications, $N - 1$ sums, and 1 division. In the worst case, each round of checking fails, which leads to the elimination of the most interfering base station and its users, which we assume

to be uniformly spread over the set of base stations, i.e., we eliminate U/N users at each round. The largest number of rounds is $N-1$. Henceforth, the number of complex operations to be performed (i.e., multiplications and divisions) for the first subframe allocation is at most z_1, as computed in the following equation:

$$z_1 = U \cdot N + U\left(1 - \frac{1}{N}\right)(N-1) + \cdots + U\left(1 - \frac{N-1}{N}\right) \cdot 1. \tag{3.7}$$

The following algorithmic step consists in allocating base stations for the second subframes. In the worst case, there are now $N-1$ candidate base stations, with $U(1-1/N)$ users. Thereby the algorithm performs at most z_2 operations for this subframe:

$$z_2 = U\left(1 - \frac{1}{N}\right)(N-1) + \cdots + U\left(1 - \frac{N-1}{N}\right) \cdot 1. \tag{3.8}$$

Similarly, for the allocation in the kth subframe, the algorithm performs at most z_k operations:

$$z_k = U\left(1 - \frac{k-1}{N}\right)(N-k+1) + \cdots + U\left(1 - \frac{N-1}{N}\right) \cdot 1. \tag{3.9}$$

Then, the following result is derived by taking into account the worst case, in which exactly N subframes are used:

$$\begin{aligned} \sum_{k=1}^{N} z_k &= 1(N \cdot U) + 2\left[(N-1) \cdot U\left(1 - \frac{1}{N}\right)\right] + 3\left[(n-2) \cdot u\left(1 - \frac{2}{n}\right)\right] + \ \ldots \\ &+ N\left[1 \cdot U\left(1 - \frac{N-1}{N}\right)\right] = U \cdot \sum_{k=1}^{N} \frac{k(N-k+1)^2}{N}. \end{aligned} \tag{3.10}$$

Recalling the results for well-known sums $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$, $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$, and $\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$, the computational complexity for our algorithm is $O(U \cdot N^3)$. Therefore, our solution scales with the number of users, and, although it grows with N^3, it can be used to optimize the scheduling of realistically small groups of interfering base stations (e.g., up to $\sim$10 neighboring base stations).

3.3. Evaluation Tools

In this section we present the two software tools used to evaluate BASICS: (i) a mathematical tool, namely MATLAB, that gives a first evaluation of the impact of BASICS in multi-cellular environments without going into the intricacies of LTE detailed implementation,[2] and (ii) the OPNET Modeler simulator, which allows to evaluate the specific impact of LTE protocols onto

[2]Available online:`http://people.networks.imdea.org/~vincenzo_sciancalepore/MATLAB_LTE_23072012.zip`

our results, although such simulations can only be carried out in small network scenarios only due to the computational cost and time required by the simulator.[3]

3.3.1. MATLAB implementation

MATLAB provides a suitable set of mathematical tools to implement and evaluate BASICS and scheduling mechanisms without going into packet level simulations. Our MATLAB implementation operates as follows. In the first step, the system is initialized, i.e., the positions of base stations are chosen at random in a square area, and mobile stations are dropped in the same area according to a uniform spatial distribution. Users are associated to base stations based on the strongest average received power, i.e., based on distance, and do not change base station during the simulation. The average received power depends on the transmission power set at each base station and on the pathloss, according to the classical *Free Space formulation*. Fading is considered in the numerical simulations in addition to pathloss, through a random variable, expressed in dB, distributed as a zero-mean Gaussian with standard deviation equal to 2 dB.

The second phase is to run the BASICS algorithm to decide which base station has to transmit in which subframe. In the algorithm, we use a unique SINR threshold for all the users.

Eventually, in the last phase, we calculate the throughput received in 1000 consecutive frames by each user. We repeat the simulation with different random seeds, averaging the results.

Note that the throughput depends on the channel state simulated, which affects the transmission rate achievable in the current frame, and on the user scheduling mechanism adopted by the base station. As for transmission rates, LTE specifications define 16 different CQI indexes, which correspond to different Modulation and Coding Schemes (MCS's), as described in Table 3.1. The mapping of SINR values onto CQI and transmission rates is done as follows. First, we compute the spectral efficiency η corresponding to the SINR using the Shannon's formula:

$$\eta = \log_2\left(1 + \frac{SINR}{\Gamma}\right), \tag{3.11}$$

where Γ is a coefficient that depends on the target BER (which is equal to 0.00005 in our case): $\Gamma = -ln(5 \cdot BER)/1.5$ [13]. Second, using Table 3.1, we find the interval for η that corresponds to the SINR, and use the efficiency reported in the rightmost column as the net rate per allocated symbol used in the subframe. Since each subframe is divided in 2 time slots, each time slot contains 100 Physical Resource Blocks (PRBs) and each PRB is structured in $7 \cdot 12 = 84$ OFDMA symbols, the maximum number of OFDMA symbols assigned to one user in a single subframe is $84 \cdot 100 \cdot 2 = 16800$. The maximum throughput we can get in this case will be exactly $16800 \cdot 5.5547 = 93318.96$ bit/subframe, i.e., 93.318 Mb/s as the subframe lasts 1 ms. Eventually, taking into account the adopted user scheduling scheme, MATLAB computes the number of PRBs to be allotted to each user, and computes the corresponding throughput.

[3] Available online:`http://people.networks.imdea.org/~vincenzo_sciancalepore/OPNET_patch_23072012.zip`

Table 3.1: LTE CQI index and efficiency

Modulation Scheme	Approximate code rate	CQI Index	Interval for η	Efficiency (bits/symbol)
No transm.	–	0	0	–
QPSK	0.076	1	$0 \div 0.15$	0.1523
	0.12	2	$0.15 \div 0.23$	0.2344
	0.19	3	$0.23 \div 0.38$	0.3770
	0.3	4	$0.38 \div 0.60$	0.6016
	0.44	5	$0.60 \div 0.88$	0.8770
	0.59	6	$0.88 \div 1.18$	1.1758
16QAM	0.37	7	$1.18 \div 1.48$	1.4766
	0.48	8	$1.48 \div 1.91$	1.9141
	0.6	9	$1.91 \div 2.40$	2.4063
64QAM	0.45	10	$2.40 \div 2.73$	2.7305
	0.55	11	$2.73 \div 3.32$	3.3223
	0.65	12	$3.32 \div 3.90$	3.9023
	0.75	13	$3.90 \div 4.52$	4.5234
	0.85	14	$4.52 \div 5.12$	5.1152
	0.93	15	≥ 5.12	5.5547

As for the scheduling of users, we implement a basic round robin scheme, allotting equal airtime to each user in round robin order, and a state of the art proportional fairness scheduler [14]. The latter allots resources according to user priorities computed at the beginning of each subframe as the ratio between the achievable rate in that frame, and the average throughput received in the past.

Although the MATLAB implementation misses the impact of detailed LTE and network-layer protocols, e.g., the impact of mechanisms used for generating realistic traffic, or for computing link adaptation and physical resource allocation in OFDMA, it does provide a platform that allows to efficiently simulate large network scenarios while providing a reasonable level of accuracy.

3.3.2. OPNET Modeler

To evaluate the impact of real protocols on our proposal, we also modified the well-established OPNET simulator[4].

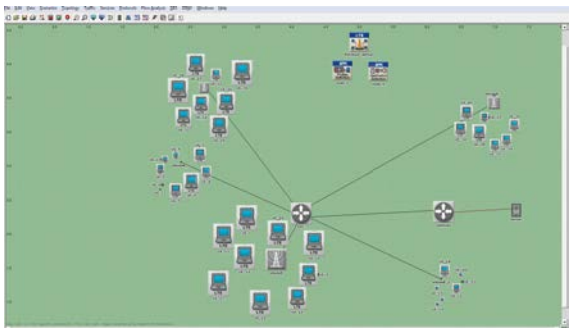


Figure 3.1: OPNET LTE scenario with 5 base stations and 40 users.

OPNET already implements several LTE scenarios, for which nodes and functionalities are

[4]OPNET University Program, `http://www.opnet.com/university_program/`

designed in a modular way. We modified the modules specifying the behavior of the base station, to simulate the control traffic needed to run BASICS, and the behavior of the physical channel, to account for dynamic fading effects not yet implemented on the simulator. Most importantly, we have implemented our base station scheduling into a central entity called Evolved Packet Core (EPC), which collects global interferences reported by users to their base stations (CQI messages). For each simulation module, new layers and process model states have been added in order to perform the newly required operations.

Furthermore, to simulate the dynamic capabilities of our proposal, we have programmed an internal interrupt for each base station, in order to collect all interferences reported by the users, prepare a control message containing such information, and send it to the EPC component. The EPC component runs BASICS periodically, and enforces a new base station scheduling with a refresh interval of $2s$, i.e., 2000 subframes. The refresh interval has been selected to track channel quality variations, while keeping the signaling overhead low.

For our scope, we need an LTE scenario with several users associated to a group of base stations served by an EPC interface. To this end, the EPC is connected to a general gateway by a serial connection (providing a speed up to 2488 Mb/s), and a server is added to serve the users' demands.

We use an OPNET-predefined video conference application to generate traffic. Specifically, all users are adjusted to request a video streaming through a UDP connection characterized by 30 frames/s, where every frame has a resolution of 352x240 pixels (i.e., 253440 bytes). The server is able to respond to each demand thereby reaching the saturation in the transmission. According to LTE specification, a single user served by a base station can reach a very high throughput, about 90 Mb/s.

As OPNET is a very complex packet simulator, each simulation takes several minutes to run over our server, which is a Dell Optiplex 990 with a Intel(R) Core(TM) i7-2600 CPU at 3.40 GHz with 8 cores, 8 GB of RAM and Windows 7 Professional SP1 64 bit. That is the reason why it is not possible to use OPNET for a very large number of base stations. Therefore, we use OPNET for small network topologies only, while we use MATLAB for larger scenarios.

3.4. Performance Evaluation & Discussion

In this section, we evaluate the performance of BASICS by using the evaluation tools presented in the previous section and show that it achieves near-optimal results and outperforms existing solutions.

3.4.1. Selection of the BASICS Threshold *Th*

Since the most suitable threshold can fall in a continuous range of values, while the achievable rates are a discrete set corresponding to the available MCS values, we search the threshold Th

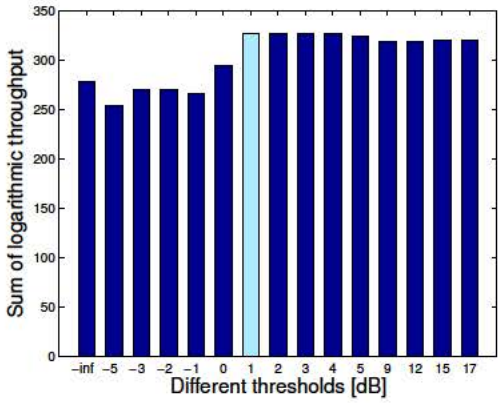


Figure 3.2: Sum of logarithmic throughputs achieved with BASICS with different thresholds (dB) in a scenario with 9 base stations and 225 users.

among the MCS thresholds corresponding to the efficiency ranges listed in Table 3.1. After evaluating all available thresholds, we select Th as the threshold that yields the best sum of logarithmic throughput; we use this well-known metric as it accounts for both throughput and fairness.

Fig. 3.2 compares the results obtained for different thresholds for a scenario with 9 base stations and 225 users. The lowest threshold reported in the figure represents the case of no minimum SINR requirements, i.e., it represents the normal network operation without BASICS. On the other hand, the highest threshold is higher than the minimum SINR required to use the highest MCS, which implies that the constraint expressed by Eq. (3.2) can be fulfilled only by scheduling no interfering base stations at all. As a result, the highest threshold corresponds to a pure TDM system in which each base station transmits in isolation. We observe from the figure that the optimal threshold lays in-between the two extremes, and provides a gain of several logarithmic units in comparison to other suboptimal thresholds.

3.4.2. Computational Complexity

We first evaluate the computational complexity of BASICS. We have already analytically addressed the worst case computational time in Section 3.2.4, where we proved that the complexity scales with the number of users in the system. Here, we measure experimentally the time required for each single execution of BASICS on our server. The experiments use our MATLAB implementation, and MATLAB's *Profile* function which returns the execution time of the software.

The experiments have been conducted with a Dell Latitude Laptop with Intel(R) Core(TM) i7 CPU at 2.80 GHz, 4,00 GB of RAM over Windows 7 Professional SP1 64 bit. The results are given in Fig. 3.3 for different numbers of base stations and users. Each point reported in the figure is the average over 10 different runs initialized with different random seeds. In the worst case, with 8 base stations and 90 users, BASICS execution takes 0.792 seconds on average. This shows that BASICS can be run in reasonable time for reasonably large networks just by means of

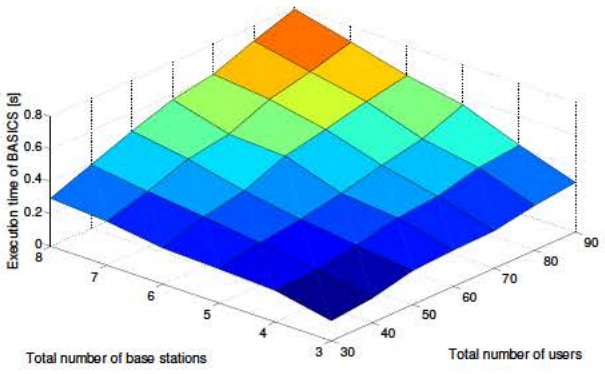


Figure 3.3: Execution time for BASICS with different combinations of number of base stations and number of users in the network.

inexpensive hardware.

3.4.3. Optimality of solution

In Section 3.1.2, we have formulated our problem as to minimize the number of subframes needed to allocate all base stations once. In the following, we evaluate the performance of BASICS and compare it against the legacy FFD bin-centric algorithms, taking the number of subframes used as the evaluation metric. To this aim, we implemented in MATLAB the FFDSum algorithm as well as a simple tool that identifies the optimal base station mapping (in terms of achieved throughput) by means of a brute-force approach.

To compare performance achieved by BASICS, FFDSum and the optimal (brute force) algorithm, we simulate a network with 3 to 7 base stations, each having 25 users. Fig. 3.4 shows that BASICS finds the same number of subframes as the optimal solution, except for the case of 5 base stations in which it uses one extra subframe. Furthermore, BASICS uses at most one subframe more than the theoretical lower bound, obtained from Eq. (3.4). In contrast, FFDSum achieves significantly worse results (as expected from the discussion in Section 3.2). Fig. 3.5 further illustrates the throughput performance obtained from these algorithms, and reveals that FFDSum not only uses more subframes, but also provides worse throughput. In contrast, BASICS achieves near-optimal results.

3.4.4. Performance gain

We next evaluate the performance of BASICS in terms of throughput and fairness by using MATLAB, which allows to explore the impact of a large number of base stations and users under various network configurations.

In order to assess its performance, we compare BASICS against the following two approaches: (i) normal network operation, in which all base stations are allowed to transmit in any subframe

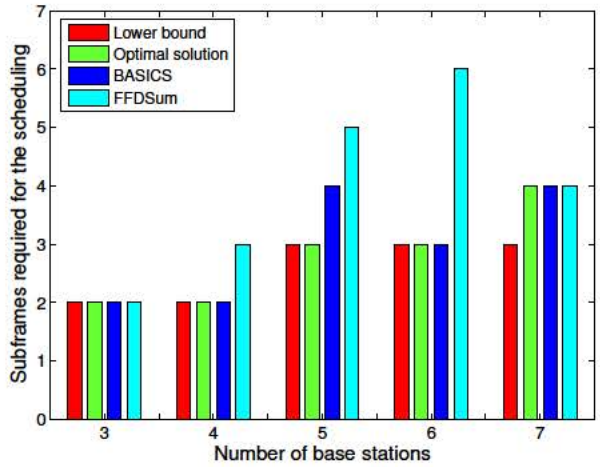


Figure 3.4: Number of subframes used with BASICS and with the optimal base station scheduling (obtained via brute force search).

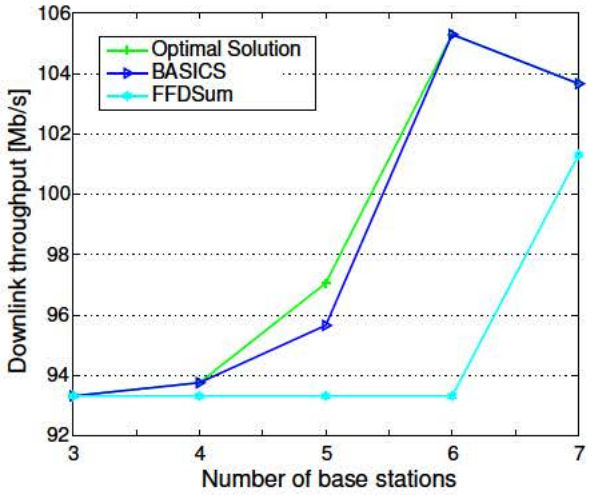


Figure 3.5: Throughputs achieved with BASICS and with the optimal base station scheduling (obtained via brute force search).

(referred to as "Legacy" in the figures), and (*ii*) a frequency reuse 3 scheme that partitions the network into three parts. For all cases, two intra base station schedulers are considered: round robin and proportional fair scheduling (for clarity of presentation, for BASICS and frequency reuse 3, we only show results achieved with the proportional fair scheduler, which are slightly better than with the round robin scheduler). We measure network performance in terms of the sum of the logarithms of the throughputs, as this is a well accepted metric to compare different scheduling mechanisms in terms of efficiency as well as fairness. Note that for the case of frequency reuse 3, we normalize the throughput to the number of carriers utilized, i.e., 3.

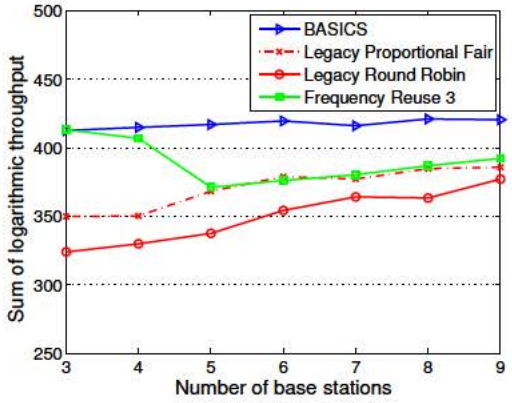


Figure 3.6: Performance comparison with a fixed number of users (150 users).

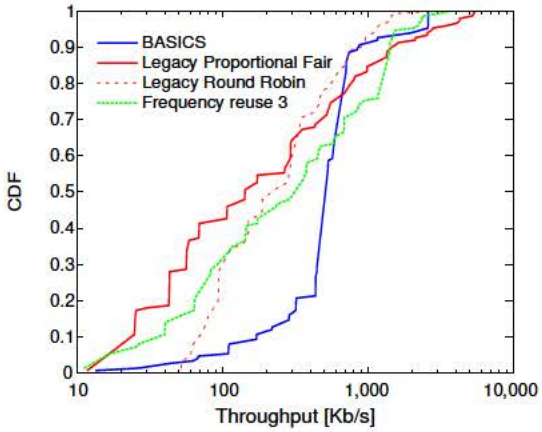


Figure 3.7: CDF of per-user throughput with 5 base stations and 150 users.

Fig. 3.6 shows the performance of the above approaches when we fix the total number of users in the system to 150, and vary the number of base stations from 3 to 9. We observe from the figure that BASICS significantly improves network performance with respect to both legacy approaches and frequency reuse 3. Results achieved with frequency reuse 3 are similar to the ones achieved with BASICS only for scenarios with very few base stations.

In order to gain additional insights on the actual distribution of throughputs, Fig. 3.7 shows the CDF of user throughputs for the specific case of a network with 5 base stations and 150 users. Notably, with BASICS, the majority of users receive a throughput in the range 600 to 700 Kb/s, while other schemes yield a throughput distribution spread over large intervals (from few Kb/s to about 2 Mb/s). This translates into improved fairness levels when BASICS is adopted. Therefore, this shows that BASICS not only ameliorates the throughput, but also enhances fairness.

Next, we evaluate the impact of the number of users. To this end, we consider a network in which the number of users is proportional to the number of base stations. Specifically, we simulate

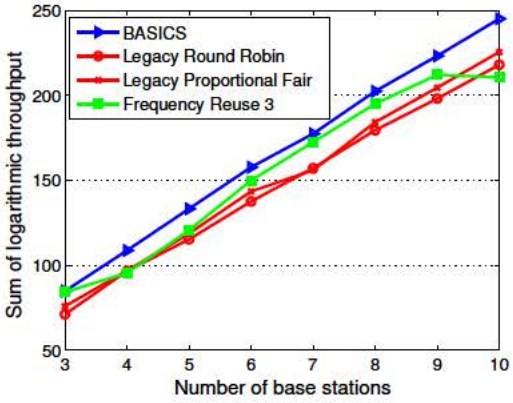


Figure 3.8: Performance comparison with fixed number of users per base station (8 users per base station).

3 to 10 base stations with 8 users each. Fig. 3.8 depicts the sum of logarithmic throughputs as a function of the number of base stations. Also for this case, BASICS exhibits the best performance over all the other approaches. On the one hand, the gain of BASICS over the legacy schemes is of several logarithmic units (and hence substantial in a linear scale). On the other hand, the gain over frequency reuse 3 is lower until the number of base stations reaches 10 (which is explained by the fact that frequency reuse becomes less effective as the network density grows). Taking into account that frequency reuse requires multiple carriers to achieve worse results, we conclude from these results that BASICS provides substantial improvements in performance also for this case.

3.4.5. Impact of LTE implementation details

In the previous subsections, we have evaluated the performance of BASICS based on our MATLAB tool, which does not take into account the impact of network protocols. In the following, we evaluate the performance of our proposal, of legacy schedulers, and of frequency reuse scheme in a more realistic scenario by using packet level simulations with the OPNET simulator, on top of which we implemented BASICS.

To assess the impact of the network protocols, we repeat the experiment of Fig. 3.8 (fixed number of users per base station) with OPNET. We limit the number of base stations to 5 due to the computational constraints of the packet level simulator. Fig. 3.9 depicts the sum of logarithmic throughputs achieved when each base station has 8 users. Results exhibit the same trend as Fig. 3.8, where BASICS outperforms all other approaches. A close look at the figures confirms that MATLAB results are very close to the OPNET ones.

Based on the above results, we draw the following two conclusions: (i) BASICS substantially boosts network performance in terms of throughput and fairness; and (ii) the gain provided by BASICS is not affected by transport-, network- and MAC-layer implementation details, which

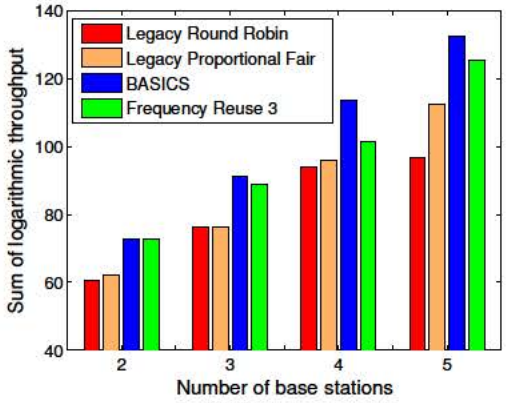


Figure 3.9: Performance evaluation of BASICS (OPNET simulator).

corroborates the MATLAB results presented throughout the chapter.

3.5. Conclusions

In this chapter, we have presented BASICS, a scheme to coordinate neighboring base stations to minimize inter-cell interference while achieving high spectral efficiency. In contrast to previous works available in the literature, the proposed approach leverages *base station* scheduling rather than *user* scheduling, which makes BASICS suitable to deploy the recently standardized ABSF mechanism. We have also formulated the base station scheduling problem and proved that it is equivalent to a multi-dimensional vector bin-packing problem, which is NP-Hard. We have then proposed a heuristic and have shown that it achieves near-optimal performance while scaling with the number of users. This chapter has revealed that mapping base station activities over subframes yields significant gains in terms of spectral efficiency and also results in a good level of fairness in the distribution of throughput among users. Additionally, BASICS incurs a very low signaling overhead and it does not require changes in the per-user scheduling policies implemented by the base stations.

Chapter 4

Lightweight Distributed eICIC Mechanism

While a centralized approach could significantly improve the network efficiency, this dramatically burdens on the complexity. Specifically, when guaranteed traffic is also considered, we need to boil down the computational effort and resort to a distributed solution. Such distributed approach with local decisions would not only be aligned with the well accepted self-organizing network concepts, but also allow to make ABSF and user scheduling decisions jointly—rather than assuming worst case conditions for the user scheduling process—which allows for further improving performance. Note that one critical aspect in the design of the distributed scheme is to limit the amount of information exchanged between base stations as well. To achieve this goal we present in this chapter a Distributed Multitraffic Scheduling (DMS), a scheme providing a lightweight ABSF coordination of *local schedulers* (base stations) with the help of a *central coordinator*, which supervises ABSF decisions of the base stations and drives the system to the best possible performance without imposing centralized decisions on ABSF patterns. Hence, our proposed solution relies on a semi-distributed approach that offloads and reduces the computational burden from a centralized controller while drastically abating the signaling overhead. This makes our approach a first step towards a practical and effective solution to ABSF that can be implemented in real networks.
We validate the proposed scheme via simulation and show that, despite its low complexity and the very limited amount of control messages required, DSM achieve near-optimal performance in terms of:

- maximizing radio resources reuse;
- providing sufficient quality for guaranteed traffic;
- minimizing the time used to guaranteed traffic to leave room for best-effort traffic;
- maximizing resource utilization efficiency for best-effort traffic.

DSM also exhibits significant advantages over existing schemes in terms of efficiency, complexity, fairness, and throughput. In addition, valuable comparisons with existing power control schemes reveal that complex approaches like [97] bring little additional gain with respect to DMS and behave less fairly, whereas low-complexity solutions like [86] exhibit lower efficiency with respect to our proposal.

4.1. Centralized problem formulations

The goal of ICIC solutions is to improve system spectral efficiency by significantly limiting base station mutual interference. To this end, our mechanism optimally orchestrates base station activities and performs user scheduling on a time-slot basis, i.e., per Transmission Time Interval (TTI). Here, we cast the ICIC problem into a 5G network, where the ABSF standard technique is implemented. Therefore, our solution instructs each base station to use a given *ABSF time-pattern*, which is a bitmap that specifies which TTIs must be blanked. For the sake of simplicity, problem formulations presented in this section consider downlink traffic only. However, an extended model could be straightforwardly derived for uplink transmissions. Additionally, our network model addresses two distinct traffic classes: (*i*) GBR traffic and (*ii*) best-effort traffic. While the former is subject to a strict rate constraint and it is accommodated with higher priority, the latter (i.e., traffic for which there are no stringent requirements in terms of latency and bandwidth) represents the most common traffic type in mobile data networks and it is gradually served with the remaining resources. Specifically, the above problem can be reformulated as finding a global time-allocation strategy for different base stations to accommodate, first, GBR traffic demands into a minimum number of TTIs, and, in the TTIs left, to serve best-effort traffic by maximizing the network spectral efficiency and guaranteeing a good level of fairness. A fundamental aspect for GBR traffic is represented by the amount of data to be transmitted within a fixed and periodic *time horizon*. In our problem formulation, the considered time horizon corresponds to the ABSF pattern, which is used to transmit both inelastic and best-effort traffic. Therefore, the final objective of the optimization strategy is to squeeze the *GBR traffic period* (i.e., TTIs used for GBR traffic) as much as possible, so as to maximize the time left for best-effort traffic demands.

In what follows, we first formulate the ICIC problem from a *centralized* scheduling perspective for both traffic types, partially recalling what has been presented in the previous chapter. We show that an optimal centralized approach, although being practically unfeasible due to computational and signaling overhead, provides us with a benchmark corresponding to the best possible performance of any implementable algorithm.

4.1.1. Optimizing GBR Traffic Period

We formalize the problem of optimizing the GBR traffic period length as follows. Let us assume that each base station $i \in \mathcal{N} = \{1, 2, \ldots, N\}$ has a fixed set of users, denoted as $\mathcal{U}_i = \{1, 2, \ldots, U_i\}$. Let the GBR traffic demand of each user be known at the base station

side, expressed in volume of traffic to be periodically served, and denoted as $D_u, u \in \mathcal{U}_i$. Let W denote the available *time horizon* (in TTIs), i.e., the length of ABSF patterns, which means that the user demand guaranteed rate is $D_u/(W \cdot T_{slot})$ bps, where T_{slot} is the duration of a TTI. Let us further assume that a base station can schedule at most one user in each TTI, and some TTIs can be *blanked* by means of the ABSF pattern.

The *GBR traffic period* devoted to serve the GBR traffic demands will be no than a given portion of the W TTIs; without loss of generality, let us assume that this period is a set of consecutive TTIs, $\mathcal{T} = \{1, 2, \ldots, T\}$. If no a-priori bound on the GBR traffic is needed, then $T = W$. The objective of the optimization problem is to allocated user demands in the smallest possible number of TTIs possible, $L < T$, satisfying channel quality constraints. This allows best-effort traffic to be scheduled both in the remaining $W - T$ TTIs and in those unused in the GBR traffic portion, $T - L$. Each TTI not used by a BS within the L TTIs is seen by that BS as *blanked time slot*, using the ABSF terminology.[1]

Since the system has limited capacity, the above problem may not be feasible as it may not be possible to allocate the entire user demand set within $\mathcal{T}$ assigned slots. In order to ensure that the problem is always mathematically feasible, we define a per-user *penalty* p_u, which represents the unserved demand. As long as no penalty is accumulated, the solution tries to minimize the GBR traffic period l, leaving more room (e.g., $W - l$ TTIs) for best-effort traffic. We formulate the optimization problem for the GBR traffic period as follows

Problem `GBR`:

$$
\begin{aligned}
\text{minimize} \quad & L + \alpha \sum_{u \in \mathcal{U}} p_u, \\
\text{subject to} \quad & ts_t \leq L, \forall t \in \mathcal{T}; \\
& \sum_{i \in \mathcal{N}} y_{i,t} \leq s_t, \forall t \in \mathcal{T}; \\
& \sum_{u \in \mathcal{U}, m \in \mathcal{M}} x^m_{u,t} \leq y_{i,t}, \forall i \in \mathcal{N}, t \in \mathcal{T}; \\
& \sum_{u \in \mathcal{U}, m \in \mathcal{M}} x^m_{u,t} \leq 1, \forall t \in \mathcal{T}; \\
& \frac{P G_{u,i}}{N_0 + \sum\limits_{k \in \mathcal{N}: k \neq i} P G_{u,k} \cdot y_{k,t}} \geq \gamma^m \cdot x^m_{u,t}, \\
& \qquad \forall i \in \mathcal{N}, u \in \mathcal{U}, t \in \mathcal{T}, m \in \mathcal{M}; \\
& \sum_{m \in \mathcal{M}, t \in \mathcal{T}} R^m x^m_{u,t} + p_u \geq D_u, \forall u \in \mathcal{U}; \\
& s_t, x^m_{u,t} \in \{0; 1\}, \forall u \in \mathcal{U}, m \in \mathcal{M}, t \in \mathcal{T}; \\
& y_{i,t} \in \{0; 1\}, \forall i \in \mathcal{N}, t \in \mathcal{T}; \\
& p_u, t \geq 0,
\end{aligned}
$$

where $\alpha > 0$ measures the relative importance of penalties over utilized TTIs,[2]. Variables s_t are

[1] In a blanked slot only control information (e.g., pilots) is transmitted, and no user data.

[2] To give priority to GBR traffic requirements over efficiency in the cost function of Problem `GBR`, α needs to be designed in order to have $\alpha \sum_{u \in \mathcal{U}_i} p_u \gg T$ as soon as any of the p_u values is non-zero. To achieve this, it is sufficient to set $\alpha \sim 1000$, as long as we have a resolution of 1 bit per time horizon in p_u and time horizons in the order of a few

binary variables indicating with value 1 whether TTI t is used for transmissions, l is a positive real variable storing the highest index of used TTIs within $\mathcal{T}$, $y_{i,t}$ are binary variables taking 1 whether BS i is scheduled in TTI t, 0 otherwise, and $x^m_{u,t}$ are binary variable that take the value of 1 if user u is scheduled into TTI t with modulation and coding scheme (MCS) $m \in \mathcal{M}$, in which case it receives a rate $R^m \in \mathcal{R}$, in bits per-TTI.

The first set of constraints force the correct value to be assigned to l. The second and third sets of constraints impose the coherence between, respectively, (i) active BSs and used TTIs, (ii) scheduled users and active BSs. The fourth set of constraints impose that at most one user may be scheduled in each TTI. In the signal-to-interference-plus-noise ratio (SINR) expression, $G_{u,k}$ is the channel gain between user u and the base station k, P is the base station transmission power,[3] and N_0 is the background noise. The use of transmission rate R^m is subject to the availability of a SINR value greater than the corresponding threshold γ_m. Although the SINR constraints are not linear, they can be easily linearized. The main assumption behind the centralized model is that users' CSI is perfectly known. Such information is gathered and updated by a centralized controller, which uses it to compute SINR constraints. The last constraint is used to set penalty values in order to compensate the unserved traffic demands. With this, the Problem `GBR` is a Mixed-Integer Linear Programming (MILP) and can be solved with state-of-the-art solvers.

4.1.2. Optimizing Best-effort Traffic Period

Once a feasible GBR traffic period L is found, the remaining ABSF pattern TTIs Z (e.g., $Z = W - L$) will be used for accommodating best-effort traffic demands. Differently from the GBR case, here the goal is to obtain a user scheduling and BS activation that can efficiently exploit the remaining network resources, aim both at spectral efficiency and at user fairness. We can formulate the centralized optimization problem with an Integer Linear Programming (ILP) model. The objective function to be maximized, $\widehat{\eta}$, is the sum of the utilities of the individual base stations. Following the widely accepted *max-min* fairness criterion, we define the utility of base station i as the minimum rate of all the users in the base station [4] and formalize as follows:
Problem `BE`:

$$
\begin{aligned}
\text{maximize} \quad & \widehat{\eta} = \sum_{i \in \mathcal{N}} \left(\min_{(u,t) \in \mathcal{U}_i \times \mathcal{Z}} R^m \cdot x_u^{m,t} \right), \\
\text{subject to} \quad & \sum_{u \in \mathcal{U}_i, m \in \mathcal{M}} x_u^{m,t} \leq y_{i,t}, \quad \forall i \in \mathcal{N}, t \in \mathcal{Z}, \\
& \frac{P G_{u,i}}{N_0 + \sum_{k \in \mathcal{N}: k \neq i} P G_{u,k} \cdot y_{k,t}} \geq \gamma^m \cdot x_u^{m,t}, \\
& \qquad \forall i \in \mathcal{N}, u \in \mathcal{U}_i, m \in \mathcal{M}, t \in \mathcal{Z}, \\
& y_{i,t}, x_u^{m,t} \in \{0;1\}, \forall i \in \mathcal{N}, u \in \mathcal{U}_i, m \in \mathcal{M}, t \in \mathcal{Z};
\end{aligned}
$$

hundreds of TTIs, which is a reasonable length under dynamic traffic conditions.

[3] Following current cellular deployments, we assume that base stations transmit at a constant power.

[4] Note that the selected objective function provides a trade-off between maximizing the spectral efficiency and guaranteeing a minimum level of service quality, as pointed out, e.g., in [2]. Nevertheless, different objective functions can be considered as well, without substantially changing the proposed approach and the following analysis.

where variables and parameters are defined exactly as in Problem GBR. The two sets of constraints correspond to the third and fourth ones in Problem GBR. Problem BE can be reduced to a bin-packing problem in which the sum of interferences cannot exceed a threshold. Therefore, this problem is NP-hard [37].

As initially stated, the centralized solution of Problem GBR and Problem BE involves a very high overhead to deliver CSI information to the centralized controller, which needs this information to select the ABSF patterns and compute the user scheduling. In addition, due to problem complexity, while the centralized approach can be an attractive option for small networks, a less complex and more distributed approach is required to deal with the case of very dense wireless networks consisting of hundreds of base stations and thousands of wireless nodes. In the following section we present a distributed approach to this joint problem in order to abate and distribute the computational load over the base stations. We first analyze the two problem individually, starting from Problem GBR, then we propose a joint framework for both problems.

4.2. Guaranteed Traffic requests

As explained in the previous section, an optimal centralized approach to solve the interference coordination problem for all base stations would be unfeasible due to the large number of exchanged messages to collect user channel state information from all users to a centralized controller. We therefore formulate the problem in a distributed way by splitting it into local problems that are solved by each base station. The local optimization problem consists in minimizing a cost function f_i, which accounts for both the number of locally used TTIs and the total penalty related to unsatisfied local demands.

The distributed approach distributes the computational burden of the original problem over the base stations present in the network. Specifically, to reduce complexity, in the distributed problem each base station only optimizes the scheduling of its own users and considers that other base stations use fixed ABSF patterns. However, this approach needs an iterative mechanism to find the optimal ABSF pattern of all base stations. Note that, with the distributed approach, *the complexity of the problem to solve is dramatically reduced, while the number of iterations required to converge will be shown to grow at most quadratically with the network size*.

In order to design a distributed version of Problem GBR, we formulate a local problem with a modified objective function such that, once included in the framework described in Section 4.4, provides a final solution quality close to the one directly computed by solving Problem GBR in a centralized way. We replace the objective function, which includes the global variable L, with a local function that aims at minimizing the total number of transmissions to local users. The minimization of the number of occupied TTIs is iteratively obtained by trying to reduce the cardinality of set $\mathcal{T}$ considered in the local problem. Numerical results in Section 4.5.1 show this approach yields indeed a very good performance.

The local problem of the distributed version can be formulated as follows:

Problem GBR-DISTR:

$$
\begin{aligned}
\text{minimize} \quad & f_i = \sum_{u \in \mathcal{U}_i, t \in \mathcal{T}} x^m_{u,t} + \alpha \sum_{u \in \mathcal{U}_i} p_u, \\
\text{subject to} \quad & \sum_{t \in \mathcal{T}, m \in \mathcal{M}} R^m \cdot x^m_{u,t} + p_u \geq D_u, \quad \forall u \in \mathcal{U}_i; \\
& \sum_{u \in \mathcal{U}_i, m \in \mathcal{M}} x^m_{u,t} \leq 1, \quad \forall t \in \mathcal{T}; \\
& \frac{PG_{u,i}}{N_0 + \sum_{k \in \mathcal{N} \setminus i} PG_{u,k} \cdot S^k_t} \geq \gamma^m \cdot x^m_{u,t}, \\
& \qquad \forall u \in \mathcal{U}_i, t \in \mathcal{T}, m \in \mathcal{M}; \\
& x^m_{u,t} \in \{0; 1\}, \quad \forall u \in \mathcal{U}_i, t \in \mathcal{T}, m \in \mathcal{M}; \\
& p_u \geq 0, \qquad \forall u \in \mathcal{U}_i.
\end{aligned}
$$

Variables and constraints are the same as in the centralized formulation, except for two aspects: (i) the local formulation considers only the users of the local base station, (ii) the activity of interfering stations in the SINR constraint is no longer optimized, but given as input. The key idea behind the distributed approach indeed affects only this constraint. Since just the base station activity is enough to compute SINR values, the exact knowledge of which users are scheduled by other base stations is not needed to solve the local problem, thus, it is sufficient to know the activity patterns of neighboring base stations provided by binary vectors $\{S^k_t\}$. Those are ABSF patterns (actually, the inverse of them) collected by other base stations. Base station b will deliver in turn its ABSF pattern, computed as $\{S^b_t = 1$ if $\sum_{u \in \mathcal{U}_b, m \in \mathcal{M}} x^m_{u,t} \geq 1, 0$ otherwise, $\forall t \in \mathcal{T}\}$. Note that channel gain values $G_{u,k}$ to neighboring base stations can be easily derived by RSS values commonly collected by UEs.

As stated above, each base station i is in charge of solving Problem GBR-DISTR, by computing the optimal user scheduling into available TTIs. Note that the solution of this problem depends on the solutions computed by the other base stations, since the SINR of each user is given by the interference generated by the other base stations in the system when they are active. Therefore, in the distributed approach formulation, each base station simply schedules local users in order to maximize the objective function defined in Problem GBR-DISTR. However, the schedule defines the activity of the base station, and the interference generated towards other base stations, which, in turn, can react readjusting their scheduling in order to adapt to changed interference conditions. A new scheduling may cause new interference levels, therefore each base station must iteratively solve Problem GBR-DISTR, until the system converges to a stable solution.

Ideally, the iterative process will converge to a quasi-optimal solution where base stations agree on their respective ABSF patterns. However, the process could not converge at all. In the next sections we derive convergence properties and provide conditions on the guaranteed convergence by casting the distributed approach into a game where base stations act as players in order to maximize the local objective function. Once the convergence is guaranteed, we finally present a practical distributed scheme that implements the distributed approach.

4.2.1. Game theoretical analysis

We introduce a new class of games, called *Distributed Inelastic Games*, to model the interference coordination problem. We define our game Γ and is represented by a tuple $\Gamma = (\mathcal{N}, (\mathcal{R}_i)_{i\in\mathcal{N}}, (f_i)_{i\in\mathcal{N}})$, where $\mathcal{N} = \{1, \ldots, N\}$ is the set of players. For each player $i \in \mathcal{N}$, $\mathcal{R}_i$ is a family of user strategies and f_i is a cost function that expresses the cost associated to the implementation of each strategy. The game consists of N base stations acting as players, where each player i plays her move in order to minimize the game cost function f_i (the terms "player" and "base station" are used interchangeably in the text). In particular, a valid move corresponds to choosing a strategy that satisfies that satisfies constraints, i.e., a user scheduling, in Problem `GBR-DISTR`. Let us consider the family $\mathcal{R}_i \in 2^{U_i \times T}$ of all possible moves for player i, where U_i is the set of users associated to base station i and T is the total number of available TTIs. Then, the set of valid moves for player i is given by $\mathbb{S}_i = \{S \subset \mathcal{R}_i : (u_i, t_i) \in S, (u_j, t_j) \in S \rightarrow t_i \neq t_j\}$. We also define the cardinality of a strategy S, namely $|S|$, as the number of (u, t) pairs selected in the strategy. Finally, note that the cost of each strategy depends on the other players' moves, as their activities may create interference towards a specific user u for a particular TTI t at base station i.

Given the above definitions, the Best Response (BR) for game Γ is defined as the strategy that produces the smallest cost function for player i, taking the other players' strategies as given. Analytically, $S_i^* \in \mathbb{S}_i$ is defined as BR if and only if

$$f(S_i^*, S_{-i}) \leq f(S_i, S_{-i}),\ \forall S_i \in \mathbb{S}_i. \tag{4.1}$$

4.2.1.1. Convergence properties and guarantees

In the following, we present a convergence analysis of game Γ, which is essential to ensure the feasibility and implementability of the distributed version. Indeed, due to the nature of the game, the arbitrary best responses taken by each player may not necessarily lead to an equilibrium (i.e., a Nash equilibrium); this is the case of game Γ and it is shown in the proof of Theorem 4.2.1, where players following the BR strategy do not reach an equilibrium.

Theorem 4.2.1. *Distributed Inelastic Game Γ does not possess a finite improvement property in best-response improvement dynamics.*

Proof: Consider a scenario with $T = 2$ TTIs and $N = 3$ base stations, each of them associated with $|U_i| = 1$ user. For each player i, the strategy space $\mathbb{S}_i$ is defined as

$$\mathbb{S}_1 = \{\{u_1, t_1\}; \{u_1, t_2\}; \{(u_1, t_1), (u_1, t_2)\}\},$$
$$\mathbb{S}_2 = \{\{u_2, t_1\}; \{u_2, t_2\}; \{(u_2, t_1), (u_2, t_2)\}\},$$
$$\mathbb{S}_3 = \{\{u_3, t_1\}; \{u_3, t_2\}; \{(u_3, t_1), (u_3, t_2)\}\}.$$

Let assume a traffic demand $D_u = 5$ bits and a user gain $c_{u,t}(S_{-i})$, in bit/TTI, according to the following table:

	$S_1^* = \{u_1, t\}$	$S_2^* = \{u_2, t\}$	$S_3^* = \{u_3, t\}$
$c_{u_1,t}$	(5.55)	5.11	**2.73**
$c_{u_2,t}$	**2.73**	(5.55)	5.11
$c_{u_3,t}$	5.11	**2.73**	(5.55)

Now we consider the sequence of strategies taken by each player, described by Table 4.3. Whenever a player i chooses a new strategy at step k, in order to minimize the cost function (bold-marked), the cost function assigned to the other players may increase. This leads to a cycle of equal strategies, such as strategies at step k and strategies at step $k + 6$. Hence, players playing arbitrary best responses do not necessarily converge to a Nash equilibrium in distributed inelastic games. ■

Table 4.1: Dynamics of game states for a Distributed Inelastic Game Γ by adopting Best Response (BR)

$S_i^{*(s)}$, $f(S_i^*, S_{-i})$	$s = k-1$	$s = k$	$s = k+1$	$s = k+2$
BS 1	$\{u_1, t_1\}, 1$	$\{u_1, t_1\}$,*(228)*	$\boldsymbol{\{u_1, t_2\}, 1}$	$\{u_1, t_2\}, 1$
BS 2	$\{u_2, t_2\}, 2$	$\{u_2, t_2\}, 1$	$\{u_2, t_2\}$,*(228)*	$\boldsymbol{\{u_2, t_1\}, 1}$
BS 3	-	$\boldsymbol{\{u_3, t_1\}, 1}$	$\{u_3, t_1\}, 1$	$\{u_3, t_1\}$,*(228)*
$S_i^{*(s)}$, $f(S_i^*, S_{-i})$	$s = k+3$	$s = k+4$	$s = k+5$	$s = k+6$
BS 1	$\{u_1, t_2\}$,*(228)*	$\boldsymbol{\{u_1, t_1\}, 1}$	$\{u_1, t_1\}, 1$	$\{u_1, t_1\}$,*(228)*
BS 2	$\{u_2, t_1\}, 1$	$\{u_2, t_1\}$,*(228)*	$\boldsymbol{\{u_2, t_2\}, 1}$	$\{u_2, t_2\}, 1$
BS 3	$\boldsymbol{\{u_3, t_2\}, 1}$	$\{u_3, t_2\}, 1$	$\{u_3, t_2\}$,*(228)*	$\boldsymbol{\{u_3, t_1\}, 1}$

While the symmetric scenario considered in the above proof is quite unlikely in realistic LTE-Advanced environments, the theorem does nonetheless point out that the game Γ may not converge in some critical scenarios, which may create applicability problem if not fixed.

Although, as stated by Theorem 4.2.1, players iteratively playing their best response do not necessarily reach a Nash equilibrium, we are able to prove that, using a particular class of best responses, they eventually reach the equilibrium. This particular best-response set consists in selecting among all possible best responses only those which just add or remove at most one (u, t), (user,TTI) pair, to the strategy of the previous step. We call such set *Single-step* set, $\mathbb{S}_i^{SS}$, and define it formally as follows. Starting from any strategy $S_i^{(p)}$ taken at the previous step p, $\mathbb{S}_i^{SS}(S_i^{(p)}) = \{S \in \mathbb{S}_i : (|S \setminus S_i^{(p)}| \leq 1) \ \vee \ (|S_i^{(p)} \setminus S| \leq 1)\}$, where the $\vee$ symbol is the OR operator. Now we can define a Single-step Best Response (SSBR) move:

Definition 1. *At step k, the Single-step best response $\hat{S}_i^k$ selected by player i is defined as a best response strategy $S_i^{*(k)}$ such that $S_i^{*(k)} \in \mathbb{S}_i^{SS}(S_i^{(k-1)})$.*

The cardinality of the set set $\mathbb{S}_i^{SS}(S_i^{(p)})$ is exactly equal to

$$|\mathbb{S}_i^{SS}(S_i^{(p)})| = 1 + |S_i^{(p)}| + [T - |S_i^{(p)}|] \cdot |U_i|. \tag{4.2}$$

The above definition states that player i will play her single-step best response by taking into consideration her strategy played at the previous step and (i) removing one of the (user,TTI) pair, (ii) adding just one additional (user,TTI) pair, or (iii) following the previous strategy (if the cost function is minimized for that particular strategy).

In order to prove that the convergence is guaranteed by following the SSBR approach, we next introduce the concept of strategy profile. Given a state of the game Γ at a particular round, the strategy profile σ is the set of strategies played by each player in that round. When a player changes her strategy, the strategy profile is updated. In the following, we define a particular strategy profile, which we call *saturation strategy profile*.

Definition 2. *The saturation strategy profile is defined as a strategy profile $\overline{\sigma} = [S_1, ..., S_N]$ belonging to a set of saturation strategy profiles Σ^{SAT}, $\overline{\sigma} \in \Sigma^{SAT}$, where each player's strategy S_i either returns a cost function with a zero penalty or occupies all available T TTIs with a non-zero penalty.*

Assuming that the players play their SSBR strategy for a game Γ, we are ready to prove the following theorem: *Game Γ has a Nash Equilibrium, which can be reached by playing SSBR moves*. In order to do that we rely on the following Lemmas, whose formal proofs are reported in the Appendix.

Lemma 4.2.1. *Given that the players' strategies belong to whatever strategy profile σ, after a finite number of single-step best responses (SSBR), all players' strategies will belong to a saturation strategy profile $\overline{\sigma}$.*

Lemma 4.2.2. *At a certain point in time, given that the strategies played by any player in the system belong to a saturation strategy profile $\overline{\sigma}$ if each player chooses a single-step best response (SSBR), the game will converge to a Nash equilibrium.*

Theorem 4.2.2. *Game Γ possesses at least one Nash equilibrium and players reach an equilibrium after a finite number SSBR moves.*

Proof: We prove it by constructive proof. Players start playing a game Γ. Regardless of the starting strategy profile σ, after playing a finite number of SSBR, the players' strategies belong to a saturation strategy profile $\overline{\sigma}$, as stated by Lemma 4.2.1. Upon all players play a strategy belonging to a saturation strategy profile, keeping choosing a SSBR strategy, they will converge in a finite number of steps to a Nash equilibrium according to Lemma 4.2.2. Therefore, we can state that each game Γ admits a Nash equilibrium, and players can reach such equilibrium. ■

The proof of the theorem is also confirmed by readily applying the SSBR to the scenario presented in Theorem 4.2.1, where players keep playing their BR without never reaching a Nash Equilibrium. In that example, choosing the SSBR for all players leads to fully schedule all available TTIs for every base station. Interestingly enough, Theorem 4.2.2 also proves that players can easily adopt a general best response strategy S_i^* during the game, with no convergence guarantees. However, if at a certain point in time, they switch to SSBR, they converge to a Nash

equilibrium with probability equal to 1. Clearly, if at least one player is not playing SSBR, the game convergence is no longer guaranteed.

4.3. Best-Effort Traffic requests

We next present a distributed formulation of Problem BE, which aims at reducing of the peak computational burden at a centralized controller by distributing the load among BSs. To formulate the distributed approach, the original problem is split into several smaller instances, which are solved locally by each base station. To solve a problem instance, the base station is provided with the activity pattern declared by other base stations. This is given by *ABSF patterns*, $S_{i,t}$, which are exchanged among base stations ($S_{i,t} = 0$ if base station i blanks TTI t). With the above information, and without explicitly forcing any additional constraint, each base station i would schedule users selfishly in the entire set of T TTIs, in order to optimize the local utility. Therefore, to avoid that base stations use all available TTIs, in the distributed problem formulation, we grant a single base station i access to up to M_i TTIs over T available TTIs; such M_i value plays a key role in the distributed mechanism, as it will be clarified in Section 4.4.2.

The above description corresponds to the following instance of the local problem for base station i, which can be formulated as an ILP model as follows:

Problem BE-DISTR:

$$
\begin{aligned}
\text{maximize} \quad & \widehat{\eta_i} = \min_{(u,t)\in\mathcal{U}_i\times\mathcal{T}} R^r \cdot x_u^{r,t} \cdot a_{u,i}, \\
\text{s.t.} \quad & \sum_{u,r} x_u^{r,t} \cdot a_{u,i} \leq 1, \qquad \forall t \in \mathcal{T}, \\
& \frac{PG_{u,i}}{N_0 + \sum_{k\in\mathcal{N}:k\neq i} PG_{u,k}\cdot ABSF_{k,t}} \geq \gamma^r \cdot x_u^{r,t}, \\
& \qquad \forall u \in \mathcal{U}_i, r \in \mathcal{R}, t \in \mathcal{T}, \\
& \sum_{u,t,r} x_u^{r,t} \leq M_i, \\
& x_u^{r,t} \in \{0;1\}, \quad \forall u \in \mathcal{U}_i, r \in \mathcal{R}, t \in \mathcal{T};
\end{aligned}
$$

where all parameters and constraints have the same meaning as in Problem BE, except for the third constraint, which limits the number of usable TTIs to M_i. Note that a feasible solution of Problem BE-DISTR can be computed by using any available max-min scheduling heuristic (see, e.g., [16]).

Following the local optimization problem presented above, the interference coordination problem is solved in a distributed fashion: each base station receives as input the ABSF patterns (i.e., $\{S_t^k\}$ bitmaps), solves Problem (4.3) and provides in turn to other base stations its ABSF pattern. Other base stations update their choices depending on the new ABSF pattern, communicate back their new ABSF decisions and the process repeats. However, as happened with Problem GBR-DISTR, this process may not converge. We investigate this issue relying on game theoretical tools and the notion of equilibrium, following the approach adopted in the GBR

analysis.

4.3.1. Game theoretical analysis

In the following, we analyze the (fully) *distributed* approach formulated above from a game theoretic standpoint and show that its *convergence is not guaranteed*. Building on this result, later in Section 4.4.2 we propose a *semi-distributed* approach that guarantees the convergence of the game.

Based on game theory, the distributed approach can be modeled as a game where base stations iteratively play in order to maximize their utility. Let us define this game as an *Interference Coordination Game* Ω, where each base station i acts as a player. Similar to previous game Γ, the set of strategies of each player $\mathbb{S}_i$ consists in the set of pairs (user, TTI), $(u,t) : u \in \mathcal{U}_i, t \in \mathcal{T}$, available for each base station according to constraints in Problem `BE-DISTR`, however the two cost functions differ.

In order to analyze the convergence of the above game, we rely on the concept of *Bottleneck Matroid Congestion Game* (for a detailed discussion, we refer the reader to [41]). A *Bottleneck Congestion Game* is a class of games where resources are shared among players. The utility of each player depends on the utility of the resources she chooses and the number of players choosing the same resources: the higher the congestion, the lower the utility. In particular, the individual player utility is the minimum of the utilities of the resources chosen in her strategy. Note that the bottleneck property is related to the max-min objective function, which distinguishes Problem `BE-DISTR` from Problem `GBR-DISTR`.

In addition to the above, regular congestion games can be generalized in *player-specific congestion games* and *weighted congestion games*. In the former, every player has her own utility function for every resource. In a weighted congestion game, every player affects the other players strategies with a different weight, namely, she causes a different level of congestion. The following theorem shows that our game falls in the intersection between the above categories, and hence existing results on these classes of games can be applied to our problem.

Theorem 4.3.1. *The Interference Coordination Game Ω is a Weighted Player-specific Bottleneck Matroid Congestion Game.*

Proof: Here we provide the reader with a sketch of the proof. The Interference Coordination Game Ω is player-specific since utility is player-specific as it depends on received interference, and it is a congestion game in which congestion weights are given by the interference caused by the scheduled users in each TTI. Moreover, strategies' constraints induced by constraints in Problem `BE-DISTR` make the strategy space a matroid, thus Ω is a Matroid Congestion Game. ■

Regular bottleneck congestion games have been proven to satisfy the *finite improvement property*, which states that an arbitrary BR sequence played by each player during the game always converges to an equilibrium in a finite number of steps [41]. However, the generalizations of

Table 4.2: Example of weighted player-specific matroid bottleneck congestion game that does not converge

Rate	Alone	With BS 1	With BS 2	With BS 3
$c_{u_1,t}$	2.0	–	1.5	**1.1**
$c_{u_2,t}$	2.0	**1.1**	–	1.5
$c_{u_3,t}$	2.0	1.5	**1.1**	–

player-specificity and different congestion weights introduce many degrees of freedom, which weakens the game structure and its convergence guarantees. Indeed, the following theorem shows that Weighted Player-specific Bottleneck Matroid Congestion Games do not satisfy the finite improvement property.

Theorem 4.3.2. *Weighted player-specific matroid bottleneck congestion games do not exhibit the finite improvement property in best-response improvement dynamics.*

Proof: Let us consider a scenario with $T = 2$ TTIs and 3 base stations, each of them associated with $|\mathcal{U}_i| = 1$ distinct user. For each player i, the strategy space $\mathbb{S}_i$ is defined as $\mathbb{S}_i = \{\{(u_i, t_1)\}; \{(u_i, t_2)\}; \{(u_i, t_1), (u_i, t_2)\}\}$. Let us assume an upper bound on available TTIs per base station $M_i = 1, \forall i \in N$ and a user rate $c_{u,t}$, expressed as bits/symb/TTI, according to Table 4.2. Now we consider the sequence of strategies taken by each player, described by Table 4.3.

Table 4.3: State evolution for a weighted player-specific matroid bottleneck congestion game that does not converge (example used in the proof of Theorem 4.3.2)

$S_i^{*(s)}$, $f(S_i^*, S_{-i})$	$s = k-1$	$s = k$	$s = k+1$	$s = k+2$
BS 1	$\{u_1, t_1\}, 2$	$\{u_1, t_1\}$, *(1.1)*	$\boldsymbol{\{u_1, t_2\}, 1.5}$	$\{u_1, t_2\}, 2$
BS 2	$\{u_2, t_2\}, 2$	$\{u_2, t_2\}, 2$	$\{u_2, t_2\}$, *(1.1)*	$\boldsymbol{\{u_2, t_1\}, 1.5}$
BS 3	-	$\boldsymbol{\{u_3, t_1\}, 1.5}$	$\{u_3, t_1\}, 2$	$\{u_3, t_1\}$, *(1.1)*
$S_i^{*(s)}$, $f(S_i^*, S_{-i})$	$s = k+3$	$s = k+4$	$s = k+5$	$s = k+6$
BS 1	$\{u_1, t_2\}$, *(1.1)*	$\boldsymbol{\{u_1, t_1\}, 1.5}$	$\{u_1, t_1\}, 2$	$\{u_1, t_1\}$, *(1.1)*
BS 2	$\{u_2, t_1\}, 2$	$\{u_2, t_1\}$, *(1.1)*	$\boldsymbol{\{u_2, t_2\}, 1.5}$	$\{u_2, t_2\}, 2$
BS 3	$\boldsymbol{\{u_3, t_2\}, 1.5}$	$\{u_3, t_2\}, 2$	$\{u_3, t_2\}$, *(1.1)*	$\boldsymbol{\{u_3, t_1\}, 1.5}$

Whenever a player i chooses a new strategy at the k^{th} step in order to maximize the utility function (bold-marked), the value of utility function calculated by the other players may decrease and they may want to change their strategy. This leads to a loop where players sequentially return on the same strategies indefinitely, such as strategies at step k and strategies at step $k + 6$. Hence, players playing arbitrary best responses do not necessarily converge to a Nash equilibrium in Weighted Player-specific Bottleneck Matroid Congestion Games, and thus, a finite improvement property does not always exist. ■

The above theorems prove that the distributed approach may not converge.[5] Moreover, the solution of Problem BE−DISTR ensures a global fairness among BSs, only if M_i values are

[5]It is worthwhile noting that the somehow pathological scheduling behavior considered in theorem's proof does not commonly exhibit in networks; indeed, according to the simulations conducted for typical realistic scenarios, the

properly tuned. In order to address these shortcomings, in the next section we propose a semi-distributed two-level mechanism where a local controller drives the behavior of the distributed game.

4.4. Distributed Multi-traffic Scheduling Framework

In this section we provide complete details on the proposed framework for adaptive interference-aware scheduling in 5G networks. We name our scheme *Distributed Multi-traffic Scheduling (DMS)*. DMS is based on the game theoretical framework introduced in Section 4.2 and Section 4.3 incorporates heuristic approaches to jointly adapt the solutions of Problem GBR and Problem BE to traffic changes. To cope with traffic and network dynamics, a practical strategy consists in periodic scheduling decisions, taken once per time horizon W (e.g., every ABSF pattern). In turn, inelastic traffic demands and best-effort data traffic are promptly served within the ABSF pattern W. DMS first accommodates inelastic traffic, which exhibits very stringent requirements. The remaining time portion is left for best-effort traffic requests.

The first objective of DMS is to smartly optimize the inelastic traffic period scheduling in order to maximize the resource efficiency while leaving more space for best-effort traffic. To this aim, DMS includes a mechanism that adaptively seeks the minimal number of TTIs to include in the inelastic traffic period T, so as inelastic traffic is served with no penalties in the shortest possible time window. A fully centralized approach to achieve this goal would consist in optimizing the lengths of all ABSF patterns in one single step at a central controller, resulting in intractable computational effort. However, in the following we show that, within the game theoretical approach proposed, a lightweight local controller suffices to solve the inelastic period minimization problem by leveraging a simple dihcotomic search algorithm over several time horizons. Then, DMS fully exploits the remaining time portion (e.g., best-effort traffic period $W - T = Z$) to maximize the aggregate system throughput for serving best-effort traffic. This is automatically performed using a distributed mechanism without incurring in a heavy centralized channel statistics collection.

4.4.1. Inelastic traffic scheduling

For inelastic traffic demands, DMS focuses on two different objectives. While the first objective is to fully accommodate the guaranteed traffic demands into the available time horizon T (*Resource Allocation*), the second objective is to iteratively reduce the number of used TTIs in order to make an efficient use of the time resources (*Time Squeezing*).

- *Resources Allocation* is completely executed into base stations, each of which is in charge of jointly scheduling local users and making ABSF pattern decisions, which are

interference coordination game Ω reaches an equilibrium with very high probability. Nevertheless, we still need to design an algorithm whose convergence is guaranteed.

exchanged with the other base stations through the local controller (LC). Game Γ is used for the base stations to accomplish this task in a coordinated way.

- *Time Squeezing* is executed at the LC. The LC collects the traffic demand offered to the base stations and iteratively adjusts the length of the time period based on the ABSF patterns announced by the base stations at the end of game Γ, and on penalties they could have incurred.

Practically speaking, DMS operation starts when user traffic demand changes [6] because of users leaving and/or joining the cellular network, as illustrated in in Fig. 4.1. User traffic demands are expressed as the volume of bits to be guaranteed in a fixed time horizon. Initially, each base station provides the LC with its cumulative traffic demand. The LC selects the initial time period as the one that would guarantee traffic constraints without adopting any interference coordination mechanism (W). As a consequence, the computation of assigned time resources is initially overestimated. After that, DMS operation consists in the interaction between the two aforementioned processes: the Resources Allocation process and the Time Squeezing process.

Resources Allocation process: guaranteeing user demands. During Resource Allocation, the number of available TTIs T is fixed. Base stations cooperatively schedule their own users into available TTIs in order to satisfy their traffic demands. It is very important to note that the mechanism perfectly complies with the requirements of the Distributed Inelastic Game Γ presented in Section 4.2. In particular, each base station schedules its own users and communicates corresponding ABSF pattern to the other base stations. Each base station limits its activity and reduces the interference caused to the other base stations by reducing the number of occupied TTIs, as stated in Problem `GBR-DISTR`. The process ends when a steady-state is reached, which always occurs, as proved in Section 4.2. Eventually, a steady state ABSF pattern for each involved base station is notified to the LC. In addition, the Resource Allocation process may output a set of non-zero user penalties, e.g., due to a too large traffic to be accommodated in the time horizon or to critical interference conditions. This event is promptly handled by the Time Squeezing process described next.

Time Squeezing: adapt time period to demand. Time Squeezing is based on a binary search scheme, as illustrated in Fig. 4.1. The initial time period T is set equal to the ABSF pattern length W, chosen as the number of TTIs needed to guarantee the entire demand. Then, a binary search is used to adapt the inelastic time period. At each step of the search (e.g., at the beginning of each ABSF pattern W), a new value is chosen for the time period and it is applied by invoking the Resource Allocation process. The Resource Allocation process runs and returns ABSF patterns and penalties. If the sum of the obtained penalties is equal to zero, meaning that user traffic demands are completely satisfied and the time period may not fully utilized, Time Squeezing reduces the time period T for the next ABSF pattern. If penalties occur, the process increases the

[6] We suppose that inelastic traffic demands are negotiated on a long-time window.

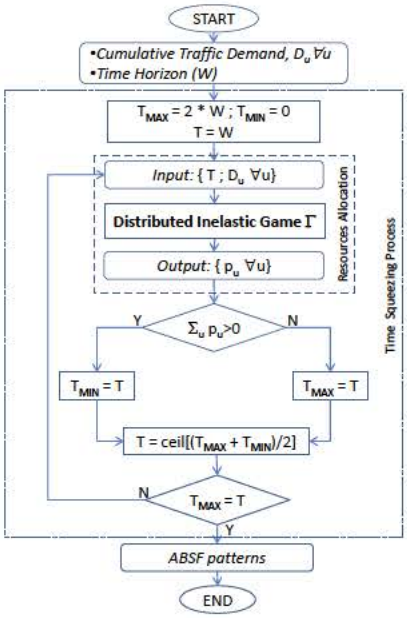


Figure 4.1: Two-level mechanism for guaranteed traffic. In the flow chart, the entire flow of the program is represented as an iteration between the execution of the Resources Allocation and of the Time Squeezing processes.

time period. The search ends if there are no penalties and no unused TTIs [7] by keeping using the same inelastic traffic period for next ABSF patterns, unless traffic demand changes.

Since during the Time Squeezing process a feasible value (no penalties) of the time period $\mathcal{T}$ is always available, the controller can command the base stations to apply the corresponding ABSF patterns and transmission scheduling without waiting for the convergence of the process. Hence, although with the first applied ABSF patterns resources are not used in a perfectly efficient way since T is larger than necessary, the system is always able to guarantee the rates of inelastic demand. However, as we will show in Section 4.5.1, in practical scenarios it takes only a few iteration for DMS to find efficient time periods.

4.4.2. Best-effort traffic scheduling

Building on the results of the previous section, DMS exploits the best-effort traffic period Z to maximize the overall system throughput for serving best-effort traffic requests. The DMS mechanism for best-effort traffic is depicted in Fig. 4.2. As shown, the scheme operates at two different timescales:

- On a *long-term timescale* (in the order of seconds), a local controller (LC), is in charge of adjusting the M_i value of each base station, where M_i gives the maximum number of TTIs that base station i can use to schedule its users within the time horizon T by solving

[7] In the worst case, the process stops after $\log_2 W$ steps, which is the number of steps need to explore the interval $[0, W]$ with a binary search

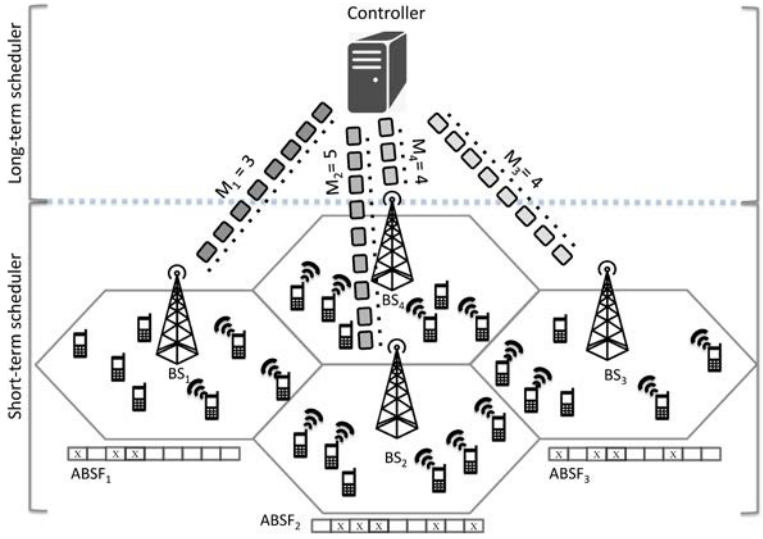


Figure 4.2: Hybrid two-level mechanism for best-effort traffic demands. In the short-term level (bottom side of the figure), game is played amongst the base stations, while in the long-term level (top side) the controller decides the number of available TTIs per base station.

Problem `BE-DISTR` (where the set $\mathcal{T}$ of available TTIs defined in Problem `BE-DISTR` corresponds to traffic period Z). In addition, adapting M_i is used to react to traffic changes in the system.

- At a *shorter timescale*, base stations play the Interference Coordination Game Ω by sequentially exchanging their scheduling decisions in terms of ABSF patterns.[8] As described in the following, the local controller does not directly participate in the game, but it controls its convergence by limiting the number of iterations.

The remaining challenge for coordinator-aided approach is the design of the algorithms executed by the local controller to (i) ensure convergence, and (ii) adjust the values M_i. In the following we address the design of those algorithms, which aim at driving the system behavior to an optimal state in the long run.

Convergence control of game Ω. In order to guarantee the convergence of the game, the central coordinator imposes a deadline of $\hat{Z}$ TTIs, with $\hat{Z} < Z < W$: if the game has not finished by this deadline, it is terminated by the LC.

When the game finishes before the deadline, the resulting scheduling corresponds to an equilibrium of the game, which ensures that resources are fairly shared among base stations. In contrast, when the game is terminated by the central coordinator, base stations use the scheduling that they computed in the latest iteration of the game, which does not correspond to an equilibrium. Thus, in the latter case some base stations could potentially have a better scheduling (i.e., more

[8]Note that there is no need to announce which specific user will be scheduled in a specific TTI, since base stations transmit at a fixed power and thus their activity causes the same level of interference independently of the scheduled user. Therefore, it is sufficient to propagate a binary string of T bits containing the ABSF pattern.

resources) than the others. However, as shown by our results of Section 4.5, we have observed that in practice the game can be interrupted after only a very few iterations without negatively impacting fairness or performance in a significant manner.

The deadline $\hat{Z}$ has been chosen in order to have a valid scheduling before the current period Z finishes: the resulting scheduling (and the corresponding ABSF pattern) will then be used for the next period. During the game, transmissions and users are scheduled according to the result of the previous period. Note that the iterations of game Ω, as it is in game Γ, do not need to be synchronized with the TTIs; they can be much faster, allowing for more than Z iterations within Z TTIs. Indeed, the execution of one iteration only requires passing the "current" ABSF patterns from one base station to another. As shown in Section 4.5.2, $\hat{Z}$ can be chosen in the range $[|\mathcal{N}|, |\mathcal{N}|^2]$.

Dynamic adjustment of TTI bounds M_i. One critical aspect for the performance of the proposed mechanism is the setting of the M_i parameters, which give the maximum number of non-blank TTIs available to each base station. Indeed, if the M_i values are too small, performance is degraded because, even if base stations can be scheduled one at a time with low interference, the number of TTIs available for transmitting can be too small to accomodate all users. Conversely, if the M_i values are too large, performance is degraded as a result of too many base stations scheduled together and creating high interference. Thus, performance is maximized when the M_i parameters are optimally set to values that are neither too large nor too small. In the rest of this section, we design an adaptive algorithm that follows an *additive-increase multiplicative-decrease* (AIMD) strategy [64] to find the optimal M_i setting.

In addition to optimally setting M_i to improve the performance of the network, the adaptive algorithm also aims at dynamically adjusting the M_i configuration to follow the changes in traffic and interference. From this perspective, the adaptive algorithm is a long-term process. In contrast, the distributed game is a short-term process played once per each period of W TTIs. This implies that the duration of the period W cannot exceed a few hundreds frames, which corresponds to a few seconds during which traffic and channel conditions remain practically unchanged.

From a high level perspective, the algorithm works as follows. At the end of each period of W TTIs, i.e., after the BE traffic serving period Z, the controller gathers from the base stations the performance resulting from the M_i values (and the corresponding ABSF pattern) used during the period. The metric chosen to represent the performance of a base station is given by the average user rate experienced by users of base station i in the period[9], i.e.:

$$\eta_i = \frac{1}{|\mathcal{U}_i|} \sum_{(u,t)\in\mathcal{U}_i\times\mathcal{Z}} c_{u,t}. \tag{4.3}$$

[9]Note that, since user allocation is carried out according to Problem LOCAL, the max-min objective tends to assign rates with limited variance; as a consequence, the average user rate and the rate of the worst-off user are likely to be similar.

The controller then uses the sum of the individual performance metrics, $\eta = \sum_{i \in N} \eta_i$, to keep track of the global system performance and drive M_i to the setting that maximizes η. The algorithm to find such M_i setting follows an AIMD strategy: the M_i values are increased as long as performance is improved, and, when performance stops improving, then the M_i values are decreased. After each update of the M_i values, these are distributed to the base stations and used in the following period (i.e., the following iteration of game Ω).

The specific algorithm executed to calculate the new set of TTI bounds M_i is described in Algorithm 2. Each iteration of the algorithm is identified by an index k. At the initial step ($k = 0$), the controller initializes the system performance metrics η to 0 and assigns the initial TTI bounds $M_i^* = \lceil T/|\mathcal{N}| \rceil$ for every base station. This initial M_i^* setting has been chosen to allow base stations to schedule their users in disjoint portions of the period, which helps the convergence of the algorithm in case of very high mutual interference between all base stations. The M_i^* also provide a lower bound for M_i.

Algorithm 2: Resource Sharing Algorithm: Adaptive algorithm to dynamically design M_i. Called at the end of $(k-1)^{th}$ ABSF pattern

Input: $\mathcal{N}, Z, M_i^*, \eta^{(k-1)}$
Initialization: $\eta^{(k)} \leftarrow 0; M_i \leftarrow M_i^*, \forall i \in \mathcal{N}$
Procedure

1: $\mathcal{V} \leftarrow \{\eta_i, \forall i \in \mathcal{N}\}$
2: Order $\mathcal{V}$ non-increasing
3: $\eta^{(k)} = \sum_{i \in N} \eta_i$
4: **if** $\eta^{(k)} > \eta^{(k-1)}$ **then**
5: **while** $\mathcal{V} \neq \emptyset$ **do**
6: e = pop($\mathcal{V}$)
7: **Consider** index i of element e
8: **if** $M_i^{(k-1)} < Z$ **then**
9: $M_i^{(k)} = M_i^{(k-1)} + 1$
10: **break**
11: **end if**
12: **end while**
13: **else**
14: **while** $\mathcal{V} \neq \emptyset$ **do**
15: e = pop($\mathcal{V}$)
16: **Consider** index i of element e
17: **if** $M_i^{(k-1)} > M_i^*$ **then**
18: $M_i^{(k)} = \max\left\{M_i^*; \left\lceil M_i^{(k-1)}/2 \right\rceil\right\}$
19: $\eta^{(k)} = 0$
20: **break**
21: **end if**
22: **end while**
23: **end if**

At each step, the controller collects the performance metrics η_i from base stations and checks whether the performance of this period, $\eta^{(k)}$, has improved with respect to the previous period, $\eta^{(k-1)}$ (line 3). If this is the case, this means that system performance is raising and the controller increases TTI bounds M_i as follows. The controller increases by 1 unit the M_i of the base station

with the smallest η_i whose M_i is below Z (lines 8-9). Once one M_i value is increased, step k of the algorithm terminates (line 10).

If no M_i can be increased, which means that all base stations are active in all TTIs, then no adjustment of the M_i values is made as long as the system performance does not degrade. In case performance degrades, i.e., $\eta^{(k)}$ decreases, (line 13), the controller drastically reduces the M_i. Specifically, the controller looks at the base station i with the largest η_i whose M_i is above M_i^*. It sets the new M_i value of this station equal to the minimum between the half of the current M_i value and the lower bound M_i^* (lines 17-18). If $M_i = M_i^*$ for all i, no change is carried out.

The rationale behind using AIMD to adjust the M_i values is that, similar to what happens with TCP, increasing the utilization of the system (i.e., increasing M_i values) may lead to congestion (in our case, this corresponds to excessive interference), which causes user rates to drop. In this case, a quick reaction is required by the controller to drive the system to a safe point of operation, by properly adjusting TTI bounds M_i. Also similar to TCP, the additive increase of TTI bounds M_i allows to gracefully approach the optimal utilization of the system. Furthermore, since the problem may admit more than one local maximum, using multiplicative decrease for the TTI bounds M_i helps our heuristic to escape from a local maximum where the optimization function may be trapped in.

As a side comment, we point out that the proposed algorithm could accommodate different goals, such as, e.g., maximum throughput or proportional fairness, by simply replacing the function that gives the global system performance, η, by another function that reflects performance according to the objective pursued.

In general, DMS framework is compatible with the SDN paradigm [1]. Only ABSF patterns and penalty indicators need to be exchanged in addition to the measure of traffic demands received by each base station. Therefore, with simple modifications, the X2 interface of LTE could be adopted as Southbound interface in an SDN implementation of DMS.

4.5. Performance Evaluation

In this section, we use numerical simulations to show that our mechanism performs near optimally and boosts achievable rates in the whole network, not just for topologically disadvantaged users. First, we present a simulation-based performance evaluation for serving guaranteed traffic and best-effort traffic in isolation. Then, we show how DMS jointly handles both traffic requests, exhibiting outstanding results. Lastly, we provide a computational evaluation about the control overhead introduced by DMS.

All simulations are carried out by means of MATLAB® with all parameters summarized in Table 4.4. Specifically, the tested network consists of $N = 7$ base stations regularly distributed in a rectangular area of size 300 m$\times$500 m. The coverage of each base station is computed as a Voronoi region, assuming all base stations use the same transmission power $P = 1$ Watt. Users are randomly dropped in each cell, according to a uniform random distribution. The average quality

Table 4.4: List of Parameters for the LTE-A wireless scenarios used in the experiments

$\|\mathcal{N}\|$	Number of Base Stations	7
$\|\mathcal{U}_i\|$	Number of UEs per Base Station	10
W	ABSF Pattern Length	70 TTIs
BW	Spectrum Bandwidth	20 MHz
P	Transmitting Power	1 Watt
ISD	Inter-Site Distance	200 m
N_0	Background Noise	1.085×10^{-14}

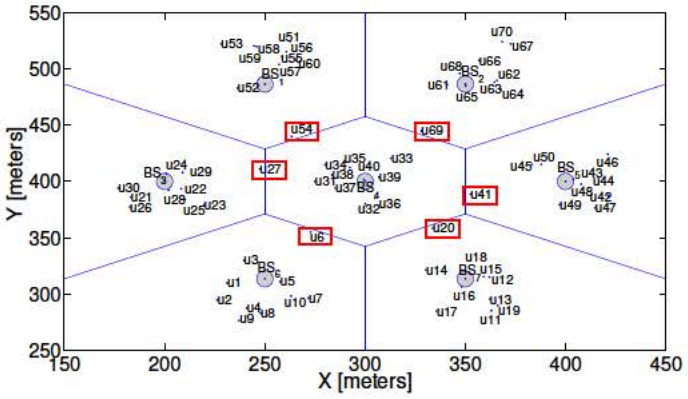


Figure 4.3: Reference network scenario. There are 7 base stations placed in a rectangle of size 500m× 300m. The figure shows an example with 10 users per base station, with a total of 6 *disadvantaged* users located at cell-edge.

of the user channel is computed as function of the distance from the base station (according to the propagation model provided by 3GPP specifications, Table A.2.1.1-3 of TR.25.814 v7.1.0), and Rayleigh fading is considered. Based on user channel qualities, each simulated base station solves the local optimization problem by means of a remote call to a commercial solver, i.e., IBM CPLEX OPL®.

4.5.1. Guaranteed traffic Management

We consider a variety of traffic demands and user distributions when DMS only handles guaranteed traffic requests. We show that DMS (i) fully guarantees guaranteed user traffic demands, (ii) minimizes resources used and make them available for best-effort traffic, and (iii) performs close to the bound corresponding to the ideal centralized scheme, presented by Problem GBR. Finally, we show the convergence of Γ, which is always achieved, even for hostile cellular environments.

4.5.1.1. Time-resources performance

To achieve traffic guarantees, we apply our DMS mechanism on the scenario described above, where all users in the network have subscribed a guaranteed bit-rate contract that provides them with 4 Mb/s. Furthermore, we benchmark DMS against Centralized, an omniscient centralized

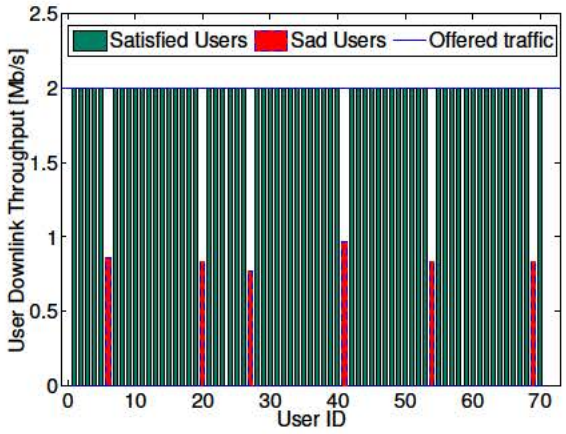


Figure 4.4: Per-user throughput for the specific topology shown in Fig. 4.3. Each user offers 4 Mb/s of inelastic traffic. Users located on cell-edge are not fully satisfied due to high interference conditions.

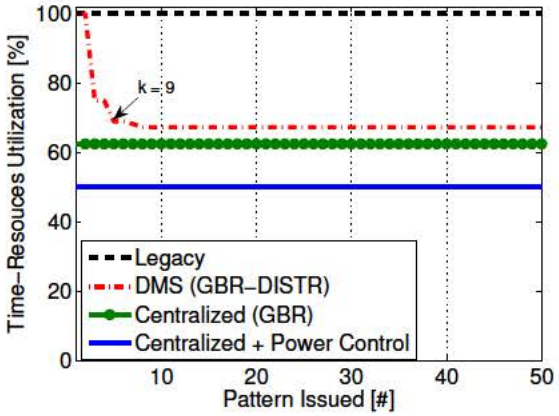


Figure 4.5: Percentage of TTIs utilized over an ABSF pattern of 70 TTIs under different schemes. DMS quickly adapts to the interferences showing an error about 7.5% w.r.t. the centralized approach.

approach able to optimally solve Problem GBR in the smallest possible number of TTIs. For the sake of completeness, we also compare our approach with a fully centralized power controlled solution, where transmitting powers are continuously tunable between 0 and 1 Watt in each TTI. Clearly, base stations receiving 0 Watt transmitting power are considered as inactive. Initially, the guaranteed traffic period T is set to the ABSF pattern $W = 70$. Iteratively, DMS reduces the guaranteed traffic period T while guaranteeing the required traffic. Presented results are averaged over multiple randomly generated instances of the simulated topology when the convergence is reached, e.g., after that DMS finds the minimum number of TTI T to successfully serve the guaranteed traffic.

Fig. 4.5 shows that our approach substantially outperforms the legacy solution (by using just 67% of the available TTIs) and exhibits near-optimal performance (DMS uses only 7.5% more

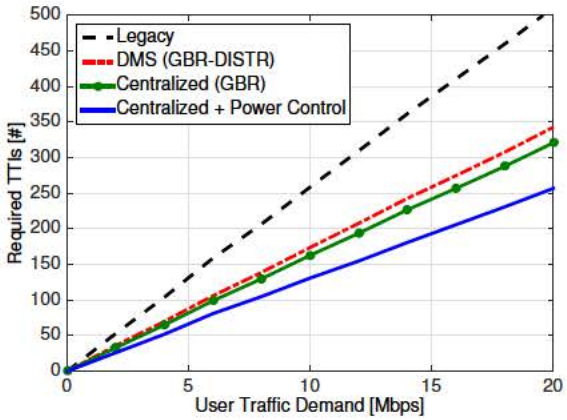


Figure 4.6: Number of required TTIs to guarantee traffic to 70 users (10 users per base station). In practice, the number of required TTIs increases linearly with the amount of per-user traffic demand.

TTIs than the Centralized scheme). The figure also shows that combining power control and a centralized approach for computing ABSF patterns would allow additional gain, although most of the gain with respect to the legacy scheme can be achieved by DMS. Therefore, considering the complexity of power control schemes, DMS can provide a practical and advantageous trade-off between performance and complexity.

Additionally, Fig. 4.5 surprisingly shows that our proposal quickly approaches the centralized solution curve. Indeed, after four or five ABSF patterns, DMS practically reaches a performance level similar to its best. It is worth noting that each single ABSF pattern in DMS has been obtained after that a Distributed Inelastic Game session has reached the convergence. Notably, for the case depicted in Fig. 4.5, convergence occurred in at most $k = 9$ rounds (for the point indicated with an arrow in the figure).

4.5.1.2. Performance in more generic cases

In Fig. 4.6 the number of TTIs used for guaranteeing the required traffic is shown as function of different user traffic demands. Results are averaged over 100 instances user random positions in the 7-base station scenario. As expected the number of required TTIs grows with the amount of user traffic. Most importantly, we need to mention that depicted curves are practically linear, resulting in a constant gain factor between the different schemes evaluated. Therefore, we confirm that with 70 users, DMS only needs 7.5% more resources than the Centralized scheme, while power control might potentially bring another 20% gain.

We study the volume of resources used under different network conditions in Fig. 4.7. In this case, we show the sources used under a given scheme as the average number of TTIs used by each base station, normalized to the number of TTIs used in the legacy approach. The resulting time-utilization index tells us how much each TTI is used by every base station in the network. The index ranges between $1/N$ (marked as Lower Bound), when base stations use resources in

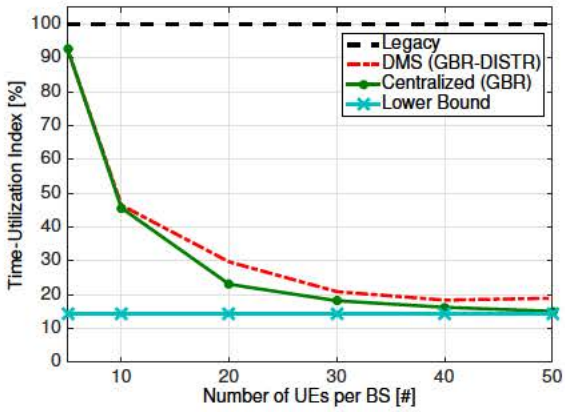


Figure 4.7: Time-utilization index over different network density values. The higher the overall interference is, the more base stations transmission are definitely disjointed.

Table 4.5: Study of the game convergence in terms of rounds to converge. Different density and inter-site distance values are assessed.

BS spacing \ #Users/BS	10	20	30	40
100 m	29.231	30.105	**35.422**	35.419
200 m	7.111	10.708	15.188	17.495
500 m	7.108	9.676	15.951	15.217
1000 m	6.941	7.155	7.244	7.239

a TDM-like way, and 1, like in the legacy scheme. The results are in line with our expectations. Considering a sparse network with a very low user population, inter-cell interference is limited and the time-utilization index is almost equal to 100%. Conversely, for high user populations the overall interference grows and base stations are forced to transmit disjointly over the ABSF patterns. Fig. 4.7 shows that DMS is near-optimal while the legacy approach keeps wasting time-resources allowing base stations being active all the time. Moreover, the figure confirms that DMS is not a conservative approach, since it allows to use slightly more intensively the TTIs with respect to the operation of the optimal scheme.

In general, presented results confirm that our approach substantially outperforms the legacy solution and exhibits near-optimal performance. The results also shows that combining power control and a centralized approach for computing ABSF patterns would allow additional gain, although most of the gain with respect to the legacy scheme can be achieved by DMS. Therefore, considering the complexity of power control schemes, DMS can provide a practical and advantageous trade-off between performance and complexity.

4.5.1.3. Convergence study of game Γ

From the theoretical analysis presented in Section 4.2, the Distribute Inelastic Game Γ is guaranteed to converge. Here we experimentally evaluate the convergence properties of Γ by simulating several network topologies, where different network densities and inter-site distances

are considered. Table 4.5 summarizes the results in terms of k, denoting the number of rounds the game needs to converge. The results are averaged over 1000 simulations per each single case. It is important to note that k inversely grows with the inter-site distance. This is due to the nature of inter-cell interference. The closer base stations are placed, the more the interference grows and the higher the number of rounds needed for Γ to converge.

Moreover, it is worth pointing out that only for 3 cases out of 1000 simulations the game Γ did not reach the convergence by using the *Best Response* strategy (BR), thus forcing players to use the *Single Step Best Response* strategy (SSBR), as explained in Section 4.2. SSBR is supposedly slow to reach convergence if used from round 1, however, it can readily achieve game convergence in a few rounds after BR has been played a few times. Specifically, in our simulation, about N^2 rounds with BR, followed by at most N rounds with SSBR, were sufficient to reach convergence in all cases. These results confirm not only that convergence can be always achieved, but also that the BR strategy typically ensures the game convergence, with no need to instruct the base stations to use the SSBR strategy since the beginning. In practice, we suggest to use the BR strategy during the first N^2 rounds of the game, and, if the game did not converged before, switch to the SSBR strategy at round $N^2 + 1$. With the above, the entire game will converge in a number of rounds in the order of $O(N^2)$.

4.5.2. Best-effort traffic Management

Once guaranteed traffic is properly accommodated within the ABSF pattern W, the non-used TTIs are fully assigned for serving best-effort traffic. In this set of simulations, we show how DMS handles the best-effort traffic given a fixed number of available TTIs $Z \leq W$.

First, we benchmark DMS against the optimal solution, obtained by solving Problem `BE` by means of an ILP solver. Additionally, we compare DMS to the case of uncontrolled base stations using the same frequencies (Legacy) and to a traditional frequency reuse 3 scheme, in which the available band is split into three orthogonal sub-bands. Finally, for the sake of completeness, we compare DMS with two existing approaches fully based on a power control schemes, showing how DMS can achieve high network performance at a bargain price of complexity. In the first scheme, namely Utility-Based Power Control (UBPC) [97], base stations are allocated in all available TTIs by tuning properly the transmitted power to reduce interference. The algorithm suggested in [97] maximizes the user net utility by ensuring that the signal-to-noise-ratio of each transmission is greater than a minimum threshold γ_i (in our simulations we assume γ_i as the minimum MCS with nonzero rate). While UBPC provides a rigorous centralized solution for the power allocation problem at the expense of a huge amount of information exchanged, a second power control scheme recently developed, namely REFerence based Interference Management (REFIM) [86], proposes a low-complex distributed scheme by exploiting the notion of reference user (e.g., the user with the worst channel condition, belonging to the surrounding cells). Although this abstraction leads to a drastic reduction of the control signal overhead and results in a practical implementation of the power control solution, it exhibits a conservative behaviour.

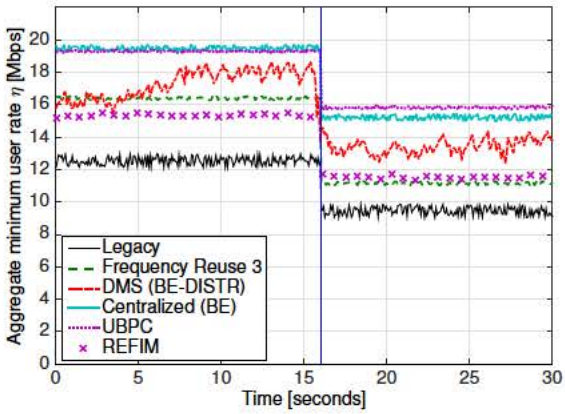


Figure 4.8: Dynamic behaviour of DMS for best-effort traffic applied to a changing scenario. On the left side, the scenario has $|\mathcal{N}| = 7$ base stations, $|U_i| = 10$ users and $T = 70$ TTIs. On the right side, the number of users is increased up to $|U_i| = 20$ users. DMS quickly adapts to a network change while keeping high the accuracy of the solution (w.r.t. centralized and legacy).

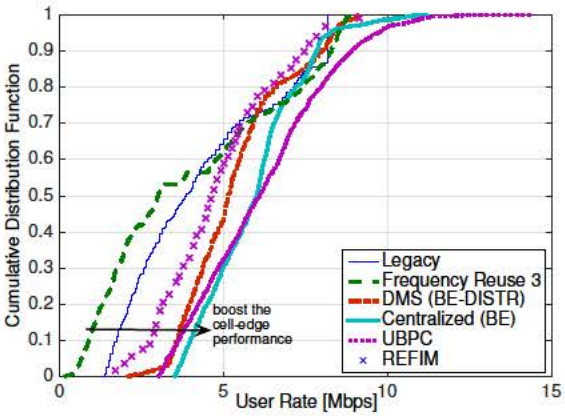


Figure 4.9: CDF of average user rates with 7 base stations and 10 users per base station. The time horizon is set to $T = 70$ TTIs.

4.5.2.1. Utility and fairness performance

We start evaluating the system utility $\widehat{\eta}$, which, according to the formulation of Problem BE, is the sum of minimum user rates experienced in the network. Fig. 4.8 shows $\widehat{\eta}$ for the 7-cells scenario above described when different schemes are applied.

Due to the adaptive nature of our algorithm, DMS shows a dynamic behavior. Specifically, it takes a few seconds for DMS to reach its stable operating point, after which it follows quite fast the evolution of channel and traffic conditions. In particular, at time $t = 16$ s, the number of users in the network doubles abruptly, but it takes only a fraction of a second for DMS to adapt. In general, DMS largely outperforms the Legacy scheme and achieves significant gain over frequency reuse 3. Indeed, DMS halves the distance between the optimal performance and the one of frequency reuse 3. Notably, after the initial adaptation period, the utility achieved by DMS lies within 85% and 90% of the one achieved with the optimal solution for Problem BE. REFIM

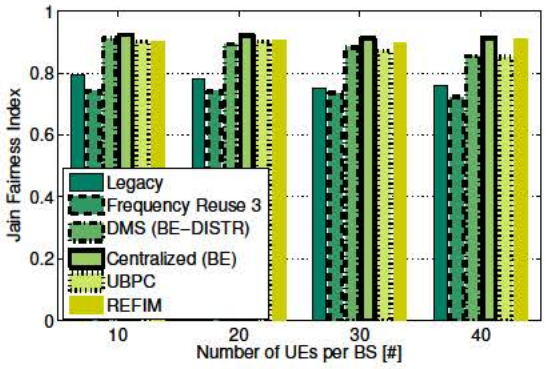


Figure 4.10: Jain fairness indexes achieved with 7 base stations and a variable number of users per base station.

and UBPC results show the real potentials of power control schemes. UBPC can even go slightly beyond the performance of the optimal solution without power control, although it requires higher complexity in terms both of execution and device hardware. REFIM, notwithstanding a low-complexity scheme, shows lower performance with respect to frequency reuse 3 for a particular set of user populations due to the conservative assumption taken on interfering cells. Therefore, our approach DMS perfectly lies in between an impermissible efficient power control scheme and a practical doable distributed power control solution.

Besides utility η, we want to evaluate the fairness achieved by the different schemes. To this aim, Fig. 4.9 presents the CDF of achieved user rates (averaged over the time horizon T). The figure clearly shows that the optimal solution, DMS and UBPC behave similarly and exhibit two main advantages: (i) they achieve user rates in a compact interval of possible values (which is symptom of fairness), and (ii) with high probability, they guarantee a minimum rate which is several times higher than the one guaranteed by Legacy or frequency reuse 3 (which is symptom of max-min fairness). In addition, REFIM shows a similar behaviour to the optimal solution in terms of fairness, even though its curve stays on the left side of the graph due to the critical user rates experienced by the users. For instance, with 95% probability, DMS guarantees 3.7 Mbps per user, REFIM guarantees 2.4 Mbps per user, while Legacy only guarantees 0.9 Mbps. Jain fairness achieved under the different schemes under evaluation is depicted in Fig. 4.10 as a function of the number of users per base station. As expected, REFIM presents a stable behaviour over different values of user population, due to its strong correlation with worst users. Also in this case, DMS achieves near-optimal results, and it outperforms UBPC.

4.5.2.2. Convergence study of game Ω

A key feature of DMS is its ability to adapt quickly to network changes. Such a feature relies on quick ABSF pattern computation, which follows the rule of the Interference Coordination Game Ω, and convergence guarantees. The game evolves over time as illustrated in Fig. 4.11. In this example two different cases are considered: the dashed line represents a case of convergence,

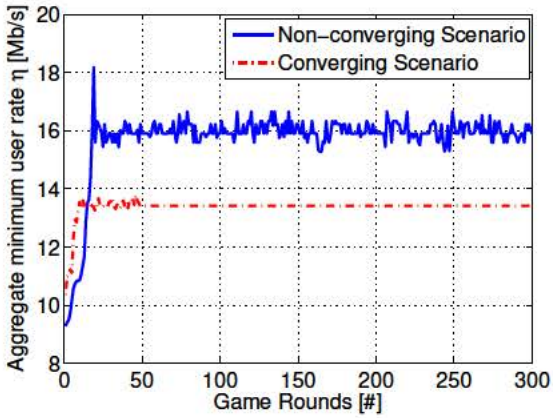


Figure 4.11: Game convergence behavior considering two different cases with $|\mathcal{N}| = 7$ base stations.

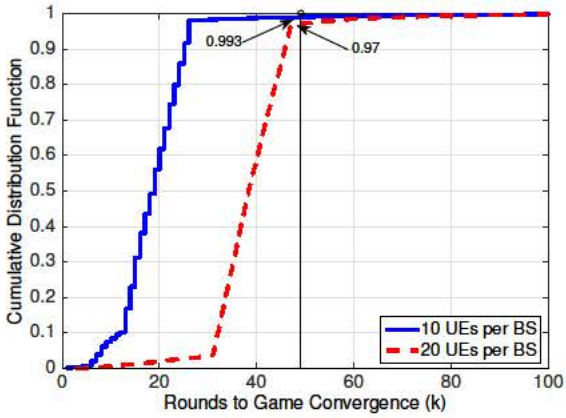


Figure 4.12: CDF of number of rounds needed for game convergence with 7 base stations and different user populations.

while the solid line is for a rare case in which the game does not converge to a Nash equilibrium point. In both cases, 7 base stations are considered, and the TTI bounds M_i are fixed.

In case of convergence, which occurs in about $|\mathcal{N}|^2$ rounds, it is clear that a few game rounds suffice to approximate the performance achieved at the Nash equilibrium with an error smaller than 3%. Notably, also in case the game fails to converge, after a few rounds the utility starts fluctuating around a stable value, with small oscillations (about $\pm 5\%$).

Interestingly, Fig. 4.12 illustrates the CDF of the number of rounds needed to converge for a few different user populations (note that we use the value 1 to indicate that the game did not converge). The figure shows how the majority of the games Ω converge much before $|\mathcal{N}|^2$ rounds (vertical line in the figure), and very few cases do not converge at all. We have observed very similar behavior for the majority of the cases analyzed in our experiments, so we conclude that reasonably high utilities can be achieved by stopping the game after a number of rounds comprised between $|\mathcal{N}|$ and $|\mathcal{N}|^2$.

Overall, our results show that DMS not only achieves near-optimal results according to the definition of utility given in the formulation of Problem BE, but also achieves high levels fairness, and boosts average rates in the entire cellular network, when best-effort traffic is required.

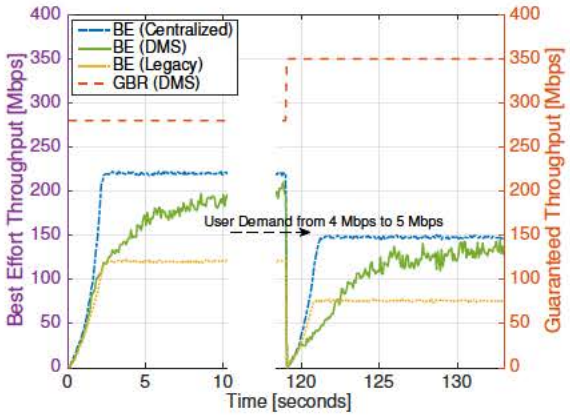


Figure 4.13: System throughput of DMS for both guaranteed (dashed line) and best effort traffic in a network area with 70 users, which require 4 Mbps each. After 120 seconds user demand increases to 5 Mbps. The time horizon is set to $W = 70$ TTIs.

4.5.3. Multi-traffic Service

A compound empirical evaluation is carried out to prove that minimizing the time needed to fully serve guaranteed traffic allows to optimally schedule best-effort traffic demands in the remaining part of the ABSF pattern. In Fig. 4.13 we show how DMS performs when applied to both guaranteed and best-effort traffic demands. We compared DMS with other approaches only when applied to best-effort traffics, considering only DMS mechanism for accommodating GBR traffic, as it has been exhaustively evaluated in Section 4.5.1. This shows that GBR and BE algorithms can work simultaneously and interestingly provides a valid benchmark when the BE time horizon Z dynamically changes.

While the guaranteed traffic is always served providing a guaranteed throughput equal to 4Mbps for 70 users, best-effort traffic is opportunistically accommodated in the saved time-pattern portion. In particular, DMS finds the optimal number of TTIs to fully accommodate the guaranteed traffic after 2 seconds, thus it dynamically schedules the best-effort traffic. DMS dramatically outperforms the legacy approach by unveiling near-optimal results when the game Ω convergence is reached after few seconds. Once the guaranteed traffic increases up to 5 Mbps per user at $t = 119s$ in Fig. 4.13, DMS reserves the full ABSF pattern for the GBR leaving no space to best-effort demands. Then, the optimization is repeated, showing again outstanding results when compared to a legacy solution and closely following the optimal results achieved by centrally solving the Problem BE (labelled as BE (Centralized)).

We can finally state that the advantage brought by DMS is twofold: (i) near-optimal results are easily achieved after few seconds without requiring high-complexity solution and (ii) the overall system capacity is drastically increased due to a smart using of system resources and inter-cell interference abating.

Table 4.6: Overhead of centralized and DMS semi-distributed approaches

Interface	Centralized approach	DMS approach
I_C	$64 \cdot \lvert U\rvert \cdot \lvert\mathcal{N}\rvert + T \cdot \lvert\mathcal{N}\rvert$	$64 \cdot \lvert\mathcal{N}\rvert$
I_B	0	$T \cdot k \cdot \lvert\mathcal{N}\rvert$

4.5.4. Control overhead

We conclude the analysis with the evaluation of the control overhead introduced by DMS. Both traffic serving procedures incur in the same amount of overhead except for the BE traffic service which requires additional 64 bits to notify the base station performance η_i to the local controller.

We can identify two different *interfaces*: one between local controller and base stations, namely I_C, and one between distinct base stations, namely I_B. They may be both implemented using, e.g., the LTE $X2$ interface [88].

In the centralized solution, the local controller requires message exchanges over I_C only. In particular, per each pair (*user, base station*), it requires the transmission of an average channel quality indicator (e.g., the *RSRP* value in the LTE-Advanced networks [92]) which can be encoded in double precision floating point format, e.g., 64 bits. Then, the controller issues a scheduling pattern (a string of T bits) per each base station.

In the DMS mechanism, the controller requires to receive the average user rate η_i per base station over I_C at the end of each game Ω for best-effort traffic only, consisting in a binary string of fixed length (e.g., 64 bits for a double precision floating point number). Regarding the interface I_B between different base stations, DMS needs a sequential exchange of ABSF scheduling patterns (strings of T bits) during the interference coordination games Γ and Ω, until both games reach a convergence state or the convergence deadline expires.

We can therefore summarize the total load in terms of bits for each interface as reported in Table 4.6. In the table, k is the number of rounds the interference coordination game plays before reaching the convergence, and $|\mathcal{U}| = \sum_i |\mathcal{U}_i|$ is the total number of users in the system. We can easily observe that the overhead of DMS is lower than that of the centralized mechanisms when the following inequality holds:

$$|U| > 1 + \frac{T}{64}(k-1) \cong \frac{T|\mathcal{N}|^2}{64}, \tag{4.4}$$

where we have considered that the number of rounds k in the worst case is a function of $|\mathcal{N}|$ (i.e., at most $k = |\mathcal{N}|^2$ iterations are enough to converge, when convergence exists, as proven mathematically in [41] and empirically shown in Section 4.5.1 and Section 4.5.2) and both T and $|\mathcal{N}|$ are (much) greater than 1. Therefore, our approach is convenient as soon as the number of users exceeds a threshold that depends on T and $|\mathcal{N}|$ (i.e., the threshold is $O\left(T|\mathcal{N}|^2\right)$). For example, in an (sub-)urban environment with $T = 70$ and $|\mathcal{N}| = 7$, DMS results convenient with as few as 54 users or more, while in a dense-urban environment with $|\mathcal{N}| = 30$, our approach exhibits a practical implementation starting with $\sim$1000 users in the entire network. Those values

are pretty low, revealing how *our approach drastically reduces the signaling overhead for existing cellular network sizes*.

4.6. Conclusions

In this chapter, we have presented the design of DMS, a practical (distributed and lighweight) approach to optimize inter-cell interference coordination for both guaranteed traffic and best-effort traffic. To design this approach, two optimization problems have been formulated, one for each traffic type, relying on game theory notions. We have then proposed distributed algorithms to solve these optimization problems, and have further conducted analysis to prove the convergence and stability of these algorithms. As a result, with the presented approach base stations make scheduling decisions for serving guaranteed traffic by using as few TTIs as possible, leaving the room for best-effort traffic, which is efficiently served.

Due to the simplicity of DMS and its limited control overhead, this can be considered as the first attempt towards a practical, efficient, scalable and adaptive implementation of ABSF in real networks addressing both traffic types. Numerical results show that DMS achieves near-optimal results with respect to a centralized omniscient network scheduler, and achieves performance levels similar to schemes relying on complex power control functionality.

Part III

Spectrum Management of Cellular Networks with D2D Offloading

Chapter 5

Cellular Traffic Offloading exploiting the Epidemic Dissemination

In the second part of this work, we focus on future heterogenous networks, in which device-to-device (D2D) technology brings an additional gain to the standard communication way. The target is to greatly boost the standard network performance by offloading the cellular traffic through opportunistic communications, performed by means of D2D links. A number of techniques have been proposed in the literature to offload cellular traffic, which are either based on heuristics (and hence do not ensure that the load of the cellular network is minimized) or fail to provide delay guarantees. However, we want to analyze how an efficient opportunistic offloading scheme may improve the spectral efficiency of the cellular network.

In this chapter, we present the HYPE (HYbrid oPportunistic and cEllular) application, which *minimizes* the load of the cellular network while meeting the constraint in terms of *delay guarantees*. The key features can be easily summarized as follows:

- Building on the foundations of *epidemic analysis* [30], we propose a model to understand the fundamental trade-offs and evaluate the performance of a hybrid opportunistic and cellular communication approach. Our model reveals that content tends to disseminate faster through opportunistic contacts when a sufficient, but not excessive, number of nodes have already received the content; in contrast, dissemination is slower when either few users have the content or few users are missing it.

- Based on our model, we derive the optimal strategy for injecting content through the cellular network. In line with our previous findings, this strategy uses the cellular network when low speed of opportunistic propagation is statistically expected, and lets the opportunistic network spread the content the rest of the time.

- We design an adaptive algorithm, based on control theory, that implements the optimal strategy for injecting content through the cellular network. The key strengths of this algorithm over previous approaches are that it adapts to the current network conditions

without monitoring the nodes' mobility and that it incurs very low signaling overhead and complexity. Both features are essential features for a practical implementation.

5.1. The HYPE approach

In this section, we present the basic design guidelines of the HYPE (HYbrid oPportunistic and cEllular) approach. HYPE is a hybrid cellular and opportunistic communications approach that delivers content to a set of users by

- sending the content through the cellular network to an initial subset of the users (which we will call *seed nodes*), and

- letting these initial users or seed nodes share the content opportunistically with the other nodes.

We aim at designing HYPE so as to combine the cellular and opportunistic communication paradigms in a way that retains the key strengths of each paradigm, while overcoming their drawbacks.

HYPE consists of two main building blocks: (i) the *Content Server*, and (ii) the *Mobile Applications*. The Content Server runs inside the network infrastructure, while the Mobile Applications run in mobile devices that are equipped with cellular connectivity, as well as able to directly communicate with each other via short range connections (e.g., via WLAN or Bluetooth). The Content Server monitors the Mobile Applications and, based on the feedback received from them, delivers the content through the cellular network to a selected subset of Mobile Applications (the seed nodes). When two mobile devices are within transmission range of each other, the corresponding Mobile Applications opportunistically exchange the content by using local (short-range) communications.

5.1.1. Objectives

The fundamental challenge of the HYPE approach is the design of the algorithm that decides *which* mobile devices and *when* they should receive the content through the cellular network. The rest of this chapter is devoted to the design of such an algorithm. The key objectives in the design are:

- *Maximum Traffic Offload*: Our fundamental objective is to maximize the traffic offloaded and thus reduce the load of the cellular network as much as possible. This is beneficial both for the operators (who may otherwise need to upgrade their network, if the cellular infrastructure is not capable of coping with current demand), as well as for the users (who must pay for cellular usage, either directly or by seeing their data rate reduced).

- *Guaranteed delay*: Most types of content have an expiration time, arising either from the content's usefulness to the user (e.g., road traffic information), its validity after an update (e.g., daily news) or its play-out time (e.g., streaming). Therefore, a key requirement for our approach is that the content reaches all the interested users before its deadline.

- *Fairness among users*: In order to make sure that all users benefit from HYPE, it is important to guarantee a good level of fairness both in terms of cellular usage (for which users have to pay), as well as in terms of opportunistic communications (which may increase the energy consumption of the device).[1]

- *Reduced signaling overhead*: The signaling overhead between the Content Server and the Mobile Applications needs to be low. This is important for two reasons: first, to ensure that HYPE *scales* with the number of mobile devices (otherwise the signaling traffic would overload the cellular network); second, to avoid using the cellular interface for small control packets (which is highly energy inefficient due to the significant tail consumption after a cellular transmission [12]).

The above objectives involve some trade-offs, making it very challenging to satisfy all of them simultaneously. For instance, to maximize the traffic offload, we may consider a greedy approach, where the Content Server sends the content to users with the highest contact rates; however this would (i) deteriorate the fairness among users, and (ii) increase the signaling overhead to gather data on user mobility patterns. Another approach may instead minimize the signaling overhead by injecting content as long as there is enough bandwidth available, avoiding thus any signaling; however, this will not maximize the traffic offload. In the following, we set the basic design guidelines of an approach that satisfies all these objectives.

5.1.2. Basic design guidelines

In order to satisfy the above objectives, a key decision of HYPE is how to deliver a certain piece of content (hereafter referred to as *data chunk*) through the cellular network. In particular, this decision involves the selection of the nodes to which the data chunk is delivered via cellular, as well as the times when to perform these deliveries.

In HYPE, a data chunk is initially delivered to one or more users through the cellular network; additional copies may be injected later if needed. The decision of when to inject another copy of the chunk is driven by the number of users that have already received it. As long as the deadline has not expired, any user with a copy of the chunk will opportunistically transmit it to all the users it meets, that do not have the chunk. Finally, upon reaching the deadline of the content, the remaining users that have not yet received the chunk, download it from the cellular network;[2] this

[1] Indeed, an important drawback of certain existing solutions is that they tend to over-exploit the users with high contact rates [40, 45], thus discouraging the participation of such users.

[2] An added advantage of this architecture is that the mobile nodes only need to keep the data chunks for forwarding until their deadline and no longer. The burden on the mobile nodes' buffers is thus kept very low.

ensures that the *delay guarantees* are met and thus we satisfy objective (5.1.1) from Section 5.1.1.

In order to provide a good level of fairness among users, which is objective (5.1.1), HYPE selects each of the seed nodes uniformly at random. Over the long term, this ensures that, on the one hand, all users have the same load in terms of cellular usage and, on the other hand, they also share fairly well the load incurred in opportunistic communications. This is confirmed by the simulation results presented in Section 5.3, which show that HYPE provides a good level of fairness while paying a small price in terms of performance.[3]

The approach sketched above meets objectives (5.1.1) and (5.1.1). In the following, we first present a model for the opportunistic dissemination of content injected by a cellular network. Based on this model, in Section 5.2 we derive the optimal strategy for the delivery of a single data chunk, that minimizes the load of the cellular network fulfilling objective (5.1.1), and then we design an algorithm to implement this strategy, that incurs very low signaling overhead thus also satisfying objective (5.1.1).

5.1.3. Model

In order to derive the optimal strategy, with the above approach, for the delivery of data chunks through the cellular network, we need to determine:

- The *total number of copies* of the data chunk to be delivered by the cellular network. This is not trivial: for example, an overly conservative approach, that delivers too few copies before the deadline, may have the side-effect of overloading the cellular network with a large number of copies when the deadline expires.

- The *optimal instants* for their delivery. The decision of when to deliver a copy of a data chunk through the cellular network is based on the current status of the network, which is given by the number of users that already have the chunk.

In the following, we model the opportunistic dissemination of content injected by a cellular network and analyze the load of the cellular network as a function of the strategy followed. Then, based on this analysis, in Section 5.2 we obtain the optimal strategy, that minimizes the load of the cellular network for a given content deadline.

Let $\mathcal{N}$ be a set of mobile nodes subscribed to the same content, with $N = |\mathcal{N}|$ the size of this set (total number of nodes). All nodes have access to the cellular network. Any two nodes also have the ability to setup pairwise bi-directional wireless links, when they are in each other's communication range (in *contact*). Thus, opportunistic communication happens via the store-carry-forward method, through the sequences of intermittent contacts established by node mobility.

At time 0, a data chunk is injected in the (opportunistic) network, i.e., copies of the chunk are pushed via the cellular interface to a small subset of $\mathcal{N}$, the seed nodes. Throughout the model

[3] This is also supported by the results of [96], which show that the difference in terms of performance between the random selection and other strategies is very small.

description, we follow the epidemic dissemination of this chunk of content. We denote by $M(t)$ the number of mobile nodes holding the chunk at time t (we refer to such nodes as "infected"). The delivery deadline assigned to a data chunk is given by T_c (its value depends on the mobile application's requirements).

5.1.3.1. Opportunistic communication

In the opportunistic phase of HYPE, data are exchanged only upon contacts in the network $\mathcal{N}$, therefore a mobility model based on contact patterns is sufficient for our analysis. We assume every pair of nodes (x, y) in the network $\mathcal{N}$ meets independently of other pairs, at exponentially distributed time intervals[4] with rate $\beta_{xy} \geqslant 0$. Then, the opportunistic network $\mathcal{N}$ can be represented as a weighted contact graph using the $N \times N$ matrix $\mathbf{B} = \{\beta_{xy}\}$. We further assume that the inter-contact rates β_{xy} are samples of a generic probability distribution $F(\beta) : (0, \infty) \rightarrow [0, 1]$ with known expectation μ_β (various distribution types for $F(\beta)$ and their effects on aggregated inter-contact times are investigated in [74]). Additionally, we assume that the duration of a contact is negligible in comparison to the time between two consecutive contacts, and that the transmission of a single chunk is instantaneous in both the cellular and the opportunistic network.

The assumptions of exponential inter-contact and negligible contact duration are the norm in analytical work dealing with opportunistic networks [38, 72, 76]. Studies based on looser assumptions (generic inter-contact models, non-zero contact duration) have, so far, only resulted in broad, qualitative conclusions (e.g., infinite vs. finite delay), while we aim at obtaining more concrete, quantitative results. In addition, all our simulations feature non-zero contact duration and some of them also have non-exponential inter-contact times, thus testing the applicability of our results outside the domain of these assumptions.

Epidemic dissemination in opportunistic networks is typically described with a pure-birth Markov chain, similar to the one in Fig. 5.1 (slightly adapted from, e.g., [38]). This type of chain only models the *number of copies* of a chunk in the network $\mathcal{N}$ at any point in time, regardless of the specific nodes carrying those copies. This is only possible when considering node mobility to be entirely homogeneous (i.e., all node pairs meet at a unique rate: $\beta_{xy} = \lambda$ for all $x, y \in \mathcal{N}$), which allows all nodes to be treated as equivalent.

However, as stated in the beginning of this subsection, we consider node mobility to be heterogeneous, with node pairs meeting at different rates β_{xy} with $x, y \in \mathcal{N}$. In this case, not only the number of spread copies must be modeled, but also the *specific nodes carrying those copies*. This results in more complex Markov chains, as illustrated in Fig. 5.2 for a 4-node network $\mathcal{N} = \{a, b, c, d\}$.

[4] Though all pairwise inter-contact rates may not always be exactly exponential (preliminary studies of traces [29] suggested that this is true for subsets of node pairs only), the most in-depth and recent studies [21, 53] conclude that inter-contact time intervals do feature an exponential tail. This is supported by the recent results of Passarella et al. [74], which show that the non-exponential aggregated inter-contacts discovered in the preliminary trace studies [29] can, in fact, be the result of exponentially distributed pairwise inter-contacts with different rates.

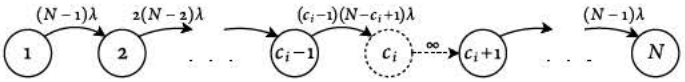


Figure 5.1: Markov chain for HYPE communication, assuming **homogeneous node mobility**. Transitions can be caused either by (i) a contact between two nodes, or (ii) injection of the chunk to one node through the cellular network (instantaneous transition, represented with ∞ rate in the figure).

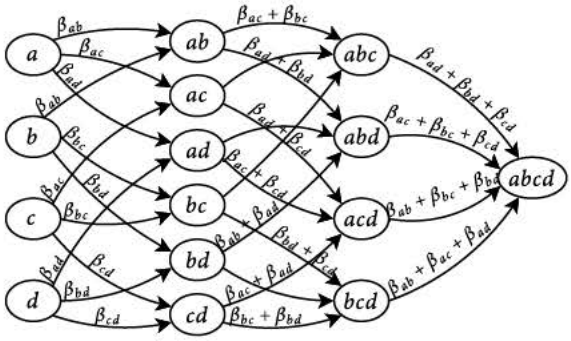


Figure 5.2: Markov chain for epidemic spreading, assuming **heterogeneous node mobility**. HYPE specific transitions (i.e., chunk injection by cellular) are left out for clarity. This Markov chain is very complex and intractable for large scenarios; in Theorem 5.1.1 we can then reduce it to an equivalent Markov chain that is much simpler and for which we can derive a closed-form solution.

Transition rates in Markov chains like the one shown in Fig. 5.2 depend on the nodes "infected" in each of the departure and the arriving states. For example, in Fig. 5.2, the transition between state a and state ab can happen if node a meets node b. Therefore, the transition time between these two states is exponential with rate given by the meeting rate of the (a, b) node pair, β_{ab}. Similarly, the transition between state ab and state abc can happen if node a meets node c, or if node b meets node c (whichever meeting happens first). Thus, the transition time for this transition is the minimum of two exponential variables with rates β_{ac} and β_{bc}. Since inter-contact times are exponential, this minimum is also exponential with rate $\beta_{ac} + \beta_{bc}$, as shown in Fig. 5.2.

5.1.3.2. Cellular communication

The decision to deliver a copy of the chunk through the cellular network is based on the current dissemination level, i.e. the number of nodes that already have the chunk. We say that the HYPE process or its associated Markov chain (similar to Fig. 5.2) is at ***level*** i, when i mobile nodes are infected, i.e., $M(t) = i$. Each level i corresponds to a set of $\binom{N}{i}$ states $\{\mathbf{K}_1^i, \mathbf{K}_2^i, \ldots, \mathbf{K}_{\binom{N}{i}}^i\}$ in the Markov chain. For instance, in our 4-node network from Fig. 5.2, the HYPE process is at level 3, when the chain is in any of the states $\mathbf{K}_1^3 = abc$, $\mathbf{K}_2^3 = abd$, $\mathbf{K}_3^3 = acd$ or $\mathbf{K}_4^3 = bcd$.

The strategy to transmit copies of the chunk over the cellular network is given by the levels at which we inject a copy. We denote these levels by $C = \{c_1, c_2, \ldots, c_d\}$: as soon as we reach one of these levels $c_i \in C$ before the deadline T_c, a copy of the chunk is sent to a randomly chosen node. With this, the transitions in the HYPE Markov chain can be caused either by:

- a contact between two nodes (one infected, the other uninfected), which occurs at rates indicated in the previous subsection, or
- the injection of the chunk to one node through the cellular network.

The latter corresponds to an instantaneous transition (since the chain instantly "jumps" to a state of the next dissemination level), and is represented in Fig. 5.1 with ∞ rate[5]. Finally, upon reaching the deadline T_c, the chunk is sent through the cellular network to those nodes that do not have the content by that time.

5.1.4. Analysis

Based on the above model, in the following, we analyze the load of the cellular network (which is the metric that we want to minimize) as a function of the strategy followed to inject content (which is given by $C = \{c_1, c_2, \ldots, c_d\}$). The cellular network load corresponds to the number of copies delivered through the cellular network, which we denote by D. Let $p_i(t) = \mathbb{P}[M(t) = i]$ denote the probability of being at level i at time t. Then, D is given by:

$$D = \sum_{i=1}^{N} (d_i + d_i^*) p_i(T_c) \tag{5.1}$$

where d_i is the number of deliveries through the cellular network that take place until level i is reached ($d_i = |\{1, 2, \ldots, i\} \cap C|$) and d_i^* is the number of copies delivered upon reaching the deadline T_c, if it expires at level i ($d_i^* = N - i$).

In order to compute $p_i(T_c)$, we first analyze the case $C = \{c_1\}$[6], i.e., when we only inject one copy of the data chunk at the beginning and do not inject any other until we reach the deadline. Let $p_i^{c_1}(T_c)$ denote the probability that, in this case, the system is at level i at time T_c. In order to compute $p_i^{c_i}(T_c)$, we model the transient solution of our Markov chain as shown in the following theorem. (The formal proofs of the theorems are provided in the Appendix.)

Theorem 5.1.1. *According to the HYPE Markov chain for heterogeneous mobility (similar to Fig. 5.2), the process $\{M(t), t \geq 0\}$ is described by the following system of differential equations:*

$$\begin{cases} \frac{d}{dt} p_1^{c_1}(t) = -\lambda_1 p_1^{c_1}(t), & i = 1 \\ \frac{d}{dt} p_i^{c_1}(t) = -\lambda_i p_i^{c_1}(t) + \lambda_{i-1} p_{i-1}^{c_1}(t), & 1 < i < N \\ \frac{d}{dt} p_N^{c_1}(t) = \lambda_{N-1} p_{N-1}^{c_1}(t), & i = N \end{cases} \tag{5.2}$$

where $\lambda_i = i(N - i)\mu_\beta$. (Recall that μ_β is the known expectation of the generic probability distribution $F(\beta) : (0, \infty) \to [0, 1]$, from which the inter-contact rates describing our network are drawn: $\{\beta_{xy}\} = \mathbf{B}$.)

[5]Note that, for clarity, the Markov chain of Fig. 5.2 does not model transitions caused by chunk injection through the cellular network. This type of transition would be the same as in Fig. 5.1 (i.e., ∞ rate).

[6]Note that c_1 must necessarily be equal to 0.

Theorem 5.1.1 has effectively reduced our complicated Markov chain for heterogeneous mobility back to a simpler Markov chain, like the one in Fig. 5.1 (the λ factor being replaced by μ_β). In the simpler chain, each state represents a level of chunk dissemination (i.e., number of nodes holding a copy of the chunk). This is possible, as shown in the proof, thanks to the fact that our heterogeneous contact rates β_{xy} are all drawn from the same distribution, $F(\beta) : (0, \infty) \to [0, 1]$, which means that all the states of a certain dissemination level i: $\{\mathbf{K}_1^i, \mathbf{K}_2^i, \ldots, \mathbf{K}_{\binom{N}{i}}^i\}$ are, in fact, statistically equivalent.

Applying the Laplace transform to the above differential equations, and taking into account that $p_i^{c_1}(0) = \delta_{i1}$, leads to

$$\begin{cases} sP_1^{c_1}(s) = -\lambda_1 P_1^{c_1}(s) + 1, & i = 1 \\ sP_i^{c_1}(s) = -\lambda_i P_i^{c_1}(s) + \lambda_{i-1} P_{i-1}^{c_1}(s), & 1 < i < N \\ sP_N^{c_1}(s) = \lambda_{N-1} P_{N-1}^{c_1}(s), & i = N \end{cases} \tag{5.3}$$

from which

$$\begin{cases} P_i^{c_1}(s) = \dfrac{1}{s + \lambda_i} \displaystyle\prod_{j=1}^{i-1} \frac{\lambda_j}{s + \lambda_j}, & i < N \\ P_N^{c_1}(s) = \dfrac{1}{s} \displaystyle\prod_{j=1}^{N-1} \frac{\lambda_j}{s + \lambda_j}, & i = N \end{cases} \tag{5.4}$$

In case we deliver the data chunk through the cellular network at the levels $C = \{c_1, c_2, \ldots, c_d\}$, then the transitions corresponding to those levels are instantaneous, and the Laplace transforms of the probabilities $P_i(s)$ are computed as:

$$P_i(s) = \begin{cases} \dfrac{1}{s + \lambda_i} \displaystyle\prod_{j \in S_{i-1}} \frac{\lambda_j}{s + \lambda_j}, & i < N,\ i \notin C \\ 0, & i < N,\ i \in C \\ \dfrac{1}{s} \displaystyle\prod_{j \in S_{N-1}} \frac{\lambda_j}{s + \lambda_j}, & i = N \end{cases} \tag{5.5}$$

where S_{i-1} is the set of levels up to level $i - 1$, without including those that belong to set C, i.e., $S_{i-1} = \{1, 2, \ldots, i - 1\} \setminus (\{1, 2, \ldots, i - 1\} \cap C)$. For the levels $i \in C$, we simply have $P_i^C(s) = 0$, since we will never be at these levels.

From Eq. (5.5), we can obtain a closed-form expression for the probabilities $p_i(t)$ as follows. The polynomial $P_i(s)$ is characterized by first and second order poles which have all negative real values. Let $\{s = -\lambda_n\}$ be the poles of $P_i(s)$. Then, $p_i(t)$ for $i < N, i \notin C$ is computed as:

$$p_i(t) = \left(\prod_{j \in S_{i-1}} \lambda_j \right) \sum_{\{s = -\lambda_n\}} Res \left(\frac{e^{st}}{\prod_{j \in S_i} (\lambda_j + s)} \right) \tag{5.6}$$

where Res indicates the residue, which is given by:

$$\underset{s=-\lambda_n}{Res}\left(\frac{e^{st}}{\prod\limits_{j\in S_i}(\lambda_j+s)}\right)=$$

$$\begin{cases}\dfrac{e^{-\lambda_n t}}{\prod\limits_{\substack{j\in S_i\\ j\neq n}}(\lambda_j-\lambda_n)}, & -\lambda_n \text{ is a } 1^{\text{st}} \text{ order pole}\\[2ex] \dfrac{e^{-\lambda_n t}\left[t-\sum\limits_{\substack{j\in S_i\\ \lambda_j\neq\lambda_n}}\dfrac{1}{(\lambda_j-\lambda_n)}\right]}{\prod\limits_{\substack{j\in S_i\\ \lambda_j\neq\lambda_n}}(\lambda_j-\lambda_n)}, & -\lambda_n \text{ is a } 2^{\text{nd}} \text{ order pole}\end{cases}$$

Additionally, for $i<N, i\in C$ we have $p_i(t)=0$, and for $i=N$, $p_N(t)=1-\sum_{i=1}^{N-1}p_k(t)$.

By evaluating $p_i(t)$ at time $t=T_c$ and applying Eq. (5.1), we can compute the average number of deliveries over the cellular network, D.

5.2. Optimal Strategy and Adaptive Algorithm

In this section, we first leverage on the above model to determine the optimal strategy for the delivery of data chunk, and then we design an adaptive algorithm to implement this strategy.

5.2.1. Optimal strategy analysis

Our goal is to find the best strategy $C=\{c_1,c_2,\ldots,c_d\}$ for injecting chunk copies over the cellular network, that minimizes the total load D of the cellular network while meeting the content's deadline T_c. To solve this optimization problem, we proceed along the following two steps:

- We show that the optimal strategy is to deliver the content through the cellular network only at the beginning and at the end of the data chunk's *period*, and never in-between. The data chunk's *period* is defined as the interval between $t=0$ (when we first start distributing the content) and $t=T_c$ (when the content's deadline expires).

- We obtain the optimal number of copies of the chunk to be delivered at the beginning of the period such that the average load of the cellular network, D, is minimized.

The following theorem addresses the first step.

Theorem 5.2.1. *In the optimal strategy, the data chunk is delivered through the cellular network to d seed nodes at time $t=0$, and to the nodes that do not have the content by the deadline at time $t=T_c$.*

According to Theorem 5.2.1, the optimal strategy is to:

- deliver a number of copies through the cellular network at the beginning of the period,
- wait until the deadline *without* delivering any additional copy,
- deliver a copy of the chunk to the mobile nodes missing the content at the end of the period.

The intuition behind this result is as follows. When few users have the content, information spreads slowly, since it is unlikely that a meeting between two nodes involves one of the few that have already the content. Similarly, information spreads slowly when many users have the content, as a meeting involving a node that does not yet have the content is improbable.

The strategy given by Theorem 5.2.1 avoids the above situations by delivering a number of chunk copies through cellular communication at the beginning (when few users have the content) and at the end (where few users miss the content). As a result, the strategy lets the content disseminate through opportunistic communication when the expected speed of dissemination is higher, which allows to minimize the average load of the cellular network.

The second challenge in deriving the optimal strategy is to compute the optimal number of copies of the chunk to be delivered at the beginning of the period, which we denote by d. To that end, the following proposition defines the notion of gain and computes it:

Proposition 5.2.1. *Let us define G_d as the gain resulting from sending the $(d+1)^{th}$ chunk of chunk copy at the beginning of the period (i.e., $G_d = D_d - D_{d+1}$, where D_{d+1} and D_d are the values of D when we deliver $d+1$ and d copies at the beginning, respectively). Then, G_d can be computed from the following equation:*

$$G_d = \sum_{j=d}^{N-1} \frac{\lambda_j}{\lambda_d} p_j^d(T_c) - 1. \tag{5.7}$$

Building on the above notion of G_d, the following theorem provides the optimal point of operation:

Theorem 5.2.2. *The optimal value of d is the one that satisfies $G_d = 0$.*

The rationale behind the above theorem is as follows. When $G_d > 0$, by sending one additional copy at the beginning, we save more than one copy at the end of the period and hence obtain a gain. Conversely, when $G_d < 0$, we do not benefit from increasing d. The proof shows that G_d is a strictly decreasing function of d, which implies that, to find the optimal point of operation, we need to increase d as long as $G_d > 0$ and stop when we reach $G_d = 0$ (after this point, $G_d < 0$ and further increasing d yields a loss).

5.2.2. Adaptive algorithm for optimal delivery

While the previous section addressed the delivery of a single data chunk, in this section we focus on the delivery of the entire content, e.g., a flow of road traffic updates, news feeds or a streaming sequence. We consider that the distribution of content in mobile applications is typically performed by independently delivering different pieces of content in a sequence of data chunks. For instance, a streaming content of 800 MB may be divided into a sequence of chunks of 1 MB. When delivering chunks in sequence, we need to adapt to the system dynamics. For instance, inter-contact time statistics may vary depending on the time of the day [24], which means that the optimal d value obtained by Theorem 5.2.2 needs to be adapted accordingly. Similarly, the number of mobile nodes N subscribed to the content may change with time, e.g., based on the content popularity.

To address the above issues, we design an adaptive algorithm based on control theory, that adjusts the number of chunk copies d delivered at the beginning of each period to the behavior observed in previous *rounds* (hereafter we refer to the sequence of periods as rounds). For instance, in the example above we would have a total of 800 rounds. In the following, we first present the basic design guidelines of our adaptive algorithm. Building on these guidelines, we then design our system based on control theory. Finally, we conduct an analysis of the system to guarantee its stability and ensure good response times.

5.2.3. Adaptive algorithm basics

In order to devise an adaptive algorithm that drives the system to optimality, we first need to identify *which variable* we should monitor and *what value* this variable should take in optimal operation. To do this, we build on the results of the previous section to design an algorithm that:

- monitors *how many additional infected nodes* we would have at the end of a round, if we injected one extra copy at the beginning of that round;

- drives the system to optimality by increasing or decreasing d depending on whether this number is above or below its optimal value.

To efficiently monitor the number of additional infected nodes, we apply the following reasoning. According to Theorem 5.2.2, in optimal operation, one extra delivery at the beginning of a round leads to one additional infected node at the end of that round. If we focus on a single copy of the chunk delivered over the cellular network and consider it as the extra delivery, the nodes that would receive the content due to this one extra delivery are those that *received this specific copy and could not have received the chunk from any other source*. Since this holds for each of the d copies delivered over the cellular network, in optimal operation there are on average a total of d nodes at the end of the round, which received the chunk from one source and could not have received it from any other source. Our algorithm focuses on this aggregate behavior of

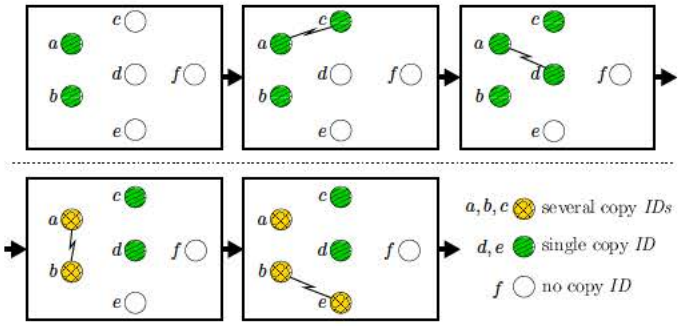


Figure 5.3: Example of chunk dissemination in optimal operation. Node a and b receive a copy of the chunk from the Content Server ($d = 2$). At the end of the round, there are two nodes with a single copy ID, that is, $s = 2$.

the d deliveries rather than on a single copy, as this provides more accurate information about the epidemic dissemination of the data chunk.

Based on this, each round of the adaptive algorithm proceeds as follows (see Fig. 5.3 for an example):

- Initially, copies of the data chunk are transmitted to a random set of d seed nodes over the cellular network. Each of the copies is marked with a different *ID* that uniquely identifies the source of the copy.

- When a node that does not have the chunk receives it from another node opportunistically, it records the *ID* of the copy received.

- If two nodes that have copies with different *ID*s meet, they mark this event, to record that they could have potentially received the chunk from different sources.[7] We say that such nodes have "several copy *ID*s", while those that keep only one *ID* have a "single copy *ID*".

- If a node who does not have any copy or has a single copy *ID* meets with another node who recorded the "several copy *ID*s" event, the first node also marks its copy with the "several copy *ID*s" mark.

- At the end of the round, the nodes whose chunk comes from a single source (i.e., no "several copy *ID*s" mark) send a signal to the Content Server.

By running the above algorithm, we count the number of nodes whose copy of the chunk comes from a single source (i.e., who have a single copy *ID* at the end of the round), which we denote by s. As argued at the beginning of this section, in optimal operation this number is equal to the number of seed nodes. This implies that, at this operating point, *the number of data chunks*

[7] Note that, for a node with "several copy *ID*s", we only mark the event and do not keep the *ID*s of the copies, since (i) we are only interested in signaling the number of nodes with a single copy *ID*, and (ii) this leads to more efficient operation, requiring fewer communications and less protocol overhead.

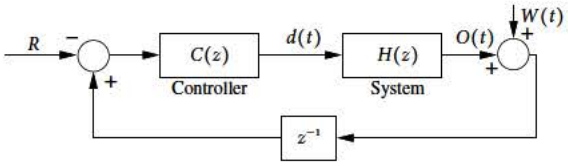


Figure 5.4: Our system is composed by two modules: the controlled system $H(z)$, that models the behavior of HYPE, and the PI controller $C(z)$, that drives the controlled system to the optimal point of operation.

injected through the cellular network at the beginning of the round is equal, in expectation, to *the number of signals received at the end of the round*, i.e., $s = d$.

A key feature of the above algorithm design is that it does not require gathering any complex statistics on the network, such as the behavior of the mobile nodes, their mobility or social patterns, or their contact rates. Instead, we just need to keep track of the number of chunks injected at the beginning of each round and the signals received at the end, and this is sufficient to drive the system to optimal operation. As a result, the proposed algorithm involves very reduced signaling overhead, which fulfills one of the objectives that we had identified in Section 5.1.1, namely objective (5.1.1).

5.2.4. System design

Based on the above design guidelines, our adaptive algorithm should (i) monitor the number of signals received at the end of each round, and (ii) drive the system to the point of operation where this value is equal to the number of copies injected at the beginning of the round. To do this, we rely on control theory, which provides the theoretical basis for monitoring a given variable (the *output signal* in control theory terminology) and driving it to some desired value (the *reference signal*).

Following a control theoretic design, we propose the system depicted in Fig. 5.4. This system is composed from a controller $C(z)$, which is the adaptive algorithm that controls the chunk delivery, and the controlled system $H(z)$, which represents the HYPE network. Furthermore, the component z^{-1} provides the delay in the feedback-loop (to account for the fact that the d value used in the current round is computed from the behavior observed in the previous round). For the controller, we have decided to use a Proportional-Integral (PI), because of its simplicity and the fact that it guarantees zero error in the steady-state. The z transform of the PI controller is given by:

$$C(z) = K_p + \frac{K_i}{z-1} \tag{5.8}$$

where K_p and K_i are the parameters of the controller.

Here, the variable that we want to optimally adjust is the number of deliveries at the beginning of the round (i.e., d). Following classical control theory [10], this variable is the *control signal* provided by the controller. In each round, the controller monitors the system behavior (and in particular the output signal, which we will define later), given the value d that is currently used.

Based on this behavior, it decides whether to increase or decrease d in the next round, in order to drive the output signal to the reference signal.

A key aspect of the system design is the definition of the output and reference signals. On the one hand, we need to enforce that by driving the output signal to the reference signal, we bring the system to the optimal point of operation. On the other hand, we also need to ensure that the reference signal is a constant value that does not depend on variable parameters, such as the number of nodes or the contact rates.

Following the arguments exposed in Section 5.2.3, we design the output signal $O(t)$ and the reference signal R of our controller as follows:

$$\begin{cases} O(t) = s(t) - d(t) \\ R = 0 \end{cases} \tag{5.9}$$

where $d(t)$ is the number of deliveries at the beginning of a given round t, and $s(t)$ is the number of signals received at the end of this round. Note that, with the above output and reference signals, by driving $O(t)$ to R we bring the system to the point of operation given by $s = d$, which, as discussed previously, corresponds to the optimal point of operation. Following classical control theory, we represent the randomness of the system by adding some noise $W(t)$ to the output signal, as shown by Fig. 5.4.

5.2.5. Control theoretic analysis

The behavior of the proposed system (in terms of stability and response time) depends on the parameters of the controller $C(z)$, namely K_p and K_i. In the following, we conduct a control theoretic analysis of the system and, based on this analysis, calculate the setting of these parameters. Note that this analysis guarantees that the algorithm quickly converges to the desired point of operation and remains stable at that point.

In order to analyze our system from a control theoretic standpoint, we need to characterize the HYPE network with a transfer function $H(z)$ that takes d as input and provides $s - d$ as output. In order to derive $H(z)$, we proceed as follows. According to the definition given in Proposition 5.2.1, G_d is the gain resulting from sending an extra copy of the chunk. In one round, by sending one extra copy of the chunk at the beginning, there are on average s/d additional nodes that have the chunk at the end. Indeed, s is the total number of nodes that receive the chunk from only one of the d initial seed nodes, which means that on average each seed node contributes with s/d to this number. This yields to:

$$G_d = s/d - 1, \tag{5.10}$$

from which:

$$s - d = G_d d. \tag{5.11}$$

The above provides a nonlinear relationship between d and $s - d$, since G_d (given by Eq. (5.7)) is a non-linear function of d. To express this relationship as a transfer function $H(z)$, we linearize it at the optimal point of operation.[8] Then, we study the linearized model and ensure its stability through appropriate choice of parameters. Note that the stability of the linearized model guarantees that our system is locally stable.[9]

To obtain the linearized model, we approximate the perturbations suffered by $s - d$ at the optimal point of operation, $\Delta(s - d)$, as a linear function of the perturbations suffered by d, Δd,

$$\Delta(s - d) \approx \frac{\partial(s - d)}{\partial d}\Delta d, \tag{5.12}$$

which gives the following transfer function for the linearized system:

$$H(z) = \frac{\partial(s - d)}{\partial d}. \tag{5.13}$$

Combining the above with Eq. (5.11), we obtain the following expression for $H(z)$:

$$H(z) = \frac{\partial(s - d)}{\partial d} = G_d + d\frac{\partial G_d}{\partial d}. \tag{5.14}$$

Evaluating $H(z)$ at the optimal point of operation ($G_d = 0$) yields:

$$H(z) = d\frac{\partial G_d}{\partial d}. \tag{5.15}$$

To calculate the above derivative, we approximate λ_i (given by $\lambda_i = i(N - i)\mu_\beta$) by its first order Taylor polynomial evaluated at level $i = \hat{d}$, where $\hat{d}$ is the average value of i at time T_c (i.e., the average number of nodes that have the chunk at the deadline). Since the Taylor polynomial provides an accurate approximation for small perturbations around $\hat{d}$, and the number of nodes that have the chunk at time T_c is distributed around this value, we argue that this approximation leads to accurate results. The first order Taylor polynomial for λ_i at $i = \hat{d}$ is:

$$\lambda_i \approx \lambda_{\hat{d}} - (i - \hat{d})(2\hat{d} - N)\mu_\beta. \tag{5.16}$$

[8] This linearization provides a good approximation of the behavior of the system when it suffers small perturbations around the stable point of operation [49]. Note that the approximation only affects the transient analysis and not the analysis of the stable point of operation at which the system is brought by the algorithm.

[9] A similar approach was used in [43] to analyze the Random Early Detection (RED) scheme from a control theoretic standpoint.

Substituting this into Eq. (5.7) yields

$$G_d = \frac{1}{\lambda_d}\sum_{i=1}^{N} p_i^d(T_c)\left(\lambda_{\hat{d}} - (i-\hat{d})(2\hat{d}-N)\lambda\right) - 1$$
$$= \frac{\lambda_{\hat{d}}}{\lambda_d} - 1 = \frac{\hat{d}(N-\hat{d})\mu_\beta}{d(N-d)\mu_\beta} - 1 \tag{5.17}$$

Since at the optimal point of operation we have $G_d = 0$, this implies that (at this operating point) $d = \hat{d}$. Moreover, from Theorem 5.2.2 we have that, when operating at the optimal point, if we deliver one additional copy at the beginning (i.e., increase d by one unit), this leads to one additional node with the chunk at the end (i.e., $\hat{d}$ also increases by one unit). Therefore, at the optimal operating point we also have $\partial\hat{d}/\partial d = 1$. Accounting for all of this when performing the partial derivative of G_d yields:

$$\frac{\partial G_d}{\partial d} = \frac{2(2d-N)}{d(N-d)}, \tag{5.18}$$

from which:

$$H(z) = d\frac{\partial G_d}{\partial d} = -\frac{2(N-2d)}{N-d}. \tag{5.19}$$

Having obtained the transfer function of our HYPE network, we finally address the configuration of the controller parameters K_p and K_i, that will ensure a good trade-off between our system's stability and response time. To this end, we apply the Ziegler-Nichols rules [34], which have been designed for this purpose. According to these rules, we first obtain the K_p value that leads to instability when $K_i = 0$; this value is denoted by K_u. We also calculate the oscillation time T_i under these conditions. Once the K_u and T_i values have been derived, K_p and K_i are configured as follows:

$$K_p = 0.4K_u, \;\; K_i = \frac{K_p}{0.85T_i}. \tag{5.20}$$

Let us start by computing K_u, i.e., the K_p value that ensures stability when $K_i = 0$. From control theory [10], we have that the system is stable as long as the absolute value of the closed-loop gain is smaller than 1. The closed-loop transfer function $T(z)$ of the system depicted in Fig. 5.4 is given by:

$$T(z) = \frac{-H(z)C(z)}{1 - z^{-1}H(z)C(z)}. \tag{5.21}$$

To ensure that the closed-loop gain of the above transfer function is smaller than 1, we need to impose $|H(z)C(z)| < 1$. Doing this for $K_i = 0$ yields:

$$|H(z)C(z)| = \left|-\frac{2(N-2d)}{N-d}K_p\right| < 1. \tag{5.22}$$

The above inequality gives the following upper bound for K_p, at which the system turns unstable:

$$K_p < \frac{N-d}{2(N-2d)}. \tag{5.23}$$

We want to ensure that the system is stable independently of N and d, that is, the above inequality holds for any N and d values. Since the smallest possible value that the right-hand side of Eq. (5.23) can take is $1/2$ (when $d \to 0$), the system is guaranteed to be stable as long as $K_p < 1/2$, and may turn unstable when K_p exceeds this value. Accordingly, we set $K_u = 1/2$. Furthermore, when the system becomes unstable, the control signal d may change its sign up to every round, yielding an oscillation period of two rounds, which gives $T_i = 2$. With these K_u and T_i values, we set K_p and K_i following Eq. (5.20),

$$K_p = \frac{0.4}{2}, \;\; K_i = \frac{0.4}{2 \cdot 2 \cdot 0.85}, \tag{5.24}$$

which terminates the configuration of the PI controller.

While the Ziegler-Nichols rules aim at providing a good trade-off between stability and response time, they are heuristic in nature and thus do not guarantee the stability of the system. The following theorem proves that the system is stable with the proposed configuration.

Theorem 5.2.3. *The HYPE control system is stable for* $K_p = 0.2$ *and* $K_i = 0.4/3.4$.

5.3. Performance Evaluation

In this section, we evaluate HYPE for a wide range of scenarios, including several instances of a heterogeneous mobility model, as well as real-world mobility traces. We show that:

- The analytical model provides very accurate results.
- The optimal strategy for data chunk delivery effectively minimizes the load incurred in the cellular network.
- The proposed adaptive algorithm is stable and quickly converges to optimal operation.
- HYPE outperforms previously proposed heuristics in terms of the cellular load, signaling load and fairness among users.

From the four design objectives introduced in Section 5.1.1, our evaluation focuses on the traffic offload, fairness and signaling overhead. Note that, since the delay guarantees are satisfied by design, we meet the objective on the delay.

Simulation setting. To evaluate the performance of HYPE, we use both real mobility traces and a heterogeneous mobility model. For the evaluation with real mobility traces, we select the contact traces collected in the Haggle project for 4 days during Infocom 2006 [24], and the GPS location traces of San Francisco taxicabs,[10] collected through the Cabspotting project [77]. The number of users for the Infocom 2006 and San Francisco traces are 78 and 536, respectively.

[10] We assume two taxicabs are in contact when they are within 100 meters of each other.

As for the heterogeneous mobility model, we generate contacts as follows. For any given node pair (x, y), the pairwise inter-contact times are exponentially distributed with rate β_{xy}. The pairwise contact rates, β_{xy}, are drawn from a Pareto distribution[11] with mean μ_β (which determines the average frequency of the contacts) and standard deviation σ (which determines the level of heterogeneity). To account for sparser scenarios, we also run some experiments where a node pair has a probability $p > 0$ of never meeting, i.e., $\beta_{xy} = 0$ (otherwise the inter-contact rate for the pair β_{xy} is drawn as above). In addition, we generate contact durations δ from a Pareto distribution with parameter $\alpha = 2$, as observed in [23]. Following the findings in [29], we choose the average contact rate μ_β and the average contact duration $\mathbb{E}[\delta]$ values such that $1/(\mu_\beta \cdot \mathbb{E}[\delta])$ is between 100 and 1000.

In all the simulations, we set the throughput of the cellular communication to one mobile node equal to 600 kb/s [11] and the bandwidth of opportunistic communication to 20 Mb/s. All the results given in this section are provided with 95% confidence intervals below 0.1%.

Baseline scenarios. For the heterogeneous mobility model, we use the following four baseline scenarios:

- **streaming**: $N = 100$, mean contact rate μ_β = 13 contacts/pair/day [24] and $\sigma = 0.58 \cdot \mu_\beta$, Pareto-distributed contact duration $\mathbb{E}[\delta]$=66.46 s, T_c = 120 s [82] and chunk size $L = 1$ MB,

- **road traffic update**: $N = 1000$, mean contact rate μ_β = 1.2 contacts/pair/day and $\sigma = 1.5 \cdot \mu_\beta$, Pareto-distributed contact duration $\mathbb{E}[\delta]$ = 72 s, T_c = 600 s, $L = 1$ MB [96],

- **news feed**: $N = 100$, mean contact rate μ_β = 0.69 contacts/pair/day [24] and $\sigma = 2 \cdot \mu_\beta$, Pareto-distributed contact duration $\mathbb{E}[\delta]$= 125 s, T_c = 3600 s [82], $L = 0.5$ MB,

- **social data**: $N = 50$, mean contact rate μ_β = 3.5 contacts/pair/day [24] and $\sigma = \mu_\beta$, Pareto-distributed contact duration $\mathbb{E}[\delta]$= 164 s, T_c = 900 s, $L = 4$ KB.

5.3.1. Validation of the model

In order to validate the analysis conducted in Section 5.1, we evaluate the total load incurred in the cellular network (D) as a function of the strategy followed (which is given by the number of copies of the data chunk delivered at the beginning of a round, d). The results obtained are depicted in Fig. 5.5 for a scenario with $N = 200$, $\sigma = 0.04$ contacts/pair/day, and different values of T_c (in seconds) and μ_β (in contacts/pair/day). We observe that the analytical results follow very closely those resulting from simulations, which validates the accuracy of our analysis. We further observe that, as pointed out in Section 5.2, performance degrades for smaller and larger values of d, since when either too few or too many nodes have the content, information spreads

[11] Under these conditions, the tail of the aggregate inter-contact times decays as a power law with exponential cut-off [74], as observed in traces, in [53].

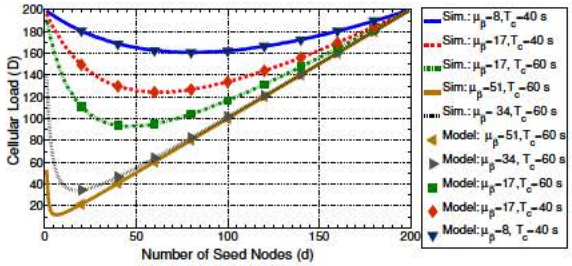


Figure 5.5: The analytical model provides very accurate results for different settings (μ_β is given in contacts/pair/day).

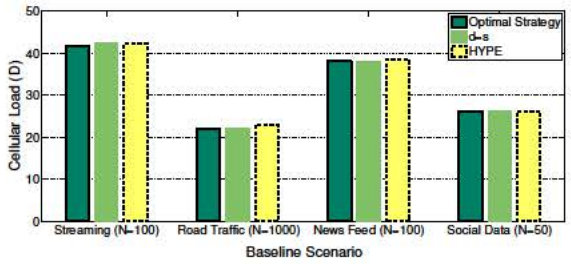


Figure 5.6: Validation of the optimal strategy for the four baseline scenarios.

slowly. The figure finally shows that – given $\mu_b = 17$ – a smaller T_c (of 40 s) causes a higher load of cellular network than a larger one (of 60 s).

5.3.2. Performance gain and validation of the optimal strategy

We next evaluate the performance gains that can be achieved by opportunistic communications in the four baseline scenarios identified earlier and validate the optimal strategy to confirm that it achieves the highest possible gains. Fig. 5.6 gives the performance obtained for the four baseline scenarios with:

- the optimal d value provided by Theorem 5.2.2, labeled *Optimal Strategy*,
- the strategy proposed in Section 5.2.3 for the design of the adaptive algorithm, labeled $d = s$,
- the adaptive algorithm implemented by HYPE, labeled *HYPE*.

For each strategy, the figure shows the absolute average load of the cellular network in number of chunk copies per round (D).

The results obtained show that the proposed approach can reduce very substantially the load of the cellular network (with offloaded traffic ranging from almost 50% in the social data scenario to more than 95% in the road traffic one). The tests also show that the adaptive algorithm implemented by HYPE is very effective in minimizing this load, as it performs practically as the benchmarks given by the optimal and $d = s$ strategies.

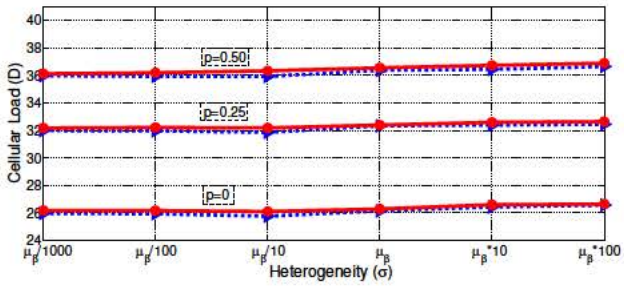


Figure 5.7: Cellular load D as a function of the level of heterogeneity (σ) and network sparsity (p).

5.3.3. Impact of heterogeneity and sparsity

To understand the impact of the heterogeneity of pairwise contact rates β_{xy} on the proposed approach, Fig. 5.7 depicts the total cellular load D for the streaming scenario, with varying σ's. The effect of network sparsity is also shown by using different values for the the probability p that a pair of nodes never meet.

We note that HYPE achieves a performance very close to the optimal, which confirms that the HYPE design also works for heterogeneous settings, as well as sparse ones. In the sparsest tested scenario ($p = 0.50$), D increases by $\approx 38\%$ as compared to $p = 0$, as a result of the slower dissemination caused by the decreasing number of connections (i.e., larger p). Furthermore, for all tested p values, the cellular load D is mostly insensitive to variations of σ both for HYPE and the optimal strategy, which is in line with Theorem 1.

5.3.4. Stability and response time

Based on the control theoretic analysis conducted in Section 5.2, the parameters K_p, K_i of the PI controller have been chosen to guarantee stability and ensure a good response time. In order to assess the effectiveness of this configuration, we evaluate its performance for the streaming baseline scenario and compare it against different choices for the values of parameters K_p, K_i. In Fig. 5.8, we show the evolution of the control signal d over time for our setting $K_p = 0.2, K_i = 0.1176$, as well as a setting of these parameters ten times larger, labeled $[K_p, K_i] \times 10$ and ten times smaller, labeled $[K_p, K_i]/10$. In the test, μ_β increases from 13 contacts/pair/day to 40 contacts/pair/day after 250 rounds. (For instance, this could be the result of an increase in the number of contacts at rush hour). Results show that our setting is stable and reacts quickly, while a larger setting of K_p, K_i is highly unstable and a smaller setting reacts very slowly. This confirms the choice of parameters made for our controller. We also conducted a similar experiment in which we varied N (which could be for instance the result of a change in content popularity) and observed a similar behavior (not shown in the figure for space reasons).

The above experiment shows the response of the controller to a drastic change. In order to confirm that this response is sufficiently quick to follow the variations of the opportunistic contacts in a realistic environment, we consider the San Francisco real traces and study the temporal

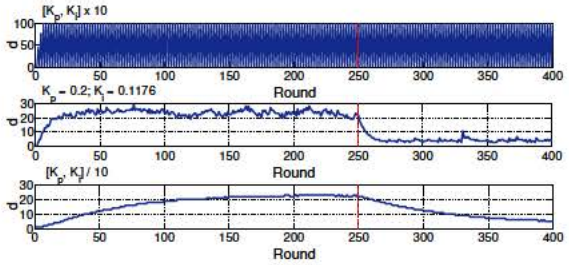


Figure 5.8: Evolution of the control signal d over time for different K_p, K_i settings. Our selection of parameters is stable and reacts quickly.

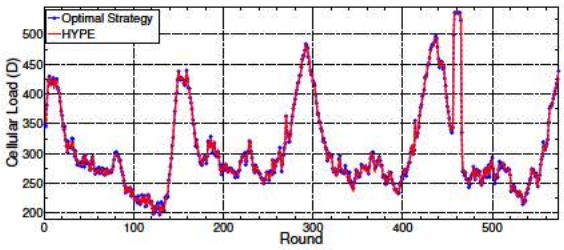


Figure 5.9: San Francisco real traces ($N = 536$): Temporal evolution of the cellular load D for HYPE and optimal strategy, with deadline $T_c = 600$ sec.

evolution of the cellular load D. To provide a benchmark, we compare HYPE against an optimal strategy that selects the best d value every ten rounds.[12] The results, for a content deadline $T_c = 600$ s, are plotted in Fig. 5.9. These results confirm that HYPE reacts rapidly to dynamic conditions: as there are fewer number of contacts during night time, HYPE needs to inject more content through the cellular network (up to $D \approx N$), while the higher number of contacts during day time greatly reduces the network load ($D \approx 220$).

5.3.5. When to deliver: HYPE strategy versus other approaches

One of our key findings in Section 5.2 is that performance is optimized when all the deliveries over the cellular network take place at the beginning and at the end of the period. To validate this result, Fig. 5.10 compares the performance of HYPE against the Push-and-Track heuristics proposed in [96] (namely, *Sqrt*, *Linear* and *Quadratic*), which distribute the deliveries along the period. Results are given for the social data scenario with a varying number of subscribed users N. We observe from the figure that HYPE substantially outperforms all other approaches (the cellular load is even halved, in some cases), and performs very closely to the *Optimal Strategy* benchmark.

In addition to the above experiment, conducted with a mobility model, we also compare the performance of HYPE against the other approaches with real mobility traces. The results, depicted in Fig. 5.11, show that HYPE closely follows the performance of the benchmark given

[12] For the optimal strategy, we make an exhaustive search over all possible d values every ten rounds and select the best one. Note that such a strategy cannot be used in practice and is only considered for comparison purposes.

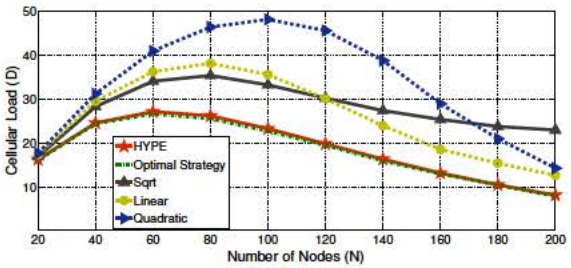
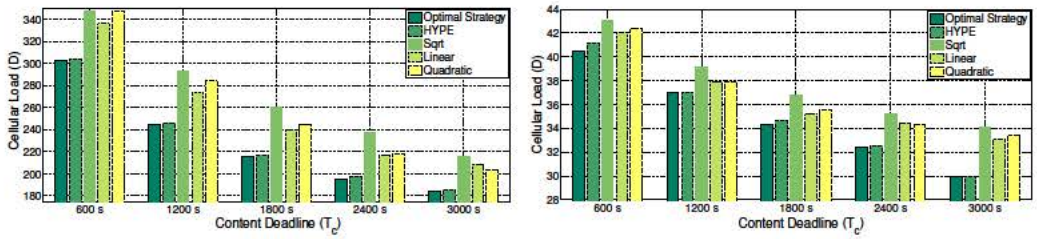


Figure 5.10: Comparison with Push-and-Track heuristics [96].



(a) San Francisco real traces ($N = 536$)

(b) Infocom 2006 real traces ($N = 78$)

Figure 5.11: Tests using real mobility traces for different deadlines T_c. HYPE performs closely to the benchmark provided by the optimal strategy and substantially outperforms previous heuristics.

by the optimal strategy and outperforms previous heuristics. For the San Francisco real traces, HYPE can offload about 20% more traffic than the previous heuristics. For the Infocom 2006 traces, the employed strategy has a smaller impact on cellular load performance, which yields to a smaller gain (up to about 12%).[13]

5.3.6. Which seed nodes: comparison to other selection methods

One of the key decisions in the HYPE design is to randomly select a node when transmitting content over the cellular network. In order to gain insight into the impact of this design decision, we compare HYPE against the heuristic approach proposed in [40] to select the seed nodes in the opportunistic network. Unlike HYPE, [40] requires full knowledge of the pairwise contact rates to identify the target set of users, which involves a much higher level of complexity. Note that, since [40] does not provide an algorithm to compute the number of copies d of the data chunk, we apply the HYPE strategy to compute d also for [40].

Fig. 5.12 shows the performance of both approaches in terms of cellular traffic load (D) and fairness for different values of heterogeneity σ, for the social data scenario. To measure fairness, we apply the Jain's Fairness Index (JFI) to the total number of cellular and opportunistic communications involving a node.[14] The results show that HYPE provides a much higher level of

[13] Indeed, by conducting experiments with the Infocom 2006 traces for many different strategies (unreported here for space reasons), we observed that performance was relatively similar for all of them, which shows that the impact of the specific strategy followed is limited for this case.

[14] For instance, a node that

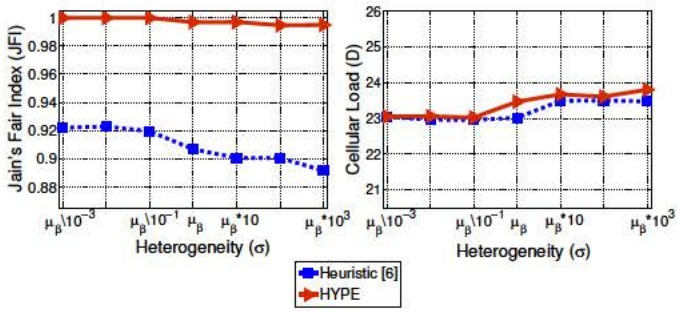


Figure 5.12: HYPE versus the heuristic solution of [40], varying the heterogeneity (σ). HYPE provides a much better trade-off between fairness and cellular load performance.

fairness than [40] with negligible loss in terms of cellular traffic load. Therefore, HYPE does not only feature a simpler implementation than [40], as it does not need to know the individual inter-contact rates, but also provides a much better trade-off between fairness and cellular load performance. The results of this and the previous section are particularly relevant as the algorithms of [96] and [40] are the only existing proposals in the literature to offload cellular networks with opportunistic communications while providing deterministic delay guarantees.

5.3.7. Signaling load

In order to gain insight into the scalability of our design, we analyze the number of uplink signaling messages sent over cellular network. In HYPE, such messages are the signals sent by the nodes with a single copy *ID* to the Content Server at the end of the period. Fig. 5.13 shows the signaling load of HYPE as a function of the number of users for each of the four baseline scenarios, and compares it with the Push-and-Track heuristics (labeled as *Heuristics [96]* in the figure), which require an uplink signal each time a node receives the chunk.[15] In contrast, HYPE scales very efficiently with the number of users: the more users are subscribed to the content, the lower the signaling load per user.

Summarizing the results of the performance evaluation conducted in this section, we have shown that our analytical model is very accurate, that the optimal strategy proposed does indeed minimize the load incurred in the cellular network, that the adaptive algorithm is stable and quick to converge to optimality and that HYPE outperforms existing heuristics in crucial aspects: cellular load, signaling load and fairness among users.

- receives the content through the cellular network and
- sends it to n nodes in range during the period, will have a total number of communications equal to $n+1$

.

[15] We have not compared the signaling messages of HYPE against the approach of [40] since that approach requires gathering data from nodes' mobility patterns. Even though [40] does not explain the signaling mechanism employed, we expect that the need to collect the mobility patterns involves a substantially higher signaling overhead than [96].

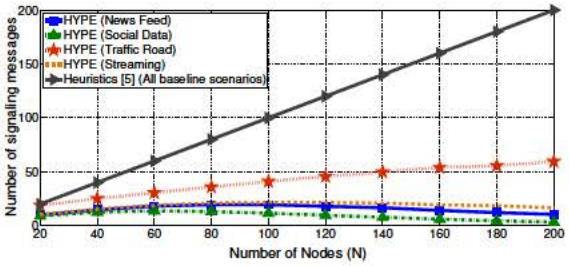


Figure 5.13: HYPE signaling load as a function of the number of nodes N for the four baseline scenarios. HYPE outperforms the previous heuristics of [96] and scales very efficiently with the number of nodes.

5.4. Conclusions

In this chapter, we have presented HYPE, a novel approach to offload cellular traffic through opportunistic communications. To design HYPE, we have developed a theoretical model to analyze the performance of opportunistic dissemination when data can be selectively injected through a cellular network. Based on this model, we have derived the optimal strategy that minimizes the total amount of data injected through the cellular network while meeting delay guarantees. To implement the optimal strategy obtained from the analysis, HYPE runs an adaptive algorithm that adjusts the data delivery over the cellular network to the current network conditions. By building on control theory, we guarantee that this algorithm is stable and quickly adapts to dynamic conditions. The algorithm incurs very low signaling overhead and does not need to monitor the contacts between nodes nor to gather complex statistics, which are important requirements for a practical implementation.

Chapter 6

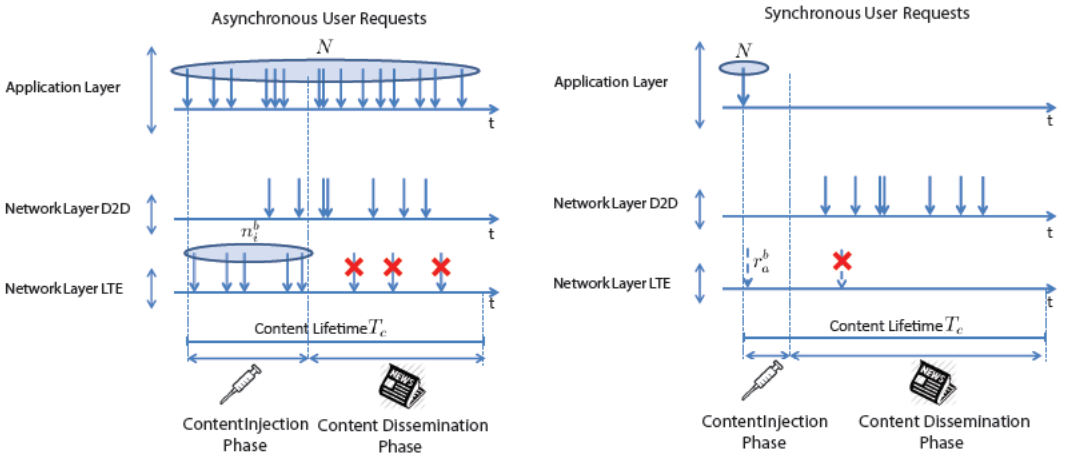
Guaranteeing D2D traffic offloading limiting Intercell Interference

Content dissemination techniques through D2D opportunistic communication for delay guaranteed networks have been extensively studied in the previous chapter. While the majority of those works focuses on the epidemic diffusion, none of them proposes a practical solution for multi-cellular networks. In particular, in this chapter we focus on how inter-cell interference coordination affects the overall content dissemination process. When scheduling the transmission of content updates at base stations, our main objective is to minimize the time required for these transmissions, since (i) the faster contents are injected, the sooner they can be disseminated, and thus D2D-based offloading performance is optimized; and (ii) the less time required for the transmissions, the more resources are freed for other applications. We address this problem by means of the ABSF paradigm. Therefore, we prove that the problem of finding an ABSF-based scheduling algorithm that minimises the time required for content update transmissions while satisfying the content deadlines is NP-Complete and NP-Hard to approximate. Thus, we design BSB, an algorithm that runs in polynomial time, achieves sub-optimal network performance and outperforms the state of the art mechanisms proposed in the literature.
The main aspect of this chapter can be summarized as follows:

- we formulate a base station scheduling problem and we show that it is NP-Complete;
- we design and validate a practical algorithm for the computation of ABSF patterns;
- we show that channel-opportunistic D2D schemes are seriously impaired by non-ideal content injection.

6.1. D2D-assisted content update distribution

In this section, we give a complete overview about the framework of our system and its building blocks. In addition, we provide a real scenario where our solution handily applies.



(a) BS performs n_c^b unicast transmissions.

(b) BS performs a multicast transmission at rate r_a^b.

Figure 6.1: Content Update Transmission Process for N users interested in the content c. On the left side, N users get interested in a content c at different points in time. BS serves the first n_c^b asynchronous content requests in the Content Injection Phase. Then, in the Content Dissemination Phase, users opportunistically exchange the content via D2D technologies. On the right side, we can see a particular case where BS performs a multicast transmission to a multicast group interested in the same content.

6.1.1. Content distribution scenario

We address a LTE-A cellular scenario where N base stations are placed, each of which covers a user set $\mathcal{U}_b$, where b is the base station index. Each user subscribes a content $c \in \mathcal{C}$, with content length L_c and a deadline T_c by which the content needs to reach all subscribed users. Note that multiple users can request the same content.

An example of this scenario is the one envisaged in [60], in which users are moving in a vehicular scenario and a new available road traffic update is considered as content c. Clearly, the content update needs to be delivered to the cars in sufficiently short time in order to be still useful to the users. To the aim of distributing a content to multiple users, while offloading the base station as much as possible, we exploit D2D technology communication: a local controller is installed on each base station, which is in charge of deciding the subset of the initial users receiving the content update. Upon the users retrieve the content, opportunistically share the content (or part of it) with other users via short-range communication technologies such as WiFi-Direct, WiFi or Bluetooth [9].

In this context, the main objective is to design a strategy to deliver content to users that minimises the total resources required by base stations, as this frees resources that can be used by other applications. An additional benefit of our approach is that power consumption of base stations decreases, since it depends on the total activity time of base stations. In order to achieve the objective, we need to address the following two challenges:

Intra-BS optimization: based on the user mobility parameters and the feedbacks received, a controller needs to select the optimal set of users in a cell that receive the content only from the base station, to ensure that (i) the content reaches as much as possible subscribers in the cell by the deadline; and (ii) resources required from the base stations are minimised.

Inter-BS optimization: in addition to determining the transmissions that need to be performed by each base station, we also need to schedule each of these transmissions among base stations, taking into account the interference between base stations in such a way that the total time required by these transmissions is minimised.

In the following, we present the mechanism that we use for the intra-BS part, and the assumptions for the inter-BS optimization problem that we tackle in this chapter.

6.1.2. Intra-BS content distribution

The content distribution process for a particular content may be divided into two phases: (i) *content injection* and (ii) *content dissemination*, hereafter described in details. Users placed under the coverage of base station b get interested in content c randomly, according to a normal distribution with average μ. Content validity period lasts T_c seconds and users may get interested only in a valid content c. We assume that the maximum number of interested users is equal to N, corresponding to the popularity index of the content c [25]. For the sake of simplicity, we suppose the same popularity index N for every content provided in the network.[1]

In the first phase, namely *content injection*, base stations transmit unicastly contents to each interested user asking for those updates. Specifically, the BS controller properly decides n_c^b the maximum number of unicast transmissions per content c BS b can perform. Then, the phase ends when exactly n_c^b interested users, called *injected users*, have received the content directly from base station b, e.g., upon n_c^b users get interested in the content. In the second phase, namely *content dissemination*, the content is spread opportunistically into the network via D2D technologies to those users which could not download the content directly from the base station. Although the two phases may overlap, this does not affect our analysis as the total time spent to deliver content replica to interested users does not change, as already proven in [4].

The number n_c^b of injected users plays a key-role in driving the system to an efficient working point. On the one hand, the more the number of injected users, the more the time required by the base station to perform the content update transmission. On the other hand, if the number of injected users n_c^b has not been designed properly, most of the users asking for the content update will not be reached within the content lifetime (T_c).

Therefore, we introduce a bi-dimensional Markov chain, where each state $S_j(t)$ is the total *number of content replica* distributed in the network at time t given j users interested in the content, regardless of the specific users carrying those replica. Assuming a homogeneous mobility model where users get in touch each other following an average inter-contact rate λ and getting

[1] Nonetheless, we can readily derive equivalent results for heterogeneous content population indexes N_c depending on the content c.

interested in a content according to an average rate μ, we obtain the average number of users with content at the end of the content lifetime T_c as follows

$$\mathbb{E}[S_j(T_c)] = \sum_{x=1}^{N} x\, p_x(T_c - d_{in}). \tag{6.1}$$

where $p_x(t)$ is the probability to stay in the state x at time t. For further details we refer the reader to Appendix.

Eq. (6.1) provides a function returning the average number of users with the content after the content lifetime expiration (T_c), based on the number of injected nodes (n_c^b) and the number of interested users in that content j. Based on such information, BS b decides the number of injected nodes n_c^b per content c by solving the following optimization problem:

Problem `Offloading`:

$$\begin{aligned} \text{maximise} \quad & \sum_{c=1}^{|\mathbb{C}|} \log(\eta_c), \\ \text{subject to} \quad & \sum_{c=1}^{|\mathbb{C}|} \tfrac{n_c^b\, L_c}{T_c} \leq \alpha C_b; \\ & n_c^b \in \{1...N\}; \end{aligned}$$

where the content throughput is defined as $\eta_c = L_c\, \frac{\mathbb{E}[S_j(T_c)]}{T_c}$, while αC_b identifies the available resources at the base station side. In other words, base station b finds the optimal n_c^b per content to ensure that the base station capacity constraint is fulfilled. We will further explore how to free resources for other applications.

The optimization of Problem `Offloading` can be easily linearised and solved by means of a commercial solver. Moreover, due to scalability issue, very large instances of the problem can be approached through a simple heuristic, providing an affordable trade-off between accuracy and complexity. Specifically, to linearize Problem `Offloading` we sample the logarithmic function into a limited number of values, as only a discrete set of n_c^b values are considered for the optimization. We obtain a matrix $\zeta = \{\zeta_{i,n}\}$ of $[|\mathbb{C}| \times |\mathbb{N}|]$ size, where $\zeta_{c,n} = \log(\eta_c)$, with $n = n_c^b$. Therefore, assuming the same content length L and lifetime T, $\forall c \in \mathbb{C}$, we can rewrite Problem `Offloading` as follows:

Problem `Offloading-Lin`:

$$\begin{aligned} \text{maximise} \quad & \sum_{c=1}^{|\mathbb{C}|} \sum_{n=1}^{|\mathbb{N}|} s_{c,n}\, \zeta_{c,n}, \\ \text{subject to} \quad & \sum_{c=1}^{|\mathbb{C}|} \sum_{n=1}^{|\mathbb{N}|} s_{c,n} \cdot n \leq K; \\ & \sum_{n=1}^{|\mathbb{N}|} s_{c,n} \leq 1, \forall c \in \mathbb{C}; \\ & s_{c,n} \in \{0,1\}; \end{aligned}$$

where $K = \alpha\, C_b \frac{T}{L}$, while $s_{c,n}$ is a binary value indicating with 1 whether n nodes are initially injected with content c, or 0 otherwise. In other words, we aim at choosing the optimal set of injected nodes values n_c^b (selecting one value per content), guaranteeing that the capacity constraint of the base station is efficiently fulfilled. When the number of available contents $|\mathbb{C}|$ or the content popularity index $|\mathbb{N}|$ tend to huge numbers, solving this problem may take very long time. Given that Problem `Offloading-Lin` can be easily mapped into a generalized assignment problem, as heuristic to solve the problem we can use an extended version of the Hungarian Algorithm [58] to provide a near-optimal solution in reasonable time.

It is worth noting that the content distribution process can be readily extended to other scenarios, such as synchronous content update subscriptions [5], where user interest rate μ tends to infinite. In such scenarios users covered by base station b subscribe a content update c arranging distinct content interesting groups per cell (multicast groups), as depicted in Fig. 6.1(b). A new content update will be issued every T_c seconds to any multicast group by any base station in the network. Each user subscribes only one single content update and the multicast groups are disjoint. Given that the multicast operation requires a transmission at the least user rate of all multicast receivers in the group [91], for a given multicast rate only a part of the users in the group will be able to decode the message (i.e., those whose channel condition enables them to receive at the chosen rate). Therefore, during the *content injection* phase, upon a new content update is available, the BS controller decides the rate r_c^b at which multicast transmissions must be performed. The *content dissemination* phase starts spreading the content (or part of it) opportunistically in the group to reach those users which have not received the content during the injection phase. Also in this case, the choice of the multicast rate for the initial injection involves the following trade-off: (i) if the selected multicast rate is too low, the number of bits injected will be small and thus efficiency will be low, (ii) however, if the selected rate is too high, the initial injection will only involve few users and hence content is unlikely to spread to all subscribed users by the content lifetime T_c. Therefore, BS b needs to optimally solve Problem `Offloading`, where the number of injected nodes n_c^b is computed as a function of rate r_c^b, as studied in [20].

6.1.3. Inter-BS scheduling

Following the previous explanations, during the *content injection* phase, the content update reaches only n_c^b users. Moreover, such injections cause interference due to the presence of multiple base stations. To address this problem, we adopt the ABSF paradigm, which has been shown to provide improved performance in presence of inter-cell interference [2, 3, 27].

On average, if $\mathcal{C}$ is the population of active contents, a single base station b needs to perform d_b content transmissions, where $d_b = \sum_{c \in \mathcal{C}} n_c^b$. Content requests arrive asynchronously, even though contents are made available at regular intervals T_c, whose duration represents the content's lifetime. In addition, a base station serves all its users with unicast transmissions, applying a scheduler with equal rate, i.e., all users with pending transmissions are scheduled and receive the same data rate on a per-TTI basis. The achievable throughput t_u of each user u in subframe i

depends on its signal-to-noise-ratio:

$$t_u(i) = B_T \log_2 \left(1 + \frac{S_u^b(i)}{N_0 + \sum_{j \neq b} I_u^j(i) x_{ij}} \right) \tag{6.2}$$

where B_T is the used bandwidth, S_u^b is the useful signal received by user u from the serving base station b, N_0 is the background noise, I_u^j is the interference created by the base station j toward user u, and x_{ij} is a binary value which indicates whether the base station j is scheduled in the subframe i. We define $w_u, u \in 1, .., d_b$, as the set of positive coefficients representing the fraction of resources allocated to active user u in a subframe, such that equal rates are achieved:

$$w_p \, t_p = w_q \, t_q, \quad \forall p, q \in \mathcal{U}_b, \tag{6.3}$$

$$\text{subject to:} \sum_{p=1}^{d_b} w_p = 1. \tag{6.4}$$

Therefore, the coefficients w_u can be computed (in each subframe i) as follows:

$$w_u(i) = \frac{\frac{1}{t_u(i)}}{\sum_{k=1}^{d_b} \frac{\delta_{ki}}{t_k(i)}}, \tag{6.5}$$

where δ_{ki} is 1 if transmission k is ongoing in subframe i, and it is 0 otherwise. With the above, the throughput of user u is $w_u(i) \cdot t_u(i)$ in subframe i.

6.2. Base station transmission time minimization

Here, we formulate the inter-BS scheduling introduced before as an optimization problem. We mathematically prove that such problem is NP-Complete and NP-hard to approximate. Therefore, we provide a sufficient condition which must be fulfilled to solve the problem. Based on such condition, we propose an efficient algorithm to automatically generate ABSF patterns, which are promptly issued to every base station in the network. In addition, we also analytically provide a lower bound for the problem solution, which will be extensively analyzed in our performance evaluation. Lastly, we show the maximum number of contents the system is able to work off in a given content deadline.

6.2.1. Problem formulation

The efficiency of the content dissemination depends on the speed of the content injection process, and therefore our goal when designing the inter-BS scheduling is to minimise the time needed to inject the content, as expressed in the following optimization problem:

Problem `BS-Time-Scheduling`

Input:
A collection of N base stations $B = \{1, 2, \cdots, N\}$, and distinct transmission entities[a] $O = \{o_1^b, o_2^b, \cdots, o_{d_b}^b\}$ associated with base station $b \in B$. Positive constants N_0, τ, Θ, L_c, B_T. Integer $Z > 0$. For a generic entity o associated with base station b: $S_o^b(i)$, $w_o(i)$ and $I_o^j(i)$ for every $j \in B \setminus \{b\}$ and every $i = 1, 2, \cdots, Z$.

Question: Is there a scheduling of the base stations in at most Z rounds, such that

$$\Psi_o^b(Z) = \tau B_T \sum_{i=1}^{Z} x_{ib} w_o(i) \log_2\left(1 + \frac{S_o^b(i)}{N_0 + \sum_{j \neq b} I_o^j(i) x_{ij}}\right) \geq L_c,$$

$\forall o \in \{1, .., d_b\},\ b \in \{1, .., N\}$, and

$$\sum_{b=1}^{N} T_{TOT}^b = \tau \sum_{b=1}^{N} \sum_{i=1}^{Z} x_{ib} \leq \Theta \ ?$$

[a] Throughout all the chapter, we refer with term *transmission entity* for both unicast user (u) and multicast group (a), as the same problem formulation can be easily applied to both unicast and multicast transmission types.

In Problem `BS-Time-Scheduling`, each term $T_{TOT}^b = \tau \sum_{i=0}^{Z} x_{ib} = \tau Z_b$ represents the activity time of base station b (τ is the subframe duration). The term $w_o(i)$ is the generic fraction of resources reserved to a transmission entity o in subframe i. Z is the number of subframes that correspond exactly to the content lifetime interval T_c, while Θ is the upper bound for the aggregate transmission time of the system. Transmission rates are computed using Shannon capacity formula.

6.2.2. Complexity of Problem `BS-Time-Scheduling`

Classical wireless scheduling problems, e.g., scheduling and channel assignment, have been shown to be NP-Hard [22, 37]. However, we are the first to address the complexity of base station resource allocation with deadlines and multicast transmissions using variable rates. Specifically, we prove through the following theorems, that problem `BS-Time-Scheduling` is NP-Complete when $Z \geq 3$ for bounded interferences, and for $Z{=}2$ for unbounded interferences. (The formal proofs of the theorems are provided in the Appendix.)

Theorem 6.2.1. *Problem `BS-Time-Scheduling` is NP-Complete, for any $Z \geq 3$, even when all interferences are $\in \{0, 1\}$.*

Theorem 6.2.2. *Problem `BS-Time-Scheduling` is NP-Complete for $Z = 2$.*

These NP-Completeness results apply to very special instances of the problem ($d_b = 1$ for every base station b). When considering the minimization version of the problem (to determine

a scheduling with smallest number of rounds), we use [103], which shows that for all $\epsilon > 0$, approximating the chromatic number of a given graph $G = (V, E), |V| = n$ within $n^{1-\epsilon}$, is NP-hard. Since coloring G with n colors is trivial, this means that this result is rather strong. Using it we show that Problem `BS-Time-Scheduling` is rather difficult to approximate, as follows:

Theorem 6.2.3. *For all $\epsilon > 0$, approximating within $n^{1-\epsilon}$ the minimal number of rounds required to solve Problem `BS-Time-Scheduling` with n base stations is NP-hard.*

6.2.3. Sufficient condition for Problem `BS-Time-Scheduling`

Since, as we have shown above, Problem `BS-Time-Scheduling` is NP-complete and NP-hard to approximate, in the following we provide a sufficient condition that guarantees that the entire content is delivered before its lifetime, i.e., in Z subframes. Specifically, we can derive the following inequality from Eq. (6.5), which holds for any subframe i:

$$w_u(i)t_u(i) = \frac{1}{\sum_{k=1}^{d_b} \frac{\delta_{ki}}{t_k(i)}} \geq \frac{t_{\min}(i)}{d_b}; \tag{6.6}$$

where $t_{\min}(i) = \min_i\{t_u(i)\}$. If we now sum over the subframes in which the user is served within the time horizon Z, we obtain a bound for the volume of traffic V_u received by a user:

$$V_u = \tau \sum_{i=1}^{Z} x_{ib} w_u(i) t_u(i) \geq \tau \sum_{i=1}^{Z} x_{ib} \frac{t_{\min}(i)}{d_b} \geq \tau \frac{Z_b}{d_b} t^*_{\min}, \tag{6.7}$$

where $t^*_{\min} = \min_i\{t_{\min}(i)\} = \min_{u,i}\{t_u(i)\}$ is the minimum instantaneous rate allotted to any user, and $Z_b = \sum_{i=1}^{Z} x_{ib}$ is the number of subframes in which base station b is active. Since it is sufficient to guarantee that $V_u \geq L_c$ to guarantee that user u received the content on time, we obtain the following sufficient condition for the doability of the scheduling:

$$t^*_{\min} \geq \frac{L_c d_b}{Z_b \tau}. \tag{6.8}$$

In conclusion, inverting the Shannon formula from the minimum value for $t^*_{\min}$ given in Eq. (6.8), we deduce that it is sufficient to schedule a base station when all its scheduled transmission entities have at least the following SINR:

$$\text{SINR} \geq 2^{\frac{d_b L_c}{\tau Z_b B_T}} - 1 \doteq \texttt{TH}. \tag{6.9}$$

Note that the above equation defines an SINR threshold `TH` that depends, in addition to some constants, on the number of subframes Z_b in which base station b is allowed to transmit. Next, we derive a lower bound on Z_b for which the inter-BS scheduling guarantees that d_b content injections are doable within the deadline.

6.2.4. Lower bound for Z_b

The throughput of a base station b can be bounded by the following expression:

$$\frac{d_b L_c}{\tau \sum_{o=1}^{d_b} \sum_{i=1}^{Z} w_o(i) x_{ib}} = \frac{d_b L_c}{\tau Z_b} \leq R_{\text{MAX}}, \tag{6.10}$$

where R_{MAX} is the maximum transmission rate permitted in the network (e.g., $R_{\text{MAX}} = 93.24$ Mbps in an FDD LTE-A network using 20 MHz bandwidth). Therefore, there is a lower bound for Z_b below which the content injection of d_b contents cannot be guaranteed:

$$Z_b \geq \frac{d_b L_c}{\tau R_{\text{MAX}}}, \quad \forall b \in B. \tag{6.11}$$

Since we aim to minimise the total transmission time, which is given by $\Theta = \tau \sum_{b \in B} Z_b$, it is reasonable to assume that an ICIC algorithm that approximates the solution of Problem `BS-Time-Scheduling` will be able to complete the injection of d_b contents at base station b in a number of subframes that is close to the bound given above, i.e., $Z_b = \frac{d_b L_c}{\tau R_{\text{MAX}}}$. With this approximation, we can express the threshold `TH` in (6.9) as a function that does not depend on Z_b.

The above provides a sufficient condition to guarantee that d_b contents are delivered within their lifetime; in particular, we have found a threshold `TH` for the SINR of users to be scheduled. In Section 6.3, we leverage this result for the design of our ICIC algorithm.

6.2.5. Maximum number of contents

Before describing our heuristic for Problem `BS-Time-Scheduling` in Section 6.3, we compute the maximum number of content updates that base stations can handle. This result will be useful in Section 6.4 to evaluate eICIC schemes. To achieve our goal, we assume that all the base stations have, at least on average, the same number of contents to inject in interval T_c.

If all base stations have the same number of contents to inject, we can derive an upper bound for Z_b. The total number of subframes used by all base stations cannot exceed $\sum_{b \in B} Z_b = N Z_b$. If Z is the total number of subframes in which the content is valid, we have that $N\, Z_b \leq Z$ and thus, we can derive an upper bound as $Z_b \leq \frac{Z}{N}, \quad \forall b \in B$, which, jointly with (6.11), yields the following range for Z_b:

$$\frac{d_b L_c}{\tau R_{\text{MAX}}} \leq Z_b \leq \frac{Z}{N}, \quad \forall b \in B. \tag{6.12}$$

From the analysis above, we can then compute the maximum number of injectable contents that can be handled by a base station while guaranteeing that all contents are served within the deadline $T_c = \tau Z$. In particular, from (6.12), it is clear that the Z_b range is not empty under the following condition, which gives an upper bound for d_b:

$$d_b \leq d_b^* = \frac{\tau\, Z\, R_{\text{MAX}}}{L_c N}. \tag{6.13}$$

6.3. Base Station Blanking Algorithm

In this section, we propose BSB (Base Stations Blanking), an algorithm to approximate the optimal solution of Problem `BS-Time-Scheduling` formulated in Section 6.2. BSB relies on the sufficient condition given by (6.9). Following this condition, BSB aims to find an optimal ABSF pattern, i.e., an allocation of base stations to subframes, in which the interfere is limited and guarantees a minimum SINR to any mobile device that might be scheduled. Note that our algorithm is meant to allocate ABSF patterns, and does not impose any user scheduling policy.

A schematic view of BSB is reported here. BSB runs in a LTE-A network, and requires the presence of a central controller, namely the Base Stations Coordinator (BSC), which could be run on the Mobility Management EntireE) [90]. Our algorithm requires cooperation between the BSC and base stations, which can be implemented over the standard $X2$ interface [88]. The main role of BSC is to collect SINR statistics from the base stations, run BSB, and announce ABSF patterns to the base stations, as detailed in what follows:

BSB Algorithm

The BSC collects user statistics, puts all active base stations in a candidate set, and checks whether the resulting SINR for each user is above the SINR threshold `TH`.

If at least one user does not reach the SINR threshold:

- compute the most interfering base station b^*
- remove b^* from the candidate set,
- check the SINR of all users of the remaining base stations.

Repeat the check and remove base stations from the candidate set until all remaining users meet the SINR constraint. The resulting set of base stations is scheduled in the first subframe and inserted in a *priority*-1 list. In general, at each subframe, scheduled base stations are added to the *priority*-k list, where k is the current number of subframes enabled for a base station to transmit. All other base stations go to a *priority*-0 list.

For each successive subframe, populate the candidate set with the *priority*-0 list and repeat the operation described for the first subframe until the SINR constraint is met.

Then, for $k = 1, 2, \ldots$, in increasing order:

- add to the candidate set all base stations in the *priority*-k list,
- within *priority*-k list, remove base stations causing SINR below `TH`.

The algorithm stops when the priority list is empty.

The BSC issues the resulting ABSF pattern to each base station via the $X2$ interface.

In the above description, the interference caused by a base station is computed as the aggregate sum of interferences caused towards all users of all other bees station in the candidate set. The threshold TH is computed based on d_b and the lowest possible value for Z_b, given by (6.11). The scheduling pattern computed with BSB can range between 1 and N subframes. However, since the standard specifies that ABSF patterns should be issued every 40 subframes, the BSB pattern is repeated in order to cover a multiple of 40 subframes. The obtained sequence of scheduling patterns represents the ABSF pattern according to [88].

For each subframe, the algorithm starts by selecting the full set of base stations that have not been scheduled in previously allotted subframes. The rationale behind this choice is twofold: (i) the aggregate interference caused by a base station grows with the size of the candidate set, and thus the importance of the interference generated by a base station is more properly quantified by the full candidate set; (ii) existing ICIC algorithms suggest to mitigate interference by preventing the transmission of a few base stations, beginning with the most interfering one [2, 71, 81]. BSB complexity is dominated by the number of base stations, as stated in Theorem 6.3.1.

Theorem 6.3.1. *The complexity of BSB is* $O(U \cdot N^3)$*, where* $U = \max_{b \in B}\{U_b\}$*, and* $N = |B|$.

Sketch of Proof: The BSB algorithm runs in at most N rounds, corresponding to N allocated subframes: in the worst case, exactly one base station is allocated in exactly one subframe. At subframe $q = 1, 2, ..., N$, there are at most q priority lists. In the worst case, the *priority*-0 list contains $N - q + 1$ base stations and each other priority list contains 1 base station. Evaluating the SINR for all users of base stations in *priority*-0 requires checking all reconfigurations with $N - q + 1$, $N - q$, ..., 1 base stations in the candidate set. Checking the possibility to add to the resulting scheduled set any base station in the other priority lists is at most involving $N - q + 2$ base stations for considering *priority*-1 list, $N - q + 3$ for *priority*-2 and so on until N base stations for the last priority list. Overall, the cost per subframe is $O(U \cdot N^2)$. Therefore, in the worst case, in which N subframes are needed, the complexity is $O(U \cdot N^3)$. □

6.4. Performance Evaluation

Here we study the impact of BSB on the performance of D2D-assisted content update distribution. We benchmark the performance achieved with BSB against the one achieved under different frequency reuse schemes (in particular frequency reuse 1, 3, and 5), and against a state-of-the-art dynamic resource allocation scheme proposed for ICIC in LTE-like networks [81]. We refer to the latter as ECE. Differently from BSB, ECE assigns resource blocks rather than subframes, thus implementing a scheme for *soft fractional frequency reuse* [39]. Summarizing, we compare five inter-BS resource allocation mechanisms, as reported in Table 6.1.

As concerns the system parameters adopted in our performance evaluation, we use FDD LTE-A frame specifications, with 20 MHz bandwidth distributed over 100 frequency chunks, resulting in 100 resource blocks per time slot, i.e., 200 resource blocks per LTE-A subframe [89]. Transmission power is fixed to 40 W, antenna gain and path loss are computed according to [59], and the

Table 6.1: Resource allocation mechanisms

BSB	ABSF patterns are computed according to BSB
FR1	No interference coordination is enforced
FR3	Base stations are allocated different frequencies, according to a frequency reuse 3 scheme, and each base station uses $1/3$ of the band available
FR5	Base stations are allocated different frequencies, according to a frequency reuse 5 scheme, and each base station uses $1/5$ of the band available
ECE	LTE-A resource blocks are allocated to base stations according to [81].

spectral noise density is $3.98 \cdot 10^{-21}$ W/Hz for all nodes [92]. Modulations and coding schemes are selected according to the SINR thresholds reported in [89], while the ratio between received power (or interfering signal) and noise, for each user in the network, is computed as for Rayleigh fading, with average computed from transmission power and path loss.

D2D communications occur *outband* (i.e., on a channel not interfering with any of the base stations), and mobile devices exchange data when their distance is 30 m or less. Content updates occur synchronously for any content c, every $T_c = 100$ s. Each mobile device is interested in at most one content (whose size is 8 Mbits). Users get interested in a content at different points in time, according to a truncated normal distribution function having μ as mean value for the interesting rate. For the sake of completeness, we have also conducted simulations to evaluate the case with an infinitive μ corresponding to a content subscription case where base stations inject beforehand the content updates through multicast transmissions. Background traffic is also generated in some of our experiments, consisting in uniformly random file requests, with file size 8 Mbits. Background requests are processes as content updates for single users.

As concerns the mobility of users, we use a Random Waypoint mobility model over a regular grid [100]. Mobile users are initially assigned uniformly over the area, then they choose uniformly random distributed destinations (waypoints P_u), and speeds (V_n) uniformly distributed in range $[1, 2]$ m/s, independently of past and present speed values. Then, the mobile user travels toward the newly chosen destination at constant speed V_n. Upon arrival to destination P_u, the mobile user randomly chooses another destination and speed. Note that, at the considered low speed, the resulting contact time is long (several seconds). Therefore, we assume that complete file transfers are possible during the contact time. This results in a particular contact rate λ.

All experiments refers to a dense LTE-A deployment, with 5 overlapping cells, and several hundreds of mobile users. Each experiment includes 50 content updates for each content, with period 100 s (i.e., the experiment simulates $5\,000$ s), and is repeated 20 times. Average and 95% confidence intervals are reported in the figures. When using BSB, a specific ABSF pattern is issued every 40 subframes, which perfectly complies with 3GPP standard specifications [88].

6.4.1. Injection Phase: empirical validation

The injection phase plays a key-role in driving the content dissemination process to extremely efficient conditions. A wrong decision on the number of injected contents brings the system to a faulty performance efficiency. Therefore, we show how that decision impacts on the system

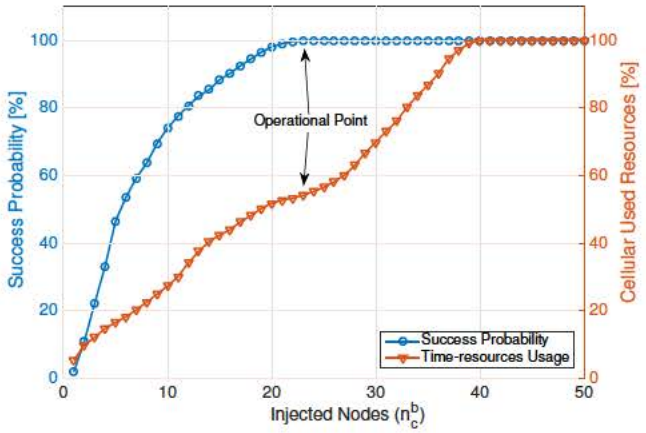


Figure 6.2: Probability of delivering successfully a content for different values of injection nodes in a scenario with 5 BSs and 750 users. Cellular resources used for the injection phase are also show (averaged over the content lifetime $T_c = 100$ s).

performance in terms of probability of successfully content delivering as well as the portion of offloaded base station time-resources. Fig. 6.2 shows the probability to receive the entire content at the end of the content lifetime (T_c) trying out different values of injections (accounting for both LTE-A and D2D transmissions), when 150 users get interested in a content. We applied on top of injection number decision our algorithm BSB to efficiently schedule the BS time-resource to intended users requiring the content. Intuitively, the more injected nodes, the more the probability that a D2D content exchange occurs, the more users will get the entire content at the end of the content lifetime. In contrast, we show the portion of time-resources saved by LTE-A base stations during the content dissemination process. Whenever more than 37 injected contents are required, the system results in a critical time-resources shortage. Also, the graph highlights the operational point of our algorithm derived from Problem `Offloading`, as explained in Section 6.1.2: with 23 content transmissions the system successfully delivers the content to all interested users while significantly limiting the time-resources usage (up to 54%) per base station.

6.4.2. Base station transmission time and delivery success probability

We simulate the network depicted in Fig. 6.3, with 5 base stations and 750 mobile devices. Therefore, in the described results, scheme FR1 represents the case with no ICIC, while FR5 guarantees no interference. Our objective is to analyze in details how scheduling procedures affect the base stations offloading throughout the whole content update process.

For the first set of results, we evaluate the effective amount of time-resources saved by each base station while applying compared scheduling approaches. Fig. 6.4 shows the per-base station average transmission time, expressed in terms of used subframes, when 200 users get interested and require a content each second. No background traffic was injected during the experiment. For the case of ECE, in which resource blocks are allotted rather than subframes, we count the total number of used resource blocks, and normalise that number with respect to the number of resource blocks per subframe. BSB clearly outperforms ECE and FR3 by a factor ~ 3, and up

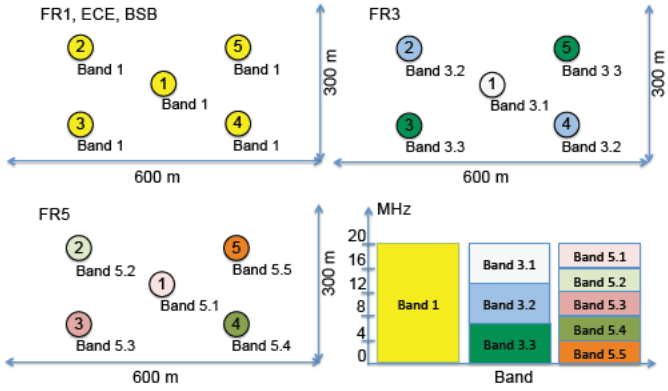


Figure 6.3: Network scenario with 5 base stations placed at regularly spaced positions, and 750 users (not shown in the figure) randomly dropped into an area of 600 m $\times$ 300 m. For each tested scheme, the figure reports the BS baseband bandwidth.

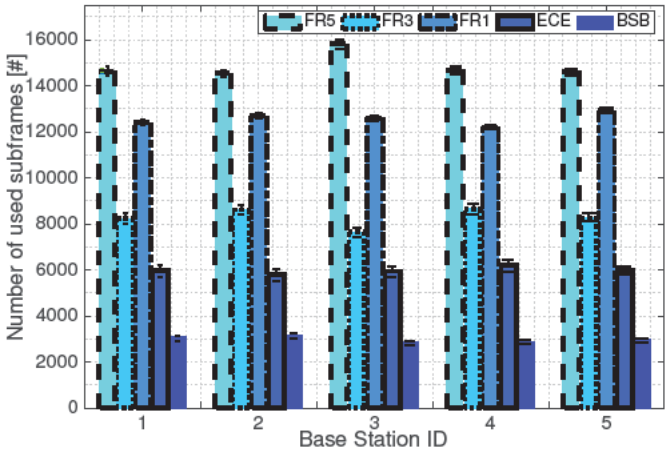


Figure 6.4: Content update transmission time with 5 base stations, 750 users, interesting rate $\mu = 200$, and no background traffic.

to ~ 5 for the case of FR5. Note that, for a fair comparison to BSB and ECE, frequency reuse schemes simulated in the experiment allocate only $1/n$, $n \in \{1, 3, 5\}$ of the available bandwidth to each base station. With the data reported in the figure, it is clear that BSB improves the results of FRn, $n \in \{3, 5\}$, by a factor $\sim n$. Therefore, we could extrapolate that modifying FR3 and FR5 schemes using n times the bandwidth used by BSB would achieve similar results as BSB. Indeed, we have validated such an intuitive result by running an experiment in which all base stations always use the entire 20 MHz bandwidth. Results show negligible performance differences (below 1%) between the schemes. However, we remark that BSB would require $1/n$ of the frequencies needed by frequency reuse schemes.

Fig. 6.5 shows a cumulative distribution function for the successfully delivered portion of each content update, under the tested schemes. BSB exhibits an impressive behaviour compared with the other solutions. Only in 0.3% of the cases BSB fails to start the content delivery, whereas

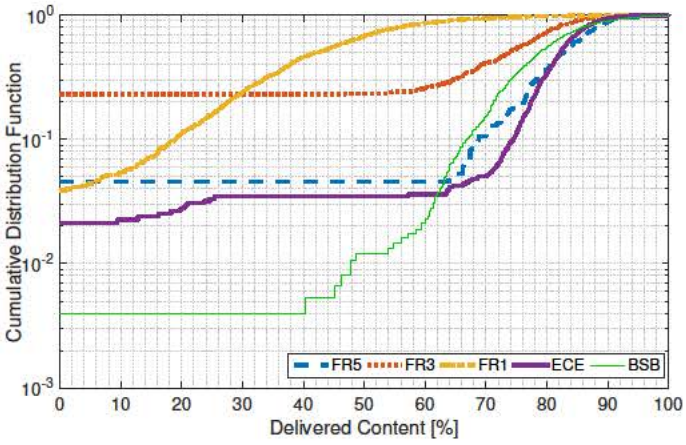


Figure 6.5: Content update success probability with 5 base stations, 750 users, interesting rate $\mu = 200$, and no background traffic.

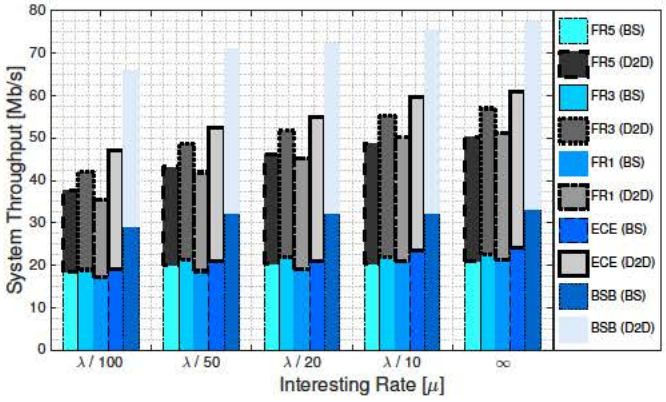


Figure 6.6: System throughput for different interesting rate μ when 5 base stations and 750 users are placed, a meeting rate $\lambda = 2000$ pair/contacts/second is considered, and different scheduling procedures are applied. The last case $\mu = \infty$ provides system performance for content subscription scenario with multicast transmission.

in almost the 99% of the cases BSB delivers at least 50% of the content update. All the other considered schemes show a much higher probability to fail to start the delivery (2% to 22% of cases). In general, FR1 and FR3 perform much worst than the others, as static frequency reuse mechanisms are not able to dynamically follow the network changes resulting in a very high probability to deliver only a few chunks of the content update, whereas FR5 and ECE and BSB manage to reduce interference sensibly and so guarantee high delivery rates, although, as shown in Fig. 6.4, BSB operates the injection much faster.

6.4.3. Throughput of base stations and of D2D exchanges

Fig. 6.6 shows the aggregate system throughput for a 5 base stations scenario where 750 users are placed. Each of proposed scheduling approaches is studied for a particular set of interesting

rates μ (expressed in terms of interested users per second), as function of meeting rate $\lambda = 2000$ pair/contact/seconds[2]. Interestingly, we show the amount of system throughput due to the base station transmissions (both for content injection and for other kinds of traffic) while, on top of the graph, the throughput due to the dissemination phase. We want to point out two main aspects. On the one hand, the faster users get interested in the content, the lower the base station load, the more free time-resources are assigned to other kinds of traffic, the higher the D2D communication throughput. The rationale behind is pretty intuitive. When users express their interest for a content at the beginning of the period T_C, base stations can promptly inject them the content update, leaving more time to the users to spread the content. In this way, much more contacts occur in the network, much more data is exchanged through D2D communication (as also confirmed in Appendix). The extreme case is modelled when $\mu = \infty$, i.e., when all users get interested at the beginning of each period T_C. On the other hand, BSB shows an incremental gain w.r.t. the other presented approaches. For the first set of interesting rates μ, FR5 and FR1 are unable to complete the injection phase, as several transmissions are required (e.g., 97 injections for $\mu = \lambda/100$ and 65 injections for $\mu = \lambda/50$) leaving no room for other traffic. When the required injections decrease to 26 for $\mu = \lambda/10$, all scheduling schemes exhibit the same base station throughput except BSB due to the ability of scheduling other traffic. This confirms that an optimal offloading base stations procedure requires a very fast injection phase, which must be properly designed through a convenient scheduling scheme.

6.4.4. Impact of background traffic

To show the efficacy of BSB in more generic traffic scenarios, in addition to periodic content updates, we next simulate background file requests uniformly distributed over time at different request rates. Note that (6.13) expresses the maximum number of contents that can be distributed with guaranteed maximum transmission time. That expression can be also interpreted as the maximum cell load that can be handled by a base station while guaranteeing that content updates will be delivered within the deadline (with each content unit used for d_b^* corresponding to an offered load $L_c/(\tau Z)$). Therefore, we expect that BSB is able to handle a background traffic equivalent to, at most, $(d_b^* - d_b) \cdot L_c/(\tau Z)$ bps. With 8-Mbit background files, $d_b = 20$, $L_c = 8$ Mbits for any content c, $\tau Z = 100$ s, and 5 base stations, the maximum background traffic is 2.125 requests per second.

In Fig. 6.7, we show the impact of background traffic on the probability to complete the content update distribution, for various background loads. Similarly to the case in which no background traffic is injected, BSB outperforms other schemes. Interestingly, BSB is more robust to background traffic than other schemes, as shown by the fact that content delivery probability under BSB is barely affected by the background traffic. The performance of BSB starts degrading only when the offered background exceeds 3 file requests per second, which is well above 2.125

[2]Please note that if not differently stated, for the sake of simplicity, we assume the same meeting rate $\lambda_c = \lambda$ as well as the same interesting rate $\mu_c = \mu, \forall c \in \mathbb{C}$

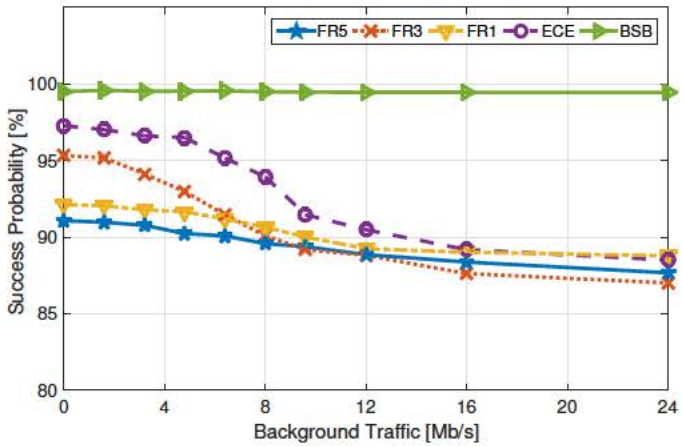


Figure 6.7: Content update success probability with 5 base stations, 750 users, and background traffic.

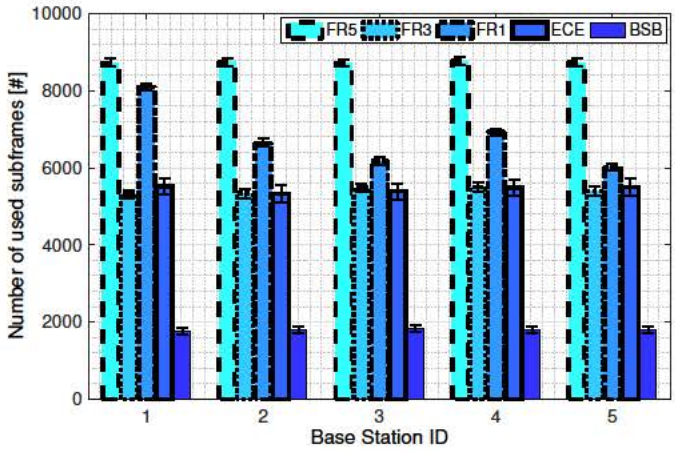


Figure 6.8: Content update transmission time with 5 base stations, 750 users, and no background traffic, for a content subscribe scenario with interesting rate $\mu = \infty$.

requests per second, i.e., the maximum value that guarantees the doability of content transmission within the deadline, according to (6.13). In contrast, frequency reuse schemes and ECE are seriously impaired by the background traffic as soon as the offered load reaches as low as 1 background file request per second.

6.4.5. Content subscription with multicast transmission

We finally assess the effect of our solution in a particular content subscription scenario in which users initially subscribe content updates (e.g., $\mu = \infty$) and get refresh replicas every time the content update is issued (every T_C seconds). This implies that base stations can easily inject the content into the network through a single multicast transmission. The transmission rate r_c^b is properly chosen to cover as many content subscribed users as possible (see Section 6.1.2).

Similarly to Fig. 6.4, Fig. 6.8 shows the per-base station average transmission time, expressed

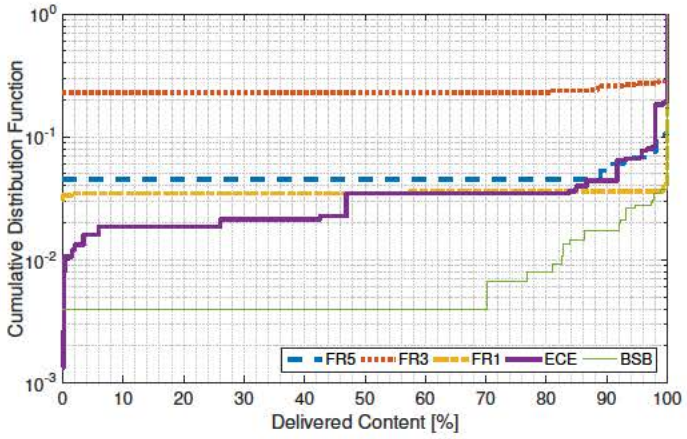


Figure 6.9: Content update success probability with 5 base stations, 750 users, and no background traffic, for a content subscribe scenario with interesting rate $\mu = \infty$.

in terms of used subframes, when the simultaneous update of 5 contents is periodically distributed in the network. Even in this case, BSB outperforms all other schemes and uses a number of subframes very close to the lower edge of the interval predicted in (6.12). Moreover, BSB outperforms FR3 and ECE by a factor ~ 3, and more than ~ 4 for the cases of FR5 and FR1. In addition, Fig. 6.9 reports the cumulative distribution function of the portion of delivered content updates, under the tested schemes. For this performance metric, we count the number of content updates that were correctly and entirely delivered to the subscribers, and normalise to the number of subscribers. BSB emerges as the scheme that guarantees the highest content delivery probabilities, resulting in 97.24% of delivered contents, on average. Noticeably, FR1, FR3, FR5 and ECE perform *much worst* than BSB. This result points out that both static frequency planning schemes and classic resource allocation schemes are not able to cope with the interference generated in dense environments. Moreover, FR3 achieves by far the worst results. Therefore, comparing FR1 (all base stations use the same wide bandwidth) and FR3 (at most two base stations share the same bandwidth, which is $1/3$ of the one used under FR1), we argue that the interference generated by few neighbors in a dense scenario is much less important than the available bandwidth. As a consequence, spectral efficiency over wide frequency bands is key to boost network performances, while bandwidth fragmentation due to frequency planning is undesirable.

6.4.6. Impact of network size

Table 6.2 illustrates a performance comparison in case of 7 base stations and 1 000 content subscribers, with 30 contents to be simultaneously distributed every 100 s. No background traffic was considered in this experiment. The table shows that the average base station activity time is minimal under BSB operations, and other schemes needs much higher utilizations. Being the overall load the same for all cases, the throughput sustained while transmitting is much higher for BSB than for any other scheme. Noticeably, BSB is only 10% below the maximum achievable

Table 6.2: Scenario with 7 base stations, 1000 users and 30 contents to disseminate

ICIC scheme	Transmitting Time [subframes in T_c]	Throughput [Mb/s]	Delivery Success Probability
FR5	14047	16.90	89.98%
FR3	6024	39.84	87.14%
FR1	11814	20.02	95.50%
ECE	6241	37.87	92.27%
BSB	2912	82.41	97.37%

LTE-A rate (the rate corresponding to perfect channel quality at any time). Moreover, BSB outperforms all other schemes also in terms of delivered contents. Most importantly, BSB achieves a 97.27% success probability, which means that D2D content dissemination is almost perfect when combined with BSB.

6.5. Conclusions

In this chapter, we have analyzed the content update dissemination process in cellular networks by shedding the light on the potential role of D2D communications while highlighting the base station interference coordination problem. Specifically, we focused on the *injection phase*, that is a key component of the dissemination process, yet it has been so far neglected. We have cast such a content injection problem into an optimization problem aiming at finding the optimal number of transmissions to maximise the content replica delivery. Notably, we have proved that the injection phase critically affects the opportunistic D2D content exchange. Based on this insight, another optimization problem has been formulated to minimize the time required to inject contents, which has been proven to be NP-Complete and NP-Hard to approximate. Then, we have proposed BSB, an eICIC algorithm for LTE-A networks that efficiently approximates the solution of the introduced problem.

The results show that BSB substantially outperforms classical intercell interference approaches and achieves performance figures better than what achievable with (soft fractional) frequency reuse schemes. Moreover, BSB boosts the D2D opportunistic communication performance by making the injection phase quasi-ideal, i.e., by minimizing the time needed to inject content replicas in the network.

Chapter 7

Summary & Conclusions

We have carried out a detailed analysis of the current mobile networking solutions, shedding the light on the main reasons because conventional cellular networks are not suitable for the coming future. We have analyzed the main problems from a mathematical viewpoint and we have proposed a set of solutions to provide high level of spectral efficiency while guaranteeing reasonable user fairness, as required by the future 5G stringent requirements. Our proposals are SDN-compliant and we have cast them into a novel framework, recently proposed in CROWD project, where CROWD Local Controllers (CLCs) decide on scheduling policies. Each optimization mechanism is developed as a SDN application by targeting two main topics: (i) mitigating the inter-cell interference problem when multiple base station are simultaneously active and (ii) supporting downlink cellular traffic offloading by exploiting the device-to-device technologies.

In Chapter 3, we have proposed a novel eICIC scheme which leverages the full knowledge of a central controller (CLC) in charge of gathering all user channel information. Based on that information, the algorithm takes time-scheduling decisions by means of the Almost Blank SubFrame (ABSF) technique. Once the ABSF patterns are properly issued, each base station schedules its own users in the next period by following the CLC instructions. In Chapter 4, we have moved towards a distributed approach due to an intractable complexity of a very-dense scenario. Base stations automatically exchange each other time scheduling information, e.g., the ABSF patterns, to reach a steady-state which has been proved to be near-optimal. This improves the network performance by offering a good level of fairness at both base station and user sides. In Chapter 5, we have focused on the content update distribution, where an adaptive algorithm has been presented to perform optimally the content dissemination phase. The network dynamics force to design a time-variant algorithm which aims at finding the optimal number of delegate users, which must disseminate the content information throughout the network. We have proved that our algorithm, in line with 5G vision, exhibits outstanding results when compared to conventional approaches in terms of offloading percentage, signalling overhead and stability. Finally, in Chapter 6, we have studied how to further enhance the offloading process by improving the base station injection phase, where the inter-cell interference problem is dominant. We have proposed

a smart and low-complexity algorithm which outperforms state-of-the-art solutions by guaranteeing minimum transmission delays. This directly benefits the entire offloading process, as the more the time to spread the content, the more users are reached through the D2D network rather than cellular network.

Appendices

Appendix A

Proofs of Theorems

This appendix provides the proofs of all theorems, propositions and lemmas in this book. Additionally, it presents in details the stochastic analysis of the content dissemination process carried out in Chapter 6. For reading convenience each section restates the theorem proven in it.

A.1. Proof of Lemma 4.2.1

Lemma 4.2.1. *Given that the players' strategies belong to whatever strategy profile σ, after a finite number of single-step best responses (SSBR), all players' strategies will belong to a saturation strategy profile $\overline{\sigma}$.*

Proof: We prove the lemma by contradiction. Let us define two sets of players that represent the state of the game at a given point before reaching a saturation strategy profile $\overline{\sigma}$. Specifically, set $\mathcal{P}$ includes all players that have penalties, and set $\mathcal{Z}$ includes the remaining players, which suffer zero penalty. Assume now that all players switch to Single-Step Best Response (SSBR) at a given point in time, and consider the composition of $\mathcal{P}$ and $\mathcal{Z}$ at that point.

If the lemma were incorrect, the game could evolve from this state and at least one player could not reach a saturation strategy profile $\overline{\sigma}$. This possibility implies that sets $\mathcal{P}$ and $\mathcal{Z}$ are always nonempty at the end of each round of the game. If not, the evolution of the game would lead to have either all players using all the TTIs (all players being in $\mathcal{P}$ would cause a progressive increment in the use of TTIs until all TTIs are used by all players), or all players suffering no penalty (all players in $\mathcal{Z}$ would reduce the use of TTIs as much as possible, without incurring in penalties). In both cases, we would have a saturation strategy profile for all players.

Moreover, there must exist a continuous flow of players moving between $\mathcal{P}$ and $\mathcal{Z}$ while the game evolves. In fact, should the flow stop after a finite number of moves, all players in $\mathcal{P}$ would increase the number of TTIs used until they reach the maximum (because they have penalties to pay), and all players in $\mathcal{Z}$ would decrease the number of TTIs used without incurring any penalty. However, by definition, this would be a saturation strategy profile $\overline{\sigma}$.

Therefore, to admit that SSBR does not lead to a saturation strategy, we have to admit that players move continuously between $\mathcal{P}$ and $\mathcal{Z}$. Moreover, since the two sets have to remain nonempty at the end of any round of the game, the flows of players from $\mathcal{P}$ to $\mathcal{Z}$ and vice versa have to be balanced. With two possible states, this also implies that the probability to be in $\mathcal{P}$ is the same as the probability to be in $\mathcal{Z}$; therefore, the average sojourn times of a player in $\mathcal{P}$ and $\mathcal{Z}$ are the same.

In particular, let us consider a player p that keeps moving between $\mathcal{P}$ and $\mathcal{Z}$, and let us call d the average time spent in each of the two states. Let us consider a *cycle* of player p, from her passage to $\mathcal{P}$ to her return to $\mathcal{Z}$. In the transition $\mathcal{Z} \rightarrow \mathcal{P}$, p will increase the TTI utilization by 1 unit, to try to come back to $\mathcal{Z}$ immediately. Then she will spend $d-1$ rounds in $\mathcal{P}$, during which she increments by $d-1$ units her TTI utilization. Afterwards, p goes back to $\mathcal{Z}$, which can occur with an increase of one TTI or with no changes (because of other player's changes of strategy). Eventually, player p will spend $d-1$ rounds in $\mathcal{Z}$, during which she will decrement the TTI utilization by *at most* $d-1$ units. The resulting balance is a net increase in the number of used TTIs. Therefore, all players moving continuously between $\mathcal{P}$ and $\mathcal{Z}$ should eventually end up using all the available TTIs and have no way to further change their strategy profiles. Hence, the flow between $\mathcal{P}$ and $\mathcal{Z}$ would stop. This would lead again to a saturation strategy profile $\overline{\sigma}$. ■

A.2. Proof of Lemma 4.2.2

Lemma 4.2.2. *At a certain point in time, given that the strategies played by any player in the system belong to a saturation strategy profile $\overline{\sigma}$ if each player chooses a single-step best response (SSBR), the game will converge to a Nash equilibrium.*

Proof: We can derive from Problem `GBR-DISTR` the cost function f_i related to a player strategy S_i and the strategies taken by the other players (S_{-i}) as follows

$$f_i(S_i, S_{-i}) = |S_i| + \alpha \cdot \sum_{u \in U_i} \rho_u(S_i, S_{-i}), \quad \forall S_i \in \mathbb{S}_i;$$

$$\rho_u(S_i, S_{-i}) = \max\left(D_u - \sum_{(u,t) \in S_i} c_{u,t}(S_{-i}), 0\right), \quad \text{(A.1)}$$

where $\rho_{S_i,S_{-i}}$ is the penalty that user u has to pay in order to satisfy its user traffic demand D_u. It is clear that each player chooses her single-step best response in order to minimize $f_i(S_i, S_{-i})$. Due to the saturation strategy profile, if player i presents at step $k-1$ a zero penalty, all users' traffic demands D_u, $\forall u \in U_i$, are satisfied with the current player's strategy $S_i^{(k-1)}$. Noticing that the cost function is not decreasing with users' strategy cardinality, in the case of saturation with zero penalty, the only relevant term in the cost function is the cardinality of the current strategy $|S_i|$ (i.e., the number of TTIs used for scheduling the users). Hence, at next step k, player i will choose a strategy $S_i^{(k)}$ such that $|S_i^{(k)}| \leq |S_i^{(k-1)}|$ due to the single-step best response, which leads

to

$$f(S_i^{(k)}, S_{-i}) \leq f(S_i^{(k-1)}, S_{-i}), \tag{A.2}$$

where $f(\cdot)$ is the cost function defined in (A.1). Given that the player change $S_i^{(k-1)}$ will not increase the inter-cell interference towards the other cells, it may benefit the other players choices. Hence, the penalty value in the cost function will not ever be increased by the other players, and the updated strategy profile $\sigma^{(k)}$ is still a saturation strategy profile at step k. Therefore, we deduce that the inequality (A.2) will be satisfied for all players' moves in the system, at any step k.

Since we assume a non-decreasing cost function, each player will get the minimum of the cost function in a finite number of steps. Upon all players choose the particular strategy returning the minimum of the cost function, they have reached a Nash equilibrium. Furthermore, if player strategies take all available T TTIs with a non-zero penalty, the players have already reached a Nash equilibrium. Since they cannot further increase the number of involved TTIs, no further move will improve their cost function. ■

A.3. Proof of Theorem 5.1.1

Theorem 5.1.1. *According to the HYPE Markov chain for heterogeneous mobility (similar to Fig. 5.2), the process $\{M(t), t \geq 0\}$ is described by the following system of differential equations:*

$$\begin{cases} \frac{d}{dt}p_1^{c_1}(t) = -\lambda_1 p_1^{c_1}(t), & i = 1 \\ \frac{d}{dt}p_i^{c_1}(t) = -\lambda_i p_i^{c_1}(t) + \lambda_{i-1} p_{i-1}^{c_1}(t), & 1 < i < N \\ \frac{d}{dt}p_N^{c_1}(t) = \lambda_{N-1} p_{N-1}^{c_1}(t), & i = N \end{cases} \tag{A.3}$$

where $\lambda_i = i(N-i)\mu_\beta$. (Recall that μ_β is the known expectation of the generic probability distribution $F(\beta) : (0, \infty) \to [0, 1]$, from which the inter-contact rates describing our network are drawn: $\{\beta_{xy}\} = \mathbf{B}$.)

Proof: Recall that we denoted by $\{\mathbf{K}_1^i, \mathbf{K}_2^i, \ldots, \mathbf{K}_{\binom{N}{i}}^i\}$ the set of $\binom{N}{i}$ states in the Markov chain corresponding to level i. Also, $\mathbf{B} = \{\beta_{xy}\}$ is our network. Then, assuming the Markov chain starts in initial state $\mathbf{K}_m^1$ for $1 \leqslant m \leqslant N$ (i.e., the chunk was initially injected to a node m), the probability of still being in this state after a small time interval dt is:

$$\mathbb{P}\left[\mathbf{K}_m^1 \,\mathrm{at}\, t + \mathrm{d}t | \mathbf{B}\right] = \mathbb{P}\left[\mathbf{K}_m^1 \,\mathrm{at}\, t | \mathbf{B}\right] \cdot \left(1 - \sum_{\substack{x \in \mathbf{K}_m^1 \\ y \notin \mathbf{K}_m^1}} \beta_{xy}\, \mathrm{d}t\right) \tag{A.4}$$

Then, averaging over all states of this dissemination level, the probability of still being at

dissemination level 1 after a small time interval dt is:

$$\mathbb{P}\left[1 \text{ at } t+\mathrm{d}t \mid \mathbf{B}\right] = \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t+\mathrm{d}t \mid \mathbf{B}\right] \tag{A.5}$$

Finally, considering all possible network realizations:

$$\begin{aligned} p_1^{c_1}(t+dt) &= \mathbb{P}\left[1 \text{ at } t+\mathrm{d}t\right] \\ &= \int_{\mathbf{B}} \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t+\mathrm{d}t \mid \mathbf{B}\right] \mathbb{P}\left[\mathbf{B}\right] \mathrm{d}\mathbf{B}, \end{aligned} \tag{A.6}$$

where $\mathbb{P}\left[\mathbf{B}\right]$ is given by the generic distribution, $F(\beta) : (0,\infty) \to [0,1]$, which determines the inter-contact rates of our network. Combining this last equation with Eq. (A.4) and using basic probability theory, we obtain Eq. (A.7) below:

$$\begin{aligned} p_1^{c_1}(t+dt) &= \int_{\mathbf{B}} \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{at} t | \mathbf{B}\right] \left(1 - \sum_{\substack{x\in\mathbf{K}_m^1 \\ y\notin\mathbf{K}_m^1}} \beta_{xy}\mathrm{d}t\right) \mathbb{P}\left[\mathbf{B}\right]\mathrm{d}\mathbf{B} \\ &= \int_{\mathbf{B}} \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t \mid \mathbf{B}\right] \mathbb{P}\left[\mathbf{B}\right] \mathrm{d}\mathbf{B} - \\ &\quad - \int_{\mathbf{B}} \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t \mid \mathbf{B}\right] \sum_{\substack{x\in\mathbf{K}_m^1 \\ y\notin\mathbf{K}_m^1}} \beta_{xy}\,\mathrm{d}t \cdot \mathbb{P}\left[\mathbf{B}\right]\,\mathrm{d}\mathbf{B} \\ &= p_1^{c_1}(t) - \sum_{m=1}^{N} \int_{\mathbf{B}} \sum_{\substack{x\in\mathbf{K}_m^1 \\ y\notin\mathbf{K}_m^1}} \beta_{xy}\,\mathrm{d}t \cdot \mathbb{P}\left[\mathbf{B}|\mathbf{K}_m^1 \text{at} t\right] \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t\right] \mathrm{d}\mathbf{B} \\ &= p_1^{c_1}(t) - \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t\right] \mathbb{E}\left[X \mid \mathbf{K}_m^1 \text{ at } t\right]\,\mathrm{d}t, \end{aligned} \tag{A.7}$$

where $X = \sum_{\substack{x\in\mathbf{K}_m^1 \\ y\notin\mathbf{K}_m^1}} \beta_{xy}$ (that is a sum of $N-1$ terms).

Since our network's inter-contact rates forming the matrix $\mathbf{B}$ are independent and identically distributed (with generic distribution $F(\beta) : (0,\infty) \to [0,1]$ of mean μ_β), the terms of the sum forming X are distributed according to $F(\beta)$, regardless of the specific node combination $\mathbf{K}_m^1$. Hence, $\mathbb{E}\left[X \mid \mathbf{K}_m^1 \text{ at } t\right] = \mathbb{E}\left[X\right]$ and Eq. (A.7) becomes:

$$p_1^{c_1}(t+dt) = p_1^{c_1}(t) - \mathbb{E}\left[X\right]\,\mathrm{d}t \cdot \sum_{m=1}^{N} \mathbb{P}\left[\mathbf{K}_m^1 \text{ at } t\right] \tag{A.8}$$

$$= p_1^{c_1}(t) - (N-1)\mu_\beta\,\mathrm{d}t \cdot p_1^{c_1}(t). \tag{A.9}$$

Thus, we obtain as desired:

$$\frac{d}{dt}p_1^{c_1}(t) = -(N-1)\mu_\beta p_1^{c_1}(t) \tag{A.10}$$

The remaining two differential equations are obtained by the same process. ■

A.4. Proof of Theorem 5.2.1

Theorem 5.2.1. *In the optimal strategy, the data chunk is delivered through the cellular network to d seed nodes at time $t = 0$, and to the nodes that do not have the content by the deadline $t = T_c$.*

Proof: The proof goes by contradiction: we first assume that in the optimal strategy the data chunk is transmitted to some mobile node at time $t \neq \{0, T_c\}$ and then we find an alternative strategy that provides a better performance.

If the chunk is transmitted to some mobile node at $t \neq \{0, T_c\}$, this means that $C \neq \{1, \ldots, d\}$ and hence there exists some missing value smaller than c_d in C. Indeed, if $C = \{1, \ldots, d\}$, all the first d states are instantaneous states and the data chunk is transmitted to d nodes at the beginning of the round.

Let us denote the largest value in C (c_d) by k and the largest value that is missing by $k - l$. Let D_k further denote the value of D for the optimal configuration $C_k = \{c_1, \ldots, c_d\}$ (where $c_d = k$), D_{k-l} the value of D for the configuration $C_{k-l} = \{c_1, \ldots, c_{d-1}, k - l\}$ and D_{k+1} the value of D for the configuration $C_{k+1} = \{c_1, \ldots, c_{d-1}, k + 1\}$.[1] In the following, we show that either D_{k-l} or D_{k+1}, or both, are smaller than D_k, which contradicts the initial assumption that the configuration $\{c_1, \ldots, c_d\}$ is optimal.

If we compare the state probabilities for the configurations C_k and C_{k+1}, we have that

$$\begin{cases} P_i^{C_k}(s) = P_i^{C_{k+1}}(s), & i < k \\ P_i^{C_k}(s) = \dfrac{\lambda_{k+1}(s+\lambda_k)}{\lambda_k(s+\lambda_{k+1})} P_i^{C_{k+1}}(s), & i > k+1. \end{cases} \tag{A.11}$$

From the above, we have that the following holds for $i > k + 1$,

$$\begin{aligned} P_i^{C_k}(s) - P_i^{C_{k+1}}(s) &= \left(\frac{\lambda_{k+1}}{s+\lambda_{k+1}} - \frac{\lambda_k}{s+\lambda_k}\right) \prod_{j \in S_{i-1}^{C_k} \setminus k} \frac{\lambda_j}{s+\lambda_j} \\ &= \frac{\lambda_{k+1} - \lambda_k}{\lambda_{k+1}\lambda_k} s P_i^{C_{d-1}}(s), \end{aligned} \tag{A.12}$$

where $S_{i-1}^{C_k} = \{1, 2, \ldots, i-1\} \setminus (\{1, 2, \ldots, i-1\} \cap \{c_1, \ldots, c_d\})$ and $P_i^{C_{d-1}}(s)$ is state i's

[1] Without loss of generality we assume that $c_d \neq N$, as it can be easily shown that a configuration with $c_d = N$ is not optimal.

probability for the configuration $C_{d-1} = \{c_1, \ldots, c_{d-1}\}$.

By doing the inverse Laplace transform of the above, we have that

$$p_i^{C_k}(t) - p_i^{C_{k+1}}(t) = \frac{\lambda_{k+1} - \lambda_k}{\lambda_k \lambda_{k+1}} \frac{dP_i^{C_{d-1}}(t)}{dt} = \frac{\lambda_{k+1} - \lambda_k}{\lambda_k \lambda_{k+1}} \Big(-\lambda_i P_i^{C_{d-1}}(t) + \lambda_{i-1} P_{i-1}^{C_{d-1}}(t) \Big). \tag{A.13}$$

Furthermore, we also have

$$P_{k+1}^{C_k}(s) - P_k^{C_{k+1}}(s) = \frac{\lambda_k - \lambda_{k+1}}{\lambda_k} P_{k+1}^{C_{d-1}}(s), \tag{A.14}$$

and, hence,

$$p_{k+1}^{C_k}(t) - p_k^{C_{k+1}}(t) = \frac{\lambda_k - \lambda_{k+1}}{\lambda_k} p_{k+1}^{C_{d-1}}(t). \tag{A.15}$$

Combining Eqs. (A.13) and (A.15) with Eq. (5.1), and taking into account that for $i > k+1$ it holds that $d_i + d_i^* = d_{i+1} + d_{i+1}^* + 1$, we obtain

$$D_k - D_{k+1} = (\lambda_k - \lambda_{k+1}) \sum_{i=k+1}^{N-1} \frac{1}{\lambda_k \lambda_{k+1}} p_i^{C_{d-1}}(T_c). \tag{A.16}$$

Following a similar approach for the configurations C_k and C_{k-l}, we obtain

$$D_k - D_{k-l} = (\lambda_k - \lambda_{k-l}) \sum_{i=k+1}^{N-1} \frac{1}{\lambda_k \lambda_{k-l}} p_i^{C_{d-1}}(T_c). \tag{A.17}$$

Since it holds that either $\lambda_k - \lambda_{k-l}$ or $\lambda_k - \lambda_{k+1}$ is greater than zero, we have that at least one of the two alternative configurations (C_{k+1} or C_{k-l}) provides a D value smaller than C_k. This contradicts the assumption that in the optimal strategy the data chunk is transmitted to some node at time $t \neq \{0, T_c\}$, which proves the theorem. ■

A.5. Proof of Proposition 5.2.1

Proposition 5.2.1. *Let us define G_d as the gain resulting from sending the $(d+1)^{th}$ chunk of chunk copy at the beginning of the period (i.e., $G_d = D_d - D_{d+1}$, where D_{d+1} and D_d are the values of D when we deliver $d+1$ and d copies at the beginning, respectively). Then, G_d can be computed from the following equation:*

$$G_d = \sum_{j=d}^{N-1} \frac{\lambda_j}{\lambda_d} p_j^d(T_c) - 1. \tag{A.18}$$

Proof: G_d can be expressed as:

$$G_d = \sum_{j=d}^{N-1} (N-j) \left(p_j^d(T_c) - p_j^{d+1}(T_c) \right) - 1. \tag{A.19}$$

The term $p_j^d(T_c) - p_j^{d+1}(T_c)$ is calculated as follows. From Eqs. (5.5), we have

$$P_j^d(s) - P_j^{d+1}(s) = -\frac{sP_j^d(s)}{\lambda_d}. \tag{A.20}$$

Making the inverse Laplace transform of the above for $j > d$ yields

$$\begin{aligned} p_j^d(T_c) - p_j^{d+1}(T_c) = -\frac{1}{\lambda_d} \frac{dp_j^d(t)}{dt}\bigg|_{T_c} = \\ = \frac{1}{\lambda_d}(\lambda_j p_j^d(T_c) - \lambda_{j-1} p_{j-1}^d(T_c)). \end{aligned} \tag{A.21}$$

Note that the above equation also holds for $j = d$ since in this case $p_{j-1}^d(t) = 0$ and $p_j^{d+1}(t) = 0$. Combining it with Eq. (A.19) leads to

$$G_d = \sum_{j=d}^{N-1} \frac{\lambda_j}{\lambda_d} p_j^d(T_c) - 1. \tag{A.22}$$

■

A.6. Proof of Theorem 5.2.2

Theorem 5.2.2. *The optimal value of* d *is the one that satisfies* $G_d = 0$.

Proof: As long as $G_d > 0$, we benefit from increasing d, since by sending one additional chunk at the beginning, we save more than one chunk at the end. Conversely, if $G_d < 0$ we do not benefit. It can be seen that $G_1 > 0$ and $G_N < 0$. Furthermore, it can also be seen that G_d strictly decreases with d:

$$\begin{aligned} G_{d+1} - G_d = \sum_{j=d}^{N-1} \frac{\lambda_j}{\lambda_{d+1}} p_j^{d+1}(T_c) - \frac{\lambda_j}{\lambda_d} p_j^d(T_c) \\ = \sum_{j=d}^{N-1} \frac{p_j^d(T_c)\lambda_j(\lambda_{j+1} - \lambda_j - (\lambda_{d+1} - \lambda_d))}{\lambda_d \lambda_{d+1}} < 0. \end{aligned} \tag{A.23}$$

From the above, it follows that the value of d that minimizes D is the one that satisfies $G_d = 0$, since up to this value we benefit from increasing d and after this value we stop benefiting, which proves the theorem. ■

A.7. Proof of Theorem 5.2.3

Theorem 5.2.3. *The HYPE control system is stable for $K_p = 0.2$ and $K_i = 0.4/3.4$.*

Proof: The closed-loop transfer function of our system is

$$T(z) = \frac{-z(z-1)HK_p - zHK_i}{z^2 + (-HK_p - 1)z + H(K_p - K_i)} \tag{A.24}$$

where

$$H = -2 \tag{A.25}$$

A sufficient condition for stability is that the poles of the above polynomial fall within the unit circle $|z| < 1$. This can be ensured by choosing coefficients $\{a1, a2\}$ of the characteristic polynomial that belong to the stability triangle [10]:

$$\begin{cases} a_2 < 1 \\ a_1 < a_2 + 1 \\ a_1 > -1 - a_2 \end{cases} \tag{A.26}$$

In the transfer function of Eq. (A.24), the coefficients of the characteristic polynomial are $a_1 = -HK_p - 1$ and $a_2 = H(K_p - K_i)$. From Eqs. (5.24) and (A.25), we have $HK_p = -0.4$ and $HK_i = -0.4/(0.85 \cdot 2)$, from which $a_1 = -0.6$ and $a_2 = -0.16$. It can be easily seen that these $\{a1, a2\}$ values satisfy Eq. (A.26), which proves the theorem. ■

A.8. Proof of Theorem 6.2.1

Theorem 6.2.1. *Problem `BS-Time-Scheduling` is NP-Complete, for any $Z \geq 3$, even when all interferences are $\in \{0, 1\}$.*

Proof: It is clear that the problem is in NP. For the NP-Hardness we use a reduction from the problem `GCk` of graph k-coloring (see [35]). We are given an instance $I_{\texttt{GCk}} = H(V, E)$ of Problem `GCk`, and construct an instance $I_{\texttt{BS-Time-Scheduling}}$ of Problem `BS-Time-Scheduling`. Assume $V = \{1, 2, \cdots, n\}$. The base stations are $B = \{b_1, b_2, \cdots, b_n\}$, and the users $U = \{u_1, u_2, \cdots, u_n\}$, where for every t base station b_t is serving user u_t. In addition, $Z = k$, $N_0 = \tau = B_T = L_c = 1$, $\Theta = n$, and $S_{u_t}^{b_t}(i) = w_a(i) = 1$ for every $i = 1, 2, \cdots, Z$, $t = 1, 2, \cdots, n$. Last, for every $t = 1, 2, \cdots, n$, every $j \neq t$ and every $i = 1, 2, \cdots, Z$, $I_{u_t}^{b_j}(i) = 1$ if $(i, j) \in E$ and is 0 otherwise.

We have to show that there is a k-coloring of $I_{\texttt{GCk}}$ if and only if for $I_{\texttt{BS-Time-Scheduling}}$ there is a scheduling of the base stations in at most k rounds, with $\Psi_{u_t}^{b_t}(Z) \geq 1 = L_c$, and $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$.

Given a graph k-coloring of $I_{\texttt{BS-Time-Scheduling}}$, with colors $1, 2, \cdots, k$. If a node t is colored p, then we schedule station b_t in round p, for $p = 1, 2, \cdots, k$.
$\Psi_{u_t}^{b_t}(Z) = \sum_{i=1}^{Z} x_{ib} \log_2 \left(1 + \frac{1}{1+\sum_{j\neq t} I_{u_t}^{b_j}(i) x_{ij}}\right)$ for every t. Since all base stations b_j scheduled with b_t are such that $(j,t) \notin E$, and since each base station is scheduled in exactly one round, therefore $\Psi_{u_t}^{b_t}(3) = \log_2\left(1+\frac{1}{1}\right) = 1$. $\sum_{t=1}^{n} T_{TOT}^{b_t} = n$ since each station is scheduled in exactly one round.

Conversely, assume that for $I_{\texttt{BS-Time-Scheduling}}$ there is a general scheduling of at most k rounds, such that for each user $\Psi_{u_t}^{b_t}(k) \geq 1$ and $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$. $\Psi_{u_t}^{b_t}(k) > 0$ implies that each user—and thus each station—is scheduled in at least one round. $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$ implies that each station—and thus each user—is scheduled in exactly one round. Moreover, if user u_i is scheduled with user u_j, then $(i,j) \notin E$ (otherwise $\Psi_{u_i}^{b_i}(Z) < 1 = L_c$). Thereby assigning color p to nodes associated with the stations in round $p = 1, 2, \cdots, k$, results in a k-coloring of graph $I_{\texttt{GCk}}$. ■

A.9. Proof of Theorem 6.2.2

Theorem 6.2.2. *Problem* `BS-Time-Scheduling` *is NP-Complete for* $Z = 2$.

Proof: We use a reduction from a variation of the Partition problem. We term this Problem `MPAR`. In the Partition problem we are given integers $A = \{a_1, a_2, \cdots, a_n\}$, such that $\sum_{j=1}^{n} a_j = 2S$, and have to determine whether there exist $\{a'_1, a'_2, \cdots, a'_k\} \subseteq A$ such that $\sum_{j=1}^{k} a'_j = S$ (see [35]). In the modified version `MPAR` (that can be shown to be NP-Complete) we are given integers $A = \{x_1, x_2, \cdots, x_{2n}\}$, $S > 0$, $S < x_i < 2S$ for all i, such that $\sum_{j=1}^{2n} x_j = 2(n+1)S$, and have to determine whether there exist $\{x'_1, x'_2, \cdots, x'_n\} \subseteq A$ such that $\sum_{1}^{n} x'_j = F$, where $F = (n+1)S$.

We are given an instance I of `MPAR`, and construct an instance $I_{\texttt{BS-Time-Scheduling}}$ of Problem `BS-Time-Scheduling` as follows. The base stations are $B = \{b_1, b_2, \cdots, b_{2n}\}$, and the users $U = \{1, 2, \cdots, 2n\}$; base station b_i is serving user i. $Z = 2$, $N_0 = F$, $\tau = B_T = L_c = 1$, $\Theta = n$, and $S_{u_t}^{b_t}(i) = 2F$, $w_a(i) = 1$ for $i = 1, 2$, $t = 1, 2, \cdots, n$. Last, for every $t = 1, 2, \cdots, n$, every $j \neq t$ and every $i = 1, 2$: $I_{u_t}^{b_j}(i) = x_j + \frac{x_i}{n-1}$.

We have to show that there is a solution to I if and only if there is a scheduling for $I_{\texttt{BS-Time-Scheduling}}$ in at most 2 rounds, such that for each user $\Psi_{u_t}^{b_t}(2) \geq 1 = L_c$, and $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$.

Assume there is a solution to I. Thus we assume the existence of a $\{x'_1, x'_2, \cdots, x'_n\} \subset A$ such that $\sum_{1}^{n} x'_j = F$. Schedule the base stations $b_{x'_1}, b_{x'_2}, \cdots, b_{x'_n}$ in the first round and the other n base stations in the second round. Clearly $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$.

Every user t is thus scheduled in exactly one round, and thus

$$\Psi_{u_t}^{b_t}(2) \quad = \quad \log_2 \left(1 + \frac{2F}{F + \sum \left\{ x'_j + \frac{x'_i}{n-1} \middle| j=1,..,n, j\neq i \right\}}\right) \quad = \quad \log_2 \left(1 + \frac{2F}{F + \sum_1^n x'_j}\right) \quad =$$

$\log_2\left(1+\frac{2F}{F+F}\right)=1$.

Conversely, assume a solution to $I_{\texttt{BS-Time-Scheduling}}$. Since each interference is positive, and since $\sum_{t=1}^{n} T_{TOT}^{b_t} \leq n$, it follows that each station is scheduled in exactly one round.

Assume the base stations at the first round are $b_1, b_2, \cdots, b_k$, and in the second round are $b_{k+1}, \cdots, b_{2n}$. If $k \neq n$ then one of these rounds has more than n base stations. Assume, with no loss of generality, that $k > n$. This means that $\sum\left\{x_j + \frac{x_i}{n-1}\middle| j=1,2,\cdots,k, j\neq i\right\} > nS + \frac{nx_i}{n-1} > F$, for every $i=1,2,\cdots,k$, thus $\Psi_{u_i}^{b_i}(2) < 1$, a contradiction. Therefore $k = n$. The interference of each of the users in the first (second) round is $\log_2\left(1+\frac{2F}{F+\sum_{i=1}^{n} x_i}\right)$ ($\log_2\left(1+\frac{2F}{F+\sum_{i=n+1}^{2n} x_i}\right)$). So, $\sum_{i=1}^{n} x_i = \sum_{i=n+1}^{2n} x_i = F$, and all interferences are 1. ■

A.10. Proof of Theorem 6.2.3

Theorem 6.2.3. *For all $\epsilon > 0$, approximating within $n^{1-\epsilon}$ the minimal number of rounds required to solve Problem* `BS-Time-Scheduling` *with n base stations is NP-hard.*

Proof: Following the same reduction from `GCk`, as done in the proof of Theorem 6.2.1, it is clear that the instance of `BS-Time-Scheduling` can be scheduled in k rounds if and only if the given graph can be colored with k colors. Therefore the existence of an algorithm with approximation ratio $n^{(1-\epsilon)}$ for `BS-Time-Scheduling` will imply the existence of an algorithm with the same approximation ratio for `GCk`. ■

A.11. Stochastic analysis of the Content Dissemination process

The content dissemination process is described with the bi-dimensional Markov chain, as depicted in Fig A.1. We define $S_j(t)$ as the total *number of content replica* distributed in the network at time t given j users interested in the content, regardless of the specific users carrying those replica. An homogeneous mobility model is assumed, users get in touch each other following an average inter-contact rate λ and get interested in a content according to an average rate μ. Therefore, transition rates depend on the j amount of users interested in the content as well as on the number of users which have already obtained the content. Finally, the number of users which have received the content directly from the base station is represented by value n_c^b (number of injected nodes). Varying the number of injected nodes n_c^b the Markov chain is slightly affected, considering as first column only those S_j states whose the number of users with the content is equal or greater than the number of injected nodes.

Therefore, in order to solve the bi-dimensional Markov chain of Fig. A.1, we write the forward

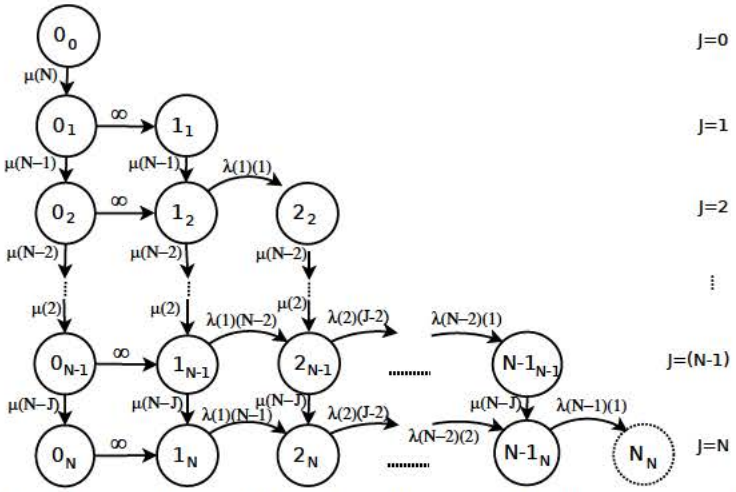


Figure A.1: Markov chain explaining the content dissemination phase performed by base station b for content c. Users get interested in the content with an average rate equal to μ, while getting in contact with a λ intercontact rate. The number of initial injected node is $n_c^b = 1$.

Kolmogorov equations as:

$$\begin{cases} \dot{p}_{1_1}(t) = -\mu\,(N-1)\,p_{1_1}(t); \\ \dot{p}_{1_2}(t) = \mu\,(N-1)\,p_{1_1}(t) - (\mu\,(N-2) + \lambda\,1\,(1))\;p_{1_2}(t); \\ \dot{p}_{2_2}(t) = \lambda\,1\,(1)\,p_{1_2}(t) - \mu\,(N-2)\,p_{2_2}(t); \\ \quad\vdots \\ \dot{p}_{i_j}(t) = \lambda\,(i-1)(j-(i-1))\,p_{i-1_j}(t) + \mu\,(N-(j-1))\,p_{i_{j-1}}(t); -(\mu(N-j)+\lambda i(J-i))\;p_{i_j}(t); \\ \quad\vdots \\ \dot{p}_{N-1_{N-1}}(t) = \lambda\,(N-2)\,(1)\,p_{N-2_{N-1}}(t) - \mu\,(1)\,p_{N-1_{N-1}}(t); \\ \dot{p}_{N_N}(t) = \lambda\,(N-1)\,(1)\,p_{N_N}(t). \end{cases} \tag{A.27}$$

Please note that we enumerate with vector $\mathbb{S} = \{S_j(t)\}$, all the states of the Markov chain, starting counting by rows from the first state $\{S_1 = 1\}$, while we use vector $\vec{K}$ to represent the set of the unique indexes associated to every state $S(t)$, regardless the amount of users interested in the content. Please note that vectors $\mathbb{S}$ and $\vec{K}$ are time-independent. Indeed, $|\vec{K}| < |\mathbb{S}| = \frac{N(N+1)}{2}$.

Let $\vec{P}_J(t) = [p_{1_1}(t), p_{1_2}(t), p_{2_2}(t), ..., p_{S_j}(t), ..., p_{N_N}(t)]$ the set of probabilities at time t to be in each of the states $S_j \in \mathbb{S}$, while $\vec{P}(t) = [p_1(t), p_2(t), ..., p_x(t), ..., p_N(t)]$ the set of probabilities to have $S(t)$ users with the content at time t, where $p_x(t) = \sum_{j=1}^{x} p_{x_j}(t)$. To solve the set of Kolmogorov equations we can rewrite (A.27) as follows:

$$\vec{P}(t) = e^{-(\mathrm{L}\lambda + \mathrm{M}\mu)\mathrm{t}}\,\vec{C} \tag{A.28}$$

where C is a null vector of $(1 \times N)$ size, with only one non-zero value equal to 1 corresponding

to the starting state index, while L and M are square matrices with $(N \times N)$ size. We define the structure of those matrices as follows:

$$L = \begin{vmatrix} 0 & & & & \\ & l_2 & & & \\ & & l_J & \cdots & \\ & & \vdots & & \vdots \\ & & & \cdots & l_N \end{vmatrix}, \text{where } l_J = \{l_{iz}\}, \text{and } l_{iz} = \begin{cases} (i)(J-i) & \text{if } i = z, \\ -(i)(J-i) & \text{if } i = z+1, \\ 0 & \text{otherwise.} \end{cases}$$

$$M = \begin{vmatrix} m_1 & & & & \\ & m_J & \cdots & & \\ & \vdots & & \vdots & \\ & & \cdots & m_{N-1} & \\ & & & & 0 \end{vmatrix}, \text{where } m_J = \{m_{iz}\}, \text{and } m_{iz} = \begin{cases} (N-J) & \text{if } i = z, \\ -(N-J) & \text{if } i = z+J, \\ 0 & \text{otherwise.} \end{cases}$$

Note that for matrix indices we use the same order as reported in vector $\vec{K}$. This is important in order to have a general scheme to create those matrices. In matrix L we can identify $N-1$ square blocks l_J with $[J \times J]$ size. Considering Fig. A.1 as a reference Markov chain, each of those blocks provides the transition rates of any single row of the Markov chain due to user meetings (except the first row). The longer the row, the larger the block, the more transition rate values. In matrix M we can identify $N-1$ non-singular blocks with $[J \times 2J]$ size, which take into account the transition rates due to new request from an interested user. Indeed, we obtain the average number of users with content at the end of the content lifetime T_c as follows

$$\mathbb{E}[S_j(T_c)] = \sum_{i=1}^{N} i \; p_i(T_c - d_{in}). \tag{A.29}$$

Neglecting the content transmission time with respect to the time between two users get interested in the content, the time elapsed after injecting n_b^i content replica is, on average, $d_{in} = \sum_{u=0}^{n_c^b-1} \frac{1}{\mu(N-u)}$. Indeed, using (A.28) in (A.29), we obtain

$$\mathbb{E}[S_j(T_c)] = \vec{V} \; e^{-(\mathrm{L}\lambda+\mathrm{M}\mu)(T_c-d_{in})} \; \vec{C} \tag{A.30}$$

where $\vec{V}$ is similar to $\vec{K}$, except that it includes only states with at least n_c^b users holding the content replica.

References

[1] H. Ali-Ahmad, C. Cicconetti, A De La Oliva, M. Draxler, R. Gupta, V. Mancuso, L. Roullet, and V. Sciancalepore. Crowd: An sdn approach for densenets. In *Software Defined Networks (EWSDN), 2013 Second European Workshop on*, pages 25–31, Oct 2013.

[2] V. Sciancalepore, V. Mancuso, and A. Banchs. Basics: Scheduling base stations to mitigate interferences in cellular networks. In *The 14th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Madrid, Spain*, June 2013.

[3] V. Sciancalepore, I. Filippini, V. Mancuso, A. Capone, and A. Banchs. A semi-distributed mechanism for inter-cell interference coordination exploiting the absf paradigm. In *The 12th IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Seattle, USA, June 2015.

[4] V. Sciancalepore, D. Giustiniano, A. Banchs, and A. Picu. Offloading cellular traffic through opportunistic communications: Analysis and optimization. *Accepted for publication at IEEE Journal on Selected Areas in Communications*, 2015.

[5] V. Sciancalepore, V. Mancuso, A. Banchs, S. Zaks, and A. Capone. Interference coordination strategies for content update dissemination in LTE-A. In *The 33rd IEEE International Conference on Computer Communications (INFOCOM)*, Toronto, Canada, May 2014.

[6] Cisco visual networking index: Global mobile data traffic forecast update, 2011 – 2016. White Paper, 2012. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html`.

[7] Gartner says worldwide smartphone sales soared in fourth quarter of 2011 with 47 percent growth. Press Release, 2012. `http://www.gartner.com/it/page.jsp?id=1924314`.

[8] M. Y. Arslan, J. Yoon, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee. FERMI: a femtocell resource management system forinterference mitigation in OFDMA networks. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 25–36, New York, NY, USA, 2011. ACM.

[9] A. Asadi, Qing Wang, and V. Mancuso. A survey on device-to-device communication in cellular networks. *Communications Surveys Tutorials, IEEE*, 16(4):1801–1819, 2014.

[10] K. J. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 1990.

[11] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of ACM MobiSys*, 2010.

[12] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of ACM IMC '09*, pages 280–293, New York, NY, USA, 2009. ACM.

[13] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero. An open source product-oriented LTE network simulator based on ns-3. In *Proceedings of ACM MSWiM*, Ocrober-November 2011.

[14] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Communications Magazine*, 38(7):70–77, 2000.

[15] R. Bendlin, Huang Y.-F., M.T. Ivrlac, and J.A. Nossek. Fast Distributed Multi-Cell Scheduling with Delayed Limited-Capacity Backhaul Links. In *Procedings of IEEE ICC*, June 2009.

[16] Dimitri Bertsekas and Robert Gallager. *Data Networks (II Ed.)*. Prentice-Hall.

[17] T. Bonald, S. Borst, and A. Proutière. Inter-cell coordination in wireless data networks. *European Transactions on Telecommunications*, 17(3):303–312, 2006.

[18] G. Boudreau, J. Panicker, N. Guo, R. Chang, N.Wang, and S. Vrzic. Interference Coordination and Cancellation for 4G Networks. IEEE Communications Magazine, April 2009.

[19] G. Boudreau, J. Panicker, Ning Guo, Rui Chang, Neng Wang, and S. Vrzic. Interference coordination and cancellation for 4G networks. *IEEE Communications Magazine*, 47(4):74–81, April 2009.

[20] H. Cai, I. Koprulu, and N. B. Shroff. Exploiting Double Opportunities for Deadline Based Content Propagation in Wireless Networks. In *Proceedings of IEEE INFOCOM*, April 2013.

[21] Han Cai and Do Young Eun. Crossing Over the Bounded Domain: From Exponential to Power-Law Intermeeting Time in Mobile Ad Hoc Networks. *IEEE/ACM Trans. Netw.*, 17(5):1578–1591, 2009.

[22] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli. Routing, scheduling and channel assignment in wireless mesh networks: Optimization models and algorithms. *Ad Hoc Networks*, 8(6):545–563, August 2010.

[23] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, 2005.

[24] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[25] Kideok Cho, Munyoung Lee, Kunwoo Park, T.T. Kwon, Yanghee Choi, and Sangheon Pack. Wave: Popularity-based and collaborative in-network caching for content-oriented networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 316–321, March 2012.

[26] M. Cierny, Haining Wang, R. Wichman, Zhi Ding, and C. Wijting. On number of almost blank subframes in heterogeneous cellular networks. *IEEE Trans. on Wireless Comm.*, 12(10):5061–5073, Oct. 2013.

[27] M. Cierny, Haining Wang, R. Wichman, Zhi Ding, and C. Wijting. On number of almost blank subframes in heterogeneous cellular networks. *Wireless Communications, IEEE Transactions on*, 12(10):5061–5073, October 2013.

[28] E.G. Coffman Jr., M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing: A survey. *Approximation Algorithms for NP-Hard Problems*, pages 46–93, 1996.

[29] Vania Conan, Jérémie Leguay, and Timur Friedman. Characterizing pairwise inter-contact patterns in delay tolerant networks. In *Proceedings of the 1st international conference on Autonomic computing and communication systems*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[30] D.J. Daley and J. Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, UK, 2005.

[31] S. Deb, P. Monogioudis, J. Miernik, and J.P. Seymour. Algorithms for enhanced inter-cell interference coordination (eicic) in lte hetnets. *IEEE/ACM Trans. on Networking*, 22(1):137–150, Feb 2014.

[32] J. Ellenbeck, C. Hartmann, and L. Berlemann. Decentralized inter-cell interference coordination by autonomous spectral reuse decisions. In *IEEE EW 2008*, pages 1–7, 2008.

[33] FP7 European Project CROWD. WP1 Deliverable D1.1, 2014.

[34] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 2nd edition, 1990.

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

[36] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T.A. Thomas, J.G. Andrews, P. Xia, and H.S. et alii Jo. Heterogeneous cellular networks: From theory to practice. *IEEE Communications Magazine*, 50(6):54–64, 2012.

[37] Olga Goussevskaia, Roger Wattenhofer, Magns M. Halldursson, and Emo Welzl. Capacity of arbitrary wireless networks. In *INFOCOM*, pages 1872–1880, 2009.

[38] Robin Groenevelt, Philippe Nain, and Ger Koole. The message delay in mobile ad hoc networks. *Perform. Eval.*, 62(1-4):210–228, 2005.

[39] A. Hamza, S. Khalifa, H. Hamza, and K. Elsayed. A Survey on Inter-Cell Interference Coordination Techniques in OFDMA-Based Cellular Networks. *IEEE Communications Surveys Tutorials*, PP(99):1–29, 2013.

[40] Bo Han, Pan Hui, V.S. Anil Kumar, Madhav V. Marathe, Guanhong Pei, and Aravind Srinivasan. Cellular traffic offloading through opportunistic communications: A Case Study. In *Proceedings of ACM CHANTS*, 2010.

[41] Tobias Harks, Martin Hoefer, Max Klimm, and Alexander Skopalik. Computing pure nash and strong equilibria in bottleneck congestion games. *Springer Mathematical Programming*, 141(1-2):193–215, 2013.

[42] O.R. Helgason, F. Legendre, V. Lenders, M. May, and G. Karlsson. Performance of opportunistic content distribution under different levels of cooperation. In *Proceedings of European Wireless*, 2010.

[43] C.V. Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong. A control theoretic analysis of red. In *Proceedings of IEEE INFOCOM*, 2001.

[44] Liang Hu, Jean-Yves Le Boudec, and Milan Vojnović. Optimal channel choice for collaborative ad-hoc dissemination. In *IEEE INFOCOM 2010*.

[45] Stratis Ioannidis, Augustin Chaintreau, and Laurent Massoulie. Optimal and scalable distribution of content updates over a mobile social network. In *Proceedings of IEEE INFOCOM*, 2009.

[46] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel. Coordinated multipoint: Concepts, performance, and field trial results. *IEEE Communications Magazine*, 49(2):102–111, February 2011.

[47] Lei Jiang and Ming Lei. Resource allocation for eicic scheme in heterogeneous networks. In *IEEE PIMRC 2012*, pages 448–453, Sept 2012.

[48] Y. Jin and L. Qiu. Joint user association and interference coordination in heterogeneous cellular networks. *IEEE Communications Letters*, 17(12):2296–2299, December 2013.

[49] R. E. Kalman. *Topics in Mathematical System Theory*. McGraw-Hill, 1969.

[50] M. Kamel and K. M. Elsayed. Absf offsetting and optimal resource partitioning for eicic in lte-advanced: Proposal and analysis using a nash bargaining approach. In *IEEE ICC 2013*, pages 6240–6244, June 2013.

[51] M.I. Kamel and K.M.F. Elsayed. Performance evaluation of a coordinated time-domain eicic framework based on absf in heterogeneous lte-advanced networks. In *IEEE GLOBECOM 2012*, pages 5326–5331, 2012.

[52] L.V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960.

[53] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnovic. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Trans. Mob. Comp.*, 9(10):1377–1390, 2010.

[54] Lorenzo Keller, Anh Le, Blerim Cici, Hulya Seferoglu, Christina Fragouli, and Athina Markopoulou. Microcast: cooperative video streaming on smartphones. In *Proceedings of ACM MobiSys*, 2012.

[55] S.G. Kiani and D. Gesbert. Optimal and Distributed Scheduling for Multicell Capacity Maximization. *IEEE Transactions on Wireless Communications*, 7(1):288–297, January 2008.

[56] E. Y. Kim and J. Chun. Coordinated beamforming with reduced overhead for the downlink of multiuser MIMO systems. *IEEE Communications Letters*, pages 810–812, 2008.

[57] Hyojoon Kim and N. Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, February 2013.

[58] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

[59] Pekka Kyösti, Juha Meinilä, Lassi Hentilä, Xiongwen Zhao, Tommi Jämsä, Christian Schneider, Milan Narandzić, Marko Milojević, Aihua Hong, Juha Ylitalo, Veli-Matti Holappa, Mikko Alatossava, Robert Bultitude, Yvo de Jong, and Terhi Rautiainen. WINNER II Channel Models. Technical report, EC FP6, September 2007.

[60] J. LeBrun, Chen-Nee Chuah, D. Ghosal, and M. Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Vehicular Technology Conference (VTC) 2005-Spring*, volume 4, pages 2289–2293, May 2005.

[61] Kyunghan Lee, Joohyun Lee, Yung Yi, Injong Rhee, and Song Chong. Mobile data offloading: how much can WiFi deliver? In *Proceedings of ACM CoNEXT*, 2010.

[62] G. Li and H. Liu. Downlink Radio Resource Allocation for Multi-Cell OFDMA System. *IEEE Transactions on Wireless Communications*, 5(12):3451–3459, 2006.

[63] Yong Li, Guolong Su, Pan Hui, Depeng Jin, Li Su, and Lieguang Zeng. Multiple mobile data offloading through delay tolerant networks. In *Proceedings of ACM CHANTS*, 2011.

[64] Chunmei Liu and Eytan Modiano. On the performance of additive increase multiplicative decrease (AIMD) protocols in hybrid space-terrestrial networks. *Comput. Netw.*, 47(5):661–678, April 2005.

[65] D. Lu and D. Li. A Novel Precoding Method for MIMO Systems with Multi-Cell Joint Transmission. In *Proceedings of IEEE VTC 2010-Spring*, May 2010.

[66] Francesco Malandrino, Claudio Casetti, Carla-Fabiana Chiasserini, and Marco Fiore. Offloading Cellular Networks through ITS Content Download. In *Proceedings of ACM CHANTS*, 2010.

[67] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[68] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[69] National Telecommunication Regulatory Authority Funded Project. 4G++: Advanced Performance Boosting Techniques in 4th Generation Wireless Systems. WP4 Deliverable D4.1, 2012.

[70] M. C. Necker. Interference Coordination in Cellular OFDMA Networks. *IEEE Network*, 22(6):12, December 2008.

[71] M. C. Necker. A Novel Algorithm for Distributed Dynamic Interference Coordination in Cellular Networks. In *Proceedings of KiVS*, pages 233–238, 2011.

[72] Giovanni Neglia and Xiaolan Zhang. Optimal delay-power tradeoff in sparse delay tolerant networks: a preliminary study. In *Proceedings of ACM CHANTS*, pages 237–244, 2006.

[73] R. Panigrahy, K. Talwar, L. Uyeda, and U. Wieder. Heuristics for Vector Bin Packing. http://research.microsoft.com/apps/pubs/default.aspx?id=147927.

[74] Andrea Passarella and Marco Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part II*, NETWORKING'11, pages 301–313, Berlin, Heidelberg, 2011. Springer-Verlag.

[75] E. Pateromichelakis, M. Shariat, A. Ul Quddus, and R. Tafazolli. Dynamic graph-based multi-cell scheduling for femtocell networks. In *Proceedings of IEEE WCNCW*, pages 98–102, 2012.

[76] Andreea Picu and Thrasyvoulos Spyropoulos. Forecasting DTN performance under heterogeneous mobility: The case of limited replication. In *IEEE SECON 2012*.

[77] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24), 2009.

[78] C. Prehofer and C. Bettstetter. Self-organization in communication networks: principles and design paradigms. *Communications Magazine, IEEE*, 43(7):78–85, July 2005.

[79] EU Commission proposal. Horizon 2020: Eu framework programme for research and innovation. *white paper*, 2012.

[80] M. Rahman and H. Yanikomeroglu. Multicell Downlink OFDM Subchannel Allocations Using Dynamic Inter-cell Coordination. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 5220–5225, 2007.

[81] M. Rahman and H. Yanikomeroglu. Enhancing cell-edge performance: a downlink dynamic interference avoidance scheme with inter-cell coordination. *IEEE Transactions on Wireless Communications*, 9(4):1414–1425, 2010.

[82] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari. Will IPTV ride the peer-to-peer stream? *IEEE Communications Magazine*, 45(6):86 –92, 2007.

[83] S.M. Shukry, K. Elsayed, A. Elmoghazy, and A. Nassar. Adaptive Fractional Frequency Reuse (AFFR) scheme for multi-cell IEEE 802.16 e systems. In *Proceedings of IEEE HONET*, December 2009.

[84] S. Singh and J.G. Andrews. Joint resource partitioning and offloading in heterogeneous cellular networks. *IEEE Trans. on Wireless Communications*, 13(2):888–901, February 2014.

[85] K. Son, Y. Yi, and S. Chong. Utility-optimal multi-pattern reuse in multi-cell networks. , *IEEE Trans. on Wireless Communications*, 10(1):142–153, 2011.

[86] Kyuho Son, Soohwan Lee, Yung Yi, and Song Chong. REFIM: A practical interference management in heterogeneous wireless access networks. *Journal on Selected Areas in Communications*, 29(6):1260–1272, 2011.

[87] X. Tang, S.A. Ramprashad, and H. Papadopoulos. Multi-Cell User-Scheduling and Random Beamforming Strategies for Downlink Wireless Communications. In *Proceedings of IEEE VTC 2009-Fall*, Sept. 2009.

[88] Third Generation Partnership Project (3GPP). Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 application protocol (X2AP). 3GPP, technical Specification Group Radio Access Network.

[89] Third Generation Partnership Project (3GPP). Evolved Universal Terrestrial Radio Access (E-UTRA), Physical Channels and Modulation. 3GPP TS 36.211 v 10.6.0, December 2012.

[90] Third Generation Partnership Project (3GPP). Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces. 3GPP TS 29.272 v 11.5.0, December 2012.

[91] Third Generation Partnership Project (3GPP). Multimedia Broadcast/Multicast Service (MBMS). 3GPP TS 36.246 v 11.1.0, December 2012.

[92] Third Generation Partnership Project (3GPP). Physical Layer Measurements. 3GPP TS 36.214 v 11.1.0, December 2012.

[93] Third Generation Partnership Project (3GPP). Physical layer procedures (Release 10) for Evolved Universal Terrestrial Radio Access (E-UTRA). 3GPP TR 36.213 v 10.5.0, March 2012.

[94] Cheng-Lin Tsao and Raghupathy Sivakumar. On effectively exploiting multiple wireless interfaces in mobile hosts. In *Proceedings of ACM CoNEXT*, pages 337–348, 2009.

[95] Y. Wang and K.I. Pedersen. Time and power domain interference management for lte networks with macro-cells and henbs. In *IEEE VTC 2011*, pages 1–6, 2011.

[96] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan. Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays. In *Proceedings of IEEE WoWMoM*, 2011.

[97] Mingbo Xiao, N.B. Shroff, and E. K P Chong. A utility-based power-control scheme in wireless cellular systems. *Networking, IEEE/ACM Transactions on*, 11(2):210–221, Apr 2003.

[98] J. Yoon, M. Y. Arslan, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee. A distributed resource management framework for interference mitigation in ofdma femtocell networks. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 233–242, New York, NY, USA, 2012. ACM.

[99] J. Yoon, M.Y. Arslan, K. Sundaresan, S.V. Krishnamurthy, and S. Banerjee. Self-organizing resource management framework in ofdma femtocells, 2014.

[100] Jungkeun Yoon, Mingyan Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1312–1321 vol.2, 2003.

[101] W. Yu, T. Kwon, and C. Shin. Multicell coordination via joint scheduling, beamforming and power spectrum adaptation. In *Proceedings IEEE INFOCOM*, pages 2570–2578, April 2011.

[102] Jianchao Zheng, Yueming Cai, Yongkang Liu, Yuhua Xu, Bowen Duan, and Xuemin Shen. Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach. *Wireless Communications, IEEE Transactions on*, 13(12):6928–6942, Dec 2014.

[103] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007.

[104] G. Zyba, G. M. Voelker, S. Ioannidis, and C. Diot. Dissemination in opportunistic mobile ad-hoc networks: the power of the crowd. In *Proceedings of IEEE INFOCOM 2011*, 2011.