



Optimization and Self-Optimization in LTE-Advanced Networks

Abdoulaye Tall

► To cite this version:

Abdoulaye Tall. Optimization and Self-Optimization in LTE-Advanced Networks. Networking and Internet Architecture [cs.NI]. Université d'Avignon, 2015. English. NNT : 2015AVIG0208 . tel-01331039

HAL Id: tel-01331039

<https://theses.hal.science/tel-01331039>

Submitted on 6 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE

Présentée pour obtenir le grade de
Docteur en Sciences de l'Université d'Avignon

Spécialité: Informatique

Ecole Doctorale 536 «Sciences et Agro-Sciences»

Laboratoire Informatique d'Avignon (UPRES N° 4128)

Optimisation et Auto-optimisation pour les réseaux LTE

présentée par

Abdoulaye TALL

Soutenue publiquement devant le jury composé de:

Prof. CHAHED Tijani	Telecom SudParis, Paris	Examineur
Dr. COUPECHOUX Marceau	Telecom Paris-Tech, Paris	Rapporteur
Dr. GUNNARSSON Fredrik	Linköping University, Suede	Rapporteur
Dr. OUOROU Adam	Orange Labs, Issy-les-Moulineaux	Examineur
Prof. ALTMAN Eitan	INRIA, Sophia Antipolis	Directeur
Dr. ALTMAN Zwi	Orange Labs, Issy-les-Moulineaux	Co-directeur
Dr. COMBES Richard	CentraleSupélec, Gif-Sur-Yvette	Co-directeur

Laboratoire Informatique d'Avignon (EA 4128)
339 chemin des Meinajaries 84911 Avignon cedex 9

Ecole Doctorale Sciences et Agrosciences (ED536)
Maison de la Recherche
Campus Hannah Arendt, bât. nord
Bureau 0e11
74 rue Louis Pasteur
84029 Avignon cedex 1

Orange Labs
38/40 rue du General Leclerc, 92794 Issy-les-Moulineaux Cedex 9

©2015
Abdoulaye TALL
ALL RIGHTS RESERVED

Optimization and Self-optimization for LTE networks

by

Abdoulaye TALL

Abstract

Mobile operators are exploring avenues for increasing the capacity of their networks in order to face the ever increasing need of their customers for mobile data. Many options are available, each with its advantages and shortcomings. The most radical way is to purchase new frequency bandwidth licenses which is possibly the most expensive one. An upgrade to a new Radio Access Technology (RAT) is another possibility that comes with its own cost. Some of the less expensive ways consist in the densification of the network with low cost equipment. These include the deployment of small cells or the addition of new sectors using Active Antenna Systems (AASs).

Many challenges are still to be overcome in order for these approaches to bring out their full potential in increasing the networks capacity. The first one is the added complexity brought by the heterogeneity and the need to adapt to varying environments. In order to solve this problem, the concept of Self-Organizing Network (SON) has been introduced to bring autonomicity to the network elements. SON aims at enabling the nodes in the networks to automatically configure, optimize and heal themselves.

We consider self-optimization in this thesis and are interested in SON functions that address specific problems related to the performance of the two network densification strategies that are small cells and AASs. We are also interested in investigating the SON coordination problem which immediately results from simultaneously operating SON algorithms. Throughout the thesis, we rely on tools from queuing theory, stochastic approximation, convex optimization and concave games theory to model, formulate and solve the problems.

Small cells are easy to deploy because of their small size and low need in power. However, they also present a number of challenges that need to be addressed in a self-optimizing fashion. The first one being their small and fixed coverage which limits their offloading capability. We extend an existing load balancing algorithm in order to dynamically adapt the coverage of small cells according to their load and the traffic distribution around them. We also take into consideration their possibly limited backhaul capacity since they can use low speed links such as Asymmetric Digital Subscriber Line (ADSL) for their connection with the core of the network. Along with the coverage problem, there is an interference problem since the macro cells transmit at higher power so the small cell users will be vulnerable to interference. We propose here self-optimizing algorithms for two interference mitigation schemes namely the time-domain Almost Blank Subframe (ABS)-based interference coordination and the frequency-domain interference coordination. For both schemes, we consider the general class of α -fair utilities which provide a broad range of policies

for the operator to choose from.

AASs bring intelligence directly inside the antenna allowing to dynamically change the pattern of the beam emitted by the Base Station (BS). It can be used for Vertical Sectorization (VeSn) i.e. create a new vertically separated cell inside each existing cell if the antenna is composed of a vertically linear array of elements. If a matrix of antenna elements is available, Virtual Sectorization (ViSn) can be implemented i.e. the new cell can be created at any location within the macro-cell to cover a traffic hotspot. In either case, since the power budget is constant, the efficiency of activating this technology depends of the traffic distribution. For example in VeSn, the new cell (denoted as inner cell) is located near the BS. So if there is a low traffic demand near the BS, activating VeSn will result in performance degradation for the outer cell users. So we propose intelligent activation mechanisms for these features and propose interference mitigation algorithms when needed. When the number of elements in the AAS becomes large, it is possible to focus the signal beam towards each user when it is scheduled. We propose a framework for implementing this feature taking into consideration the feedback and computation costs. This framework includes a codebook design approach as well as a hierarchical beam selection algorithm that is fast enough to be implemented at the scheduler time scale.

We finally tackle the SON coordination problem by proposing a generic methodology for detecting instability in a system of concurrently operating SON algorithms, and for enforcing the stability using a coordination matrix that is computed by solving a convex optimization problem with Linear Matrix Inequality (LMI) constraints. All the solutions provided are thoroughly tested in an event-based network simulator that we developed and adapted to each use case.

Keywords: self-organizing networks, SON, stochastic approximation, convex optimization, concave games, queuing theory, LTE, LTE-Advanced, heterogeneous networks, HetNets, active antenna systems, AAS, small cells, vertical sectorization, virtual sectorization, multilevel beamforming, load balancing, backhaul-constrained load balancing, interference coordination, eICIC, SON coordination, linear matrix inequalities, LMI.

Résumé

Les opérateurs de téléphonie mobile explorent des moyens d'augmenter la capacité de leurs réseaux afin de faire face à l'explosion continue de la demande de trafic de leurs utilisateurs. Plusieurs options sont disponibles, chacune avec ses avantages et inconvénients. L'option la plus radicale est l'achat de nouvelles licences de spectre de fréquence qui représente un investissement énorme. La mise à niveau des équipements vers une nouvelle technologie plus efficace spectralement constitue une autre possibilité qui a aussi un coût élevé. Parmi les solutions moins coûteuses figure la densification du réseau avec des équipements low-cost. Il s'agit notamment des *small cells* ou encore des antennes actives permettant l'ajout de nouveaux secteurs sans déployer de nouveaux sites.

Plusieurs difficultés restent à surmonter afin de tirer le maximum de ces possibilités. En premier il y a la complexité accrue due à l'hétérogénéité qu'apportent ces solutions au réseau ainsi que la nécessité d'adapter la configuration du réseau à un environnement changeant. Le concept SON (réseau auto-organisant) fut introduit pour résoudre ce problème. SON a pour objectif de rendre les éléments autonomes c'est-à-dire qu'ils pourront s'auto-configurer, s'auto-optimiser ou encore s'auto-réparer.

Nous nous intéressons plus particulièrement à l'auto-optimisation dans cette thèse et proposons des algorithmes SON répondant à des problématiques spécifiques liées à la performance des deux stratégies de densification que sont les *small cells* et les antennes actives. Nous étudions aussi le problème de la coordination SON qui est une conséquence immédiate de l'opération simultanée de plusieurs algorithmes SON en parallèle. Tout au long de la thèse, nous utilisons des outils provenant de la théorie des files d'attente, de l'approximation stochastique, de l'optimisation convexe et des jeux concaves afin de modéliser, formuler et résoudre les problèmes abordés.

Les *small cells* sont faciles à déployer du fait de leur petite taille et de leur faible besoin en puissance électrique. Cependant, elles présentent aussi de nombreux défis à relever afin de les rendre plus autonomes. Le premier provient de leur faible couverture fixe qui limite leur capacité à décharger les macro-cellules qu'elles sont censées décongestionner. Nous adaptons un algorithme d'équilibrage de charge existant afin qu'il puisse ajuster dynamiquement la couverture des *small cells* en fonction de leur charge et la distribution du trafic dans leur voisinage. Nous intégrons aussi dans cet algorithme la possibilité d'une limitation de la capacité du lien backhaul vers le cœur du réseau qui peut être une liaison faible débit telle que l'ADSL. En plus du problème de couverture s'ajoute le problème d'interférence qui vient du fait que les macro-cellules ont une puissance

de transmission beaucoup plus élevée, ce qui rend les utilisateurs des small cells vulnérables à l'interférence des macro-cellules. Nous proposons ici des algorithmes auto-optimisants pour les deux méthodes de coordination d'interférence dans le domaine temporel (basé sur le mécanisme de l'*Almost Blank Subframe*) et dans le domaine fréquentiel. Pour chacune des deux méthodes, nous considérons toute la classe des utilités α -fair ce qui donne un large choix de politiques à l'opérateur.

Les antennes actives introduisent l'intelligence directement dans les antennes, ce qui permet de changer dynamiquement le diagramme des faisceaux émis par la station de base. Cette technologie peut être utilisée pour la sectorisation verticale qui consiste à créer une nouvelle cellule séparée verticalement dans chaque cellule existante à l'aide d'une colonne d'éléments d'antenne. Lorsqu'une matrice d'éléments d'antennes est disponible, la nouvelle cellule peut être créée à tout endroit dans la cellule macro d'origine pour couvrir par exemple un hotspot de trafic, on parle alors de sectorisation virtuelle. Dans tous les cas, étant donné que le budget de puissance ne change pas lorsqu'on déploie la technologie des antennes actives, sa performance dépend de la distribution du trafic. Par exemple pour la sectorisation verticale, la nouvelle cellule est créée proche de la station de base. Il faut donc qu'il y ait suffisamment de trafic dans cette zone sinon les performances du système peuvent se dégrader. Nous proposons alors des mécanismes d'activation intelligente de ces fonctionnalités ainsi que des algorithmes de coordination d'interférence pour les cas où cela est nécessaire. Avec un grand nombre d'éléments dans la matrice d'antennes, il est possible de focaliser le signal sur chaque utilisateur. Nous proposons un cadre d'implémentation de cette possibilité prenant en compte les coûts liés aux retours d'information et aux calculs. Ce cadre inclut une approche de design du codebook ainsi qu'un algorithme itératif de sélection de faisceau qui est suffisamment rapide pour être implémenté à la même périodicité qu'un ordonnanceur.

Finalement, nous abordons le problème de coordination d'algorithmes SON en proposant une méthodologie permettant de détecter l'instabilité dans un système d'algorithmes fonctionnant simultanément et un mécanisme de stabilisation utilisant une matrice de coordination obtenue en résolvant un problème d'optimisation convexe avec des contraintes d'inégalités matricielles linéaires liées à l'architecture du réseau. Toutes les solutions proposées ont été minutieusement testées dans un simulateur évènementiel de réseau mobile que nous avons développé et adapté à chaque scénario.

Mots-clés: réseaux auto-organisant, SON, approximation stochastique, optimisation convexe, jeux concaves, theorie des files d'attente, LTE, LTE-Advanced, réseaux hétérogènes, antennes actives, small cells, sectorisation verticale, sectorisation virtuelle, beamforming hiérarchique, équilibrage de charge, équilibrage de charge avec limitation du lien backhaul, coordination d'interférence, eICIC, coordination SON, inégalités matricielles linéaires.

A la mémoire de ma grand mère et mon grand père.

Preface

This thesis is the result of the research I performed between December 2012 and December 2015 under the supervision of Zwi ALTMAN, Eitan ALTMAN and Richard COMBES.

I started the thesis right after my graduation from Tunisia Polytechnic School having had a first research experience during my graduation internship at King Abdullah University of Science and Technology (KAUST). So my knowledge in mobile networks was not very deep let alone my skills in stochastic approximation, queuing theory or convex optimization. But I was ready for the challenge and I gratefully thank Zwi ALTMAN for giving me the opportunity to deepen my knowledge and contribute to the advancement of the mobile technology in an industrial environment.

During my thesis, I mostly worked in Orange Labs premises in Issy Les Moulineaux, France. I was initially in the Radio Engineering for Mobile (REM) team which then merged with the Radio CEC, Performance, Optimisation and Tools (RCT) team to produce the Radio Engineering and Support (RES) team. In short, a lot of change happened in the work organization during the thesis but I remained surrounded by brilliant colleagues who inspired, supported and motivated me through these 3 years to make the journey even more enjoyable. I particularly enjoyed the different technical presentations and the discussions with the colleagues which were the occasion to broaden my knowledge in the research areas of mobile networks as well as the operational aspects.

I grew a lot during this thesis intellectually but also personally and I would like to thank all those who helped through these three years starting with Zwi ALTMAN. Thank you for your teaching, your guidance, your patience and your hard work which truly inspired me. I would also like to express my gratitude to Eitan ALTMAN for his strong and essential theoretical contribution in this thesis. And yes you have guessed right, they are brothers which made the thesis as smooth as a baby's cheek. I also thank Richard COMBES who inspired me and helped me get well in this thesis. I thank all my colleagues (permanents, PhD students, interns, apprentices and Post-Docs) for bearing with me during my thesis and making my stay full of positive experience.

Finally, I would like to thank my family and friends for supporting me during all my studies leading to this thesis. I will now be supporting you too.

Abdoulaye TALL
Issy-Les-Mx, Sept, 15th, 2015

Contents

Abstract	v
Résumé	vii
List of Figures	xvii
List of Tables	xix
List of Notations	xxi
List of Acronyms	xxiii
1 Introduction	1
1.1 The context	1
1.2 Thesis objectives	3
1.3 Our Contributions	4
1.4 List of publications	6
2 Mathematical background	9
2.1 Introduction	10
2.2 Reminder on probability theory	10
2.2.1 Exponential distribution	10
2.2.2 Poisson process	10
2.2.3 Channel models	11
2.2.4 Martingales	11
2.3 Queuing theory	12
2.3.1 Markov chains and Markov processes	12
2.3.2 Performance evaluation for queues of interest	14
2.3.3 Wireless downlink performance	15
2.4 Convex optimization	16
2.4.1 Notion of convexity	16
2.4.2 Convex optimization problem	17
2.4.3 KKT optimality conditions	18
2.4.4 Algorithms	18

2.5	Stochastic Approximation	19
2.5.1	Robbins-Monro algorithm	20
2.5.2	Kiefer-Wolfowitz algorithm	20
2.5.3	A sketch of the convergence proof	21
2.5.4	Constant step size selection	21
2.5.5	Projected Stochastic Approximation (SA)	22
2.6	Diagonal strict concavity	22
3	Self-optimizing strategies for heterogeneous networks	23
3.1	Introduction	24
3.2	Load balancing	25
3.2.1	Literature overview	25
3.2.2	Infinite-capacity backhaul heterogeneous networks scenario	27
3.2.3	Backhaul-constrained heterogeneous networks scenario	29
3.2.4	Validation through simulation results	31
3.3	ABS ratio optimization	35
3.3.1	ABS mechanism	35
3.3.2	Implementation alternatives	35
3.3.3	Exact Proportional-Fair algorithm	38
3.3.4	Lower-bound approximated PF algorithm	40
3.3.5	General α -fair algorithm	42
3.3.6	Load minimization algorithm	42
3.4	Frequency splitting optimization	44
3.4.1	Exact α -fair algorithms	45
3.4.2	Lower-bound α -fair algorithms	47
3.4.3	eICIC algorithms implementation: Centralized or Distributed	48
3.5	Numerical results	49
3.5.1	Deployment scenario	49
3.5.2	Static Analysis - Full Buffer performance	50
3.5.3	Dynamic analysis	53
3.6	AUTOSDN framework	58
3.7	Conclusion	61
4	Self-optimization for active antenna systems	63
4.1	Introduction	64
4.2	Vertical sectorization	65
4.2.1	Problem formulation	65
4.2.2	Analytical approach to VeSn activation and calibration with realistic measurements	68
4.2.3	Frequency splitting approach	71

4.3	Virtual sectorization	80
4.3.1	Antenna model	81
4.3.2	System description	83
4.3.3	Self-optimizing algorithms	84
4.3.4	Numerical results	84
4.4	Hierarchical beamforming	89
4.4.1	Antenna design methodology	89
4.4.2	Multilevel beamforming algorithm	90
4.4.3	Numerical results	92
4.5	Conclusion	102
5	Coordination of SON algorithms	103
5.1	Introduction	104
5.2	Problem Description	104
5.2.1	General Problem	104
5.2.2	Linear Case	105
5.3	Coordination Mechanism	106
5.3.1	ODE stabilization	106
5.3.2	Fully distributed coordination	108
5.3.3	Stochastic Control Stabilization	109
5.4	SON Coordination use case: Application to wireless networks	110
5.4.1	System Model	110
5.4.2	Numerical Results	112
5.5	Conclusion	118
6	Conclusion	119
6.1	Summary of contribution	119
6.2	Directions for future work	120
A	Theorem proofs for chapter 3	123
A.1	Proof of Theorem 2	123
A.2	Proof of Theorem 3	124
A.3	Proof of Theorem 4	124
A.4	Proof of Theorem 5	124
A.5	Proof of Theorem 6	125
A.6	Proof of Theorem 7	126
B	Event-based Matlab simulator	129
	Bibliography	133

List of Figures

3.1	Illustration of Cell Range Extension in a HetNet	27
3.2	Network Layout	31
3.3	Cell Individual Offsets	33
3.4	Local loads using Eq. (3.6) (dashed lines) and Eq. (3.7) (plain lines)	33
3.5	Global loads using Eq. (3.9) (dashed lines) and Eq. (3.11) (plain lines)	33
3.6	File Transfer Times comparison with Local SON or Global SON	34
3.7	Time evolution of MUT and CET for the cluster of the macro BS and the 4 small cells	34
3.8	Illustration of Almost Blank Sub-Frames in a HetNet	35
3.9	Maximum Cell Individual Offset (CIO) for SINR improvement and Maximum SINR gain as a function of the number of macro BSs applying ABSs	37
3.10	Time evolution of ABS ratio using stochastic approximation (solid line) and optimal solution (dashed line)	40
3.11	Asynchronous ABS/Frequency Splitting Example 1	49
3.12	Asynchronous ABS/Frequency Splitting Example 2	49
3.13	Network layout scenario	50
3.14	Traffic scenario for Full Buffer simulation	52
3.15	Full buffer Cell Edge Throughputs	52
3.16	Full buffer Mean User Throughputs	53
3.17	Full buffer Geometric Mean Throughputs	53
3.18	Comparison of the maximum loads among the macro - and small cells	54
3.19	Cell Edge Throughputs with elastic traffic	55
3.20	Mean User Throughputs with elastic traffic	55
3.21	File Transfer Times with elastic traffic	56
3.22	Geometric Mean Throughputs with elastic traffic	56
3.23	Users Throughputs CDF: Exact PF utility (3.25) vs Lower bound PF utility (3.27) . .	57
3.24	AutoSDN overall architecture.	59
3.25	UMF dashboard view of the network	59
3.26	Coverage maps	60
3.27	Parameters and KPIs view from the SDN controller	60
4.1	BS architecture evolution	64
4.2	Vertical Sectorization illustrated	66

4.3	Inner activation decision in the inner/outer sector load plane.	70
4.4	Inner deactivation decision in the inner/ outer sector load plane.	70
4.5	VeSn (de)activation decision boundaries in the inner/outer sector load plane.	71
4.6	Evolution of the traffic distribution in inner/outer cells' areas.	72
4.7	VeSn activation Decisions over time.	72
4.8	Mean User Throughput over time.	73
4.9	Simulation scenario network layout	77
4.10	Mean user throughput for increasing arrival rates	79
4.11	Cell edge user throughput for increasing arrival rates	79
4.12	Maximum loads for increasing arrival rates	80
4.13	Network layout with ViSn enabled	80
4.14	Antenna array with dipole radiating elements.	81
4.15	Traffic profile over time (HH:MM means hours:minutes)	85
4.16	Best server map	86
4.17	Mean user throughput evolution over time	87
4.18	Cell edge user throughput evolution over time	87
4.19	Maximum loads' (all cells, virtual and macro) evolution over time	88
4.20	File transfer time's evolution over time	88
4.21	Example of beam hierarchy	91
4.22	Traffic intensity map (in users/s/km ²).	94
4.23	Multilevel beam coverage maps for the mass event scenario	95
4.24	Successive narrowing of the beam for a given user	96
4.25	Antenna diagrams for a given user's best beam in each level.	97
4.26	Histogram of selected beams throughout the simulation: X is the beam number (see Figure 4.23) and Y - its selection probability.	98
4.27	User throughput CDFs comparison between optimal, hierarchical and no beamforming.	100
4.28	Coverage maps for different beamforming levels for the rural scenario	101
5.1	Coordination system block diagram	108
5.2	19 cells hexagonal network with wrap-around	112
5.3	Impact of Coordination on Loads	115
5.4	Impact of Coordination on Coverage Probability	115
5.5	Impact of Coordination on Blocking Rate	116
5.6	Stationary KPIs with all SON functions equally weighted	116
5.7	Stationary KPIs with outage probability prioritized	117
B.1	General simulator architecture	130

List of Tables

3.1	Network and Traffic Parameters	32
3.2	Network and Traffic characteristics	51
3.3	Traffic characteristics	54
4.1	Network and Traffic characteristics	78
4.2	Network and Traffic characteristics	85
4.3	ViSs antenna configurations	86
4.4	Network and Traffic characteristics	93
4.5	Network and traffic characteristics for the mass event scenario	98
4.6	Performance gain using multilevel beamforming for the mass event scenario	99
4.7	Network and traffic characteristics for the rural scenario	102
4.8	Performance gain using multilevel beamforming for the rural scenario for different beam levels	102
5.1	Network and Traffic characteristics	113

List of Notations

$[\cdot]_S^+$	Projection on the set S
$[A]_{k,k}$	k^{th} order leading principal submatrix of A
$\dot{\alpha}$	Derivative of α over time
$\mathbb{E}(\cdot)$	Expectation of a random variable
$\mathbb{P}(e)$	Probability of event e
$\mathbb{1}_{\{cond\}}(x)$	Indicator function on the set of x values satisfying the condition $cond$
$\nabla_x f$	Gradient of f with regard to x
$\text{tr}(\cdot)$	Trace of a matrix
θ^*	Equilibrium points of system comprising control loops - Also optimal parameters
$A \prec 0$	A is a negative definite matrix
A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
I	The identity matrix i.e. diagonal matrix with ones on the diagonal
$\det(\cdot)$	Determinant of a matrix
$\text{eig}(A)$	Eigenvalues of A
$JF(\cdot)$	Jacobian of $F(\cdot)$
$\text{dom}f$	The domain of function f
$k!$	Factorial of integer number k
$\ \cdot\ $	Frobenius norm
$[a, b)$	Interval of real numbers between a and b including a excluding b
\emptyset	Empty set

int S The interior of the set S

\mathbb{X}_+ Subset of positive elements of \mathbb{X}

$h \circ g$ Composition of functions h and g defined by $(h \circ g)(x) = h(g(x))$

List of Acronyms

3GPP	3rd Generation Partnership Project	2
a.s	almost surely	21
AAS	Active Antenna System	4
ABSrO	ABS ratio optimization	25
ABSr	ABS ratio	5
ABS	Almost Blank Subframe	5
ADSL	Asymmetric Digital Subscriber Line	5
AFUA	α -Fair User Association	26
ANR	Automated Neighbour Relationship	2
API	Application Programming Interface	58
AutoSDN	Autonomic SDN	58
BBU	Baseband Unit	64
BS	Base Station	5
c.d.f	cumulative distribution function	10
CAPEX	Capital Expenditures	1
CCD	Co-Channel Deployment	44
CCO	Coverage and Capacity Optimization	3
CET	Cell-Edge throughput	32
CIO	Cell Individual Offset	3
CRE	Cell Range Extension	24
D2D	Device-To-Device	121
dB	decibels	111
DL	Downlink	6
eICIC	enhanced Inter Cell Interference Coordination	5
FDD	Frequency Division Duplex	6

FFR	Fractional Frequency Reuse	3
FIFO	First In First Out	14
FL	Fractional Load	3
FTP	File Transfer Protocol	110
FTT	File Transfer Time	29
GBR	Guaranteed Bit-Rate	27
GMT	Geometric Mean user Throughput	51
HetNet	Heterogeneous Network	4
HO	Handover	2
HSPA	High Speed Packet Access	1
ICIC	Inter-Cell Interference Coordination	3
KKT	Karush-Kuhn-Tucker	18
KPI	Key Performance Indicator	2
LB	Load Balancing	5
LHP	Left Half Plane	106
LHS	Left Hand Side	21
LIFO	Last In First Out	14
LMI	Linear Matrix Inequality	107
LoS	Line of sight	11
LTE	Long Term Evolution	6
MAB	Multi-Armed-Bandit	120
MDT	Minimization of Drive Tests	2
MIMO	Multiple Input Multiple Output	6
MLB	Mobility Load Balancing	3
MRO	Mobility Robustness Optimization	3
MUT	Mean User Throughput	32
NEM	Network Empowerment Mechanism	58
NMS	Network Management System	2
NRT	Neighbour Relationship Table	2
ODE	Ordinary Differential Equation	21
OD	Orthogonal Deployment	44

OLRE	Orange Labs Research Exhibition	5
OPEX	Operational Expenditures	2
p.d.f	probability density function	10
PCI	Physical Cell Identity	2
PC	Power Consumption	98
PEC	Perfect Electrical Conductor	81
PF	Proportional Fair	39
PRACH	Physical RACH	3
PRB	Physical Resource Block	76
PS	Processor Sharing	14
QoS	Quality of Service	55
RACH	Random Access Channel	3
RAN	Radio Access Network	29
RAT	Radio Access Technology	3
RF	Radio Frequency	64
RHS	Right Hand Side	106
RRH	Remote Radio Head	64
RSRP	Reference Signal Received Power	69
SA	Stochastic Approximation	10
SDN	Software-Defined Networking	2
SfO	Split factor Optimization	
SFR	Soft Frequency Reuse	3
SINR	Signal to Interference plus Noise Ratio	25
SON	Self-Organizing Network	1
TDD	Time Division Duplex	6
UA	User Association	26
UE	User Equipment	2
UL	Uplink	3
UMF	Unified Management Framework	58
VeSn	Vertical Sectorization	5
ViSn	Virtual Sectorization	5

ViS	Virtual Sector	64
WCDMA	Wideband Code Division Multiple Access	1
WLAN	Wireless Local Area Network	25
DPS	Discriminatory Processor Sharing	120

Chapter 1

Introduction

This chapter introduces the subject of this thesis. We begin by presenting the technological context of the thesis focusing on its topic which is the Self-Organizing Network (SON) concept. Then we expose the main objectives of the thesis and finally list all the original contributions made throughout the thesis.

1.1 The context

The massive adoption of smartphone usage in the last decade has spurred tremendous need for mobile data. The networks operators have been facing an exponentially increasing demand for better data connectivity from their customers. In order to answer this demand, new mobile technologies were developed and enhanced starting with Wideband Code Division Multiple Access (WCDMA) further improved with High Speed Packet Access (HSPA) and then the introduction of 4G LTE-Advanced. Throughout these technologies, many features have been added to address specific topics such as small cells which address uncovered areas or traffic hotspots. These evolutions have added more and more complexity to the mobile networks making their management and operation difficult for the operators.

In order to alleviate the management of mobile networks and allow them to further evolve, the SON concept has been introduced. SON aims at introducing autonomicity in the network by automating repetitive or complex operations. SON algorithms can be grouped into three categories:

Self-configuration algorithms

They enable plug-and-play deployment of new network elements by automating their initial configuration. Self-configuration includes software download and update, network discovery for authentication, IP address assignment, core network association, neighbour listing and coverage parameters setting. This SON feature helps reduce Capital Expenditures (CAPEX) and also limits the introduction of human error in the installation of network elements. The feature is particularly necessary in some cases such as the deployment of femto-cells in order

to avoid the need for an expert in the field, the customer can thus install the equipment himself.

Self-optimizing algorithms

The network optimization is one of the most time consuming activities in the operation of a network. Self-optimizing algorithms help to reduce that burden by automating some optimization tasks thus reducing Operational Expenditures (OPEX). This SON feature monitors Key Performance Indicators (KPIs) and adjusts parameters in order to improve those KPIs. Depending on the scope and the time scale of the optimization, the self-optimizing algorithm can be located in the Network Management System (NMS) (centralized SON) or directly within the network elements (distributed SON).

Self-healing algorithms

They enable the network elements to autonomously detect their own failures and repair themselves. This feature helps reduce the network operation costs by avoiding manual intervention whenever a failure occurs. It also limits the consequences of failures by bringing faster reactivity in addressing those failures. The healing consists in locally solving the issue causing the failure or using neighbouring nodes to replace the faulty node's function until it is back online.

Minimization of Drive Tests (MDT) is considered a SON feature even though it does not directly fit into one of these categories, instead it provides meaningful information that can be used to perform either of them. It consists in building an accurate representation of the network state in terms of coverage or traffic by collecting data from the User Equipments (UEs) [19].

SON algorithms can be provided by equipment vendors (especially the ones implemented inside the network nodes), network operators via their NMS or third-parties which interface their tools with the NMS. The Software-Defined Networking (SDN) paradigm is an interesting way to introduce SON functions in a flexible and vendor agnostic way. In general, SON algorithms do not need standardization except the ones that require action from the UEs such as MDT. Nonetheless, some use cases for SON in LTE-Advanced have been identified in standardization bodies such as 3rd Generation Partnership Project (3GPP) [2]. These are listed below along with their objective and parameters.

Automated configuration of Physical Cell Identity (PCI)

Automatically configure the PCI of a newly introduced cell to avoid collision (two cells covering the same area having the same PCI) and/or confusion (two neighbouring cell having the same PCI).

Automated Neighbour Relationship (ANR)

Automatically manage and optimize the neighbour relations by updating the Neighbour Relationship Tables (NRTs). This SON algorithm mostly makes use of Handover (HO) events.

Coverage and Capacity Optimization (CCO)

Provide both optimal coverage and capacity by optimizing antenna tilt, azimuth or transmit power. Since these are contradictory objectives, a trade-off must be made according to the operator's priorities.

Energy Savings

Save energy by switching off some network elements e.g. some Radio Access Technologies (RATs) at certain times in the day depending on the traffic demand.

Interference reduction

Reduce interference in the network by switching off strongly interfering cells on the same RAT.

Inter-Cell Interference Coordination (ICIC)

Reduce inter-cell interference by optimizing transmit power and frequency allocations. Many schemes have been proposed in the literature including Fractional Load (FL), Fractional Frequency Reuse (FFR) and Soft Frequency Reuse (SFR) [24].

Mobility Robustness Optimization (MRO)

Detect and minimize HO-related failures (too early/late HO, HO to wrong cell or unnecessary HO) by tweaking HO-related parameters (Hysteresis, Time to Trigger, Cell Individual Offset (CIO), Cell reselection parameters).

Mobility Load Balancing (MLB)

Optimisation of HO-related parameters in order to balance the traffic across the network (intra-RAT or inter-RAT).

Random Access Channel (RACH) optimization

Minimize the delay for accessing the network, minimize the Uplink (UL) interference due to RACH and minimize interference among RACH attempts by optimizing RACH-related parameters (Physical RACH (PRACH) configuration index, RACH preamble split, RACH backoff parameter value, PRACH transmission power control parameters, etc.).

3GPP has only provided some use cases description for SON focusing on the performance objectives and the parameters to optimize. The actual implementation and algorithms are left to SON providers which can also provide other SON use cases. In this PhD thesis, we are interested in designing actual SON algorithms for specific use cases related to network densification and studying their performance.

1.2 Thesis objectives

The objective of this thesis is to use several mathematical tools such as queuing theory, stochastic approximation and convex optimization to model the behavior of the radio interface of mobile

networks at link and network level and propose self-optimizing algorithms for specific use cases mostly related to network densification. We target distributed solutions in order to get the highest reactivity to network environment and traffic changes.

The implementation of SON algorithms in real networks poses many challenges including the trust of the operator in their efficiency and stability. Thus we aim at developing SON algorithms that comply with the following quality criteria:

- The algorithms should be kept as **simple** as possible with the update being only one equation. This allows to reduce the computational power needed in the network elements where the algorithms will be deployed and more importantly facilitate the understanding of the algorithms by the network engineers who will implement them. Simplicity also implies to reduce information exchange among network entities as much as possible.
- The algorithms should come with **strong, mathematically provable, performance guarantees** using either tools from control theory, game theory or convex optimization. This requirement is essential in the adoption of SON by network operators who cannot rely on mere simulation results to entrust their whole business to an algorithm. The occurrence of a single failure can bring about huge losses including the distrust of customers. Stability must also hold when several algorithms are operating simultaneously.
- Since we focus on distributed SON algorithms, the parameters can be modified at a fast time scale (in the order of seconds). The algorithms must take advantage of this and thus provide **very fast convergence** in order to track the environment changes. A load balancing algorithm for example, should be able to track traffic variations in the order of a minute.
- The stochastic nature of wireless systems impose that the SON algorithms be **robust to noise** so they must be designed taking noise into consideration from the start. We apply stochastic approximation results in order to comply with this requirement. The convergence and the stability must both be valid under noisy environment.

In order to assess the actual performance of the proposed algorithms, another objective of this thesis is to develop and evolve a network simulator in which all proposed algorithms will be tested. This simulator must mimic as much as possible real network behavior especially the dynamicity of the traffic and environments.

1.3 Our Contributions

The contributions in this thesis can be regrouped into three main parts: SON for Heterogeneous Networks (HetNets), SON for Active Antenna Systems (AASs) and SON coordination.

In the HetNets part (Chapter 3), we tackled some important issues that occur in a network comprising nodes with different characteristics (e.g. different transmit powers). We are particularly

interested in a network with macro cells and small cells which have different coverage and produce different inter-cell interference.

The purpose of network densification with small cells is to offload the heavily loaded macro cells. However, the small coverage of small cells limits this offloading. We proposed a Load Balancing (LB) algorithm for HetNets based on the one provided in [23]. We also extended this algorithm in order to take into consideration possible backhaul limitations that are particularly relevant for small cells which can be deployed with low capacity backhubs such as Asymmetric Digital Subscriber Line (ADSL).

Extending the coverage of small cells makes them vulnerable to interference. So we proposed self-optimizing algorithms for the ABS ratio (ABSR) which is involved in the enhanced Inter Cell Interference Coordination (eICIC) proposed in 3GPP. We also designed and compared alternative frequency domain interference mitigation algorithms with the time-domain Almost Blank Subframe (ABS)-based eICIC algorithms. The objective function we used for all these interference coordination optimization algorithms is the well-known general class of α -fair utility of users throughputs. We also proposed an interference mitigation algorithm based on ABS but with the objective of minimizing the maximum of loads in the system. Extensive numerical results using our own simulator were provided to support the efficiency of the proposed algorithms.

We integrated some of the results in this part into a demonstrator for the AutoSDN project showcased in Orange Labs Research Exhibition (OLRE). This demonstrator provides a novel framework allowing the flexible implementation of SON algorithms in mobile networks. Several papers were also published including [71], [59] and [77].

The second part (Chapter 4) is concerned with network densification solutions based on AASs which provide a way to add more capacity in the network without creating new Base Stations (BSs)' sites. AASs constitute an evolution of traditional antennas allowing more flexibility in the configuration of coverage areas and frequency reuse schemes.

The first evolution brings us to Vertical Sectorization (VeSn) which allows to create two vertically separated cells by using an additional downtilted antenna that covers the inner part of the original cell. The users in the inner area are then served with increased antenna gain and the frequency bandwidth can be reused in the inner cell. However, adding one more cell vertically brings also more interference and since the power budget is conserved, reduced transmit power is available for both cells. We provided self-optimizing algorithms that enable the smart activation of this feature ensuring no performance degradation after the activation. We also provided dynamic frequency reuse schemes that allow to take advantage of the downtilted antenna gain in all traffic conditions.

The next evolution is the use of antenna arrays allowing the creation of the inner cell anywhere in the original cell (Virtual Sectorization (ViSn)) in order to cover a traffic hotspot for example. We also provided in this case a self-optimizing algorithm for frequency-based interference coordination scheme along with a switching mechanism to a full frequency reuse when the traffic demand is high enough.

When the number of antenna elements in the antenna array is further increased (to the hundreds) we fall into the domain of Massive Multiple Input Multiple Output (MIMO) when the transmitted beam can be tailored to each user. In this case, we provided a multilevel beamforming algorithm that allows to reduce the feedback needed for the efficient utilization of the antenna array in Frequency Division Duplex (FDD) systems. The algorithm can also be used in Time Division Duplex (TDD) systems in order to reduce the processing at the BS.

All the algorithms were thoroughly tested in our simulator that we evolved in order to implement these new features. The following papers were published on this part [72],[73], [75] and [76].

The third part (Chapter 5) deals with the SON coordination problem. Operation of several SON algorithms in parallel is not provably globally stable since the SON algorithms are generally designed in a stand-alone fashion and possibly by different SON providers. It is thus necessary to provide tools to detect any instability and coordinate the SON mechanisms if needed. A novel approach to solving this problem was proposed in [26] based on the concave games theory developed in [63]. In this thesis, we extend the approach in [26] to provide a generic methodology for the coordination of SON algorithms. We apply the methodology proposed to a Long Term Evolution (LTE) use case with three different SON algorithms deployed in several BSs and show its efficiency. The results in this part were published in [74].

It is noted that all the proposed algorithms are studied and their performance is evaluated for the Downlink (DL) but they can be easily adapted for the UL.

1.4 List of publications

Journal papers

- [J1] **A. Tall**, R. Combes, Z. Altman, and E. Altman, “Distributed Coordination of Self-Organizing Mechanisms in Communication Networks,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 328–337, Dec. 2014.
- [J2] **A. Tall**, Z. Altman, and E. Altman, “Self-optimizing load balancing with backhaul constrained radio access networks,” *IEEE Wireless Communication Letters*, 2015.

Magazine paper

- [M1] K. Tsagkaris, G. Poullos, P. Demestichas, **A. Tall**, Z. Altman, “An open framework for programmable, self-managed radio access networks,” *Communications Magazine, IEEE*, vol. 53, no. 7, pp. 154–161, 2015.

Conference papers

- [C1] **A. Tall**, Z. Altman, and E. Altman, “Self organizing strategies for enhanced ICIC (eICIC),” in *2014 12th WiOpt*, May 2014, pp. 318–325.
- [C2] —, “Self-optimizing Strategies for Dynamic Vertical Sectorization in LTE Networks,” in *2015 IEEE WCNC*, New Orleans, USA, Mar. 2015.
- [C3] —, “Virtual sectorization: design and self-optimization,” in *2015 IEEE VTC Spring Workshop IWSON*, Glasgow, Scotland, May 2015.
- [C4] H. Sidi, Z. Altman, and **A. Tall**, “Self-optimizing mechanisms for EMF reduction in heterogeneous networks,” in *2014 12th WiOpt*, May 2014, pp. 341–348.
- [C5] K. Trichias, R. Litjens, **A. Tall**, Z. Altman, and P. Ramachandra, “Performance evaluation & SON aspects of vertical sectorisation in a realistic LTE network environment,” in *2014 11th ISWCS*, Aug. 2014, pp. 131–137.
- [C6] —, “Self-optimisation of vertical sectorisation in a realistic LTE network,” in *European Conference on Networks and Communications (EuCNC)*. IEEE, 2015, pp. 149–153.

Technical report

- [T1] **A. Tall**, Z. Altman, and E. Altman, “Multilevel beamforming for high data rate communication in 5G networks,” *arXiv preprint arXiv:1504.00280*, 2015.

Submitted

- [S1] **A. Tall**, Z. Altman, and E. Altman, “On beam planning for low complexity multilevel beamforming with large antenna arrays,” *submitted to IEEE TVT*, 2015.
- [S2] T. Maroua, **A. Tall**, Z. Hind, and Z. Altman, “Multilevel beamforming for mass event scenario,” *submitted to IEEE WCNC*, 2016.
- [S3] F. Salem, **A. Tall**, Z. Altman, and A. Gati, “Energy consumption optimization in 5G networks using multilevel beamforming and large scale antenna systems,” *submitted to IEEE WCNC*, 2016.

Chapter 2

Mathematical background

“If a ‘religion’ is defined to be a system of ideas that contains unprovable statements, then Gödel taught us that mathematics is not only a religion, it is the only religion that can prove itself to be one.”

– John Barrow

2.1 Introduction

In this chapter, we introduce the various mathematical tools used throughout the thesis along with key references that the reader can refer to for more details. The random nature of many aspects in wireless communications require extensive use of probability theory. We present some probability distributions which model the wireless traffic and channels in Section 2.2. The link level performance of a wireless system is generally studied using queuing theory which we address in Section 2.3. Section 2.4 deals with convex optimization theory which is the primary tool used throughout the thesis in order to formulate and solve optimization problems and also give the corresponding optimization algorithms. Taking into account randomness in the optimization problems, we extensively use results from Stochastic Approximation (SA) theory which are recalled in Section 2.5.

2.2 Reminder on probability theory

We recall some probability distributions of interest in this section.

2.2.1 Exponential distribution

A real random variable X follows the exponential distribution of parameter λ if its probability density function (p.d.f) is

$$f(x) = \lambda \exp(-\lambda x), \forall x \in \mathbb{R}_+. \quad (2.1)$$

Equivalently, its cumulative distribution function (c.d.f) is defined as

$$\mathbb{P}(X \leq x) = F(x) = 1 - \exp(-\lambda x), \forall x \in \mathbb{R}_+. \quad (2.2)$$

The exponential distribution possesses a memoryless property in the sense that

$$\mathbb{P}(X > s + t | X > s) = \mathbb{P}(X > t), \forall s, t \in \mathbb{R}_+. \quad (2.3)$$

2.2.2 Poisson process

A point process over \mathbb{R}_+ is an increasing series of positive random variables T_1, T_2, \dots . By assuming $T_0 = 0$, we denote by $\tau_1, \tau_2, \tau_3, \dots$ the inter-arrival times of the point process with $\tau_n = T_n - T_{n-1}$. A point process is a Poisson process of intensity λ if its inter-arrival times follow the exponential distribution of parameter λ .

The superposition of K independent Poisson processes with intensities $\lambda_1, \dots, \lambda_K$ is a Poisson process with intensity $\lambda = \lambda_1 + \dots + \lambda_K$.

2.2.3 Channel models

The wireless channel is random by nature due to the many obstacles the signal encounters in its path. Several models have been devised in the literature in order to give a simple representation of the random attenuation experienced by the signal. The model to be used depends on the environment. The reader is referred to [68, Chapter 2] for an extended discussion on the channel models. We focus here on a few models that cover almost all cases of interest.

For a dense urban environment with rich scattering (lots of reflexions) and no Line of sight (LoS) path, the Rayleigh distribution can be used to model the fading. The channel fading amplitude H (the signal attenuation due to reflexions) follows the Rayleigh distribution of parameter Ω if its p.d.f is

$$f(h) = \frac{2h}{\Omega} \exp\left(-\frac{h^2}{\Omega}\right), \forall h \in \mathbb{R}_+. \quad (2.4)$$

When a LoS path exists, the Nakagami- m distribution can be used instead with the m parameter adjusted to the relative strength of the LoS component with regard to the multipath components. The channel fading amplitude H follows the Nakagami- m distribution of parameters m and Ω if its p.d.f is

$$f(h) = \frac{2m^m h^{(2m-1)}}{\Omega^m \Gamma(m)} \exp\left(-\frac{mh^2}{\Omega}\right), \forall h \in \mathbb{R}_+. \quad (2.5)$$

where $\Gamma(\cdot)$ is the gamma function. If $m = 1$, the Nakagami- m distribution reduces to Rayleigh fading, and as m grows to infinity, the Nakagami- m distribution gets closer to a pure LoS fading which is equal to 1.

If the electromagnetic signal experiences a mask effect from solid objects, a log-normal distribution is used to model the additional attenuation it experiences. Since these attenuations are more persistent in time, they are often called slow-fading. The p.d.f of the Log-normal shadowing is given by

$$f(\gamma) = \frac{\xi}{\sqrt{2\pi}\sigma\gamma} \exp\left(-\frac{(10\log_{10}(\gamma) - \mu)^2}{2\sigma^2}\right), \forall \gamma \in \mathbb{R}_+. \quad (2.6)$$

where μ and σ are the mean and variance of $10\log_{10}(\gamma)$, and $\xi = \frac{10}{\log(10)}$.

2.2.4 Martingales

Martingales are commonly used to characterize noise in stochastic approximation algorithms [15, 48]. We hereby give a succinct definition of martingales and martingale differences along with an insight in why they are useful.

Let $(\Omega, \mathcal{F}, \mathcal{P})$ denote a probability space, where Ω is the sample space, \mathcal{F} a σ -algebra of subsets of Ω , and P a probability measure on (Ω, \mathcal{F}) . Let $\{M_n\}$ be a sequence of real-valued random variables defined on (Ω, \mathcal{F}) . If

$$\mathbb{E}(M_{n+1} | M_i, i \leq n) = M_n \quad (2.7)$$

then $\{M_n\}$ is a martingale sequence. In this case, the sequence $N_n = M_n - M_{n-1}$ is a martingale difference sequence.

An important result on martingales is the martingale convergence theorem which proves that under certain conditions, martingale sequences converge. This result is useful to characterize convergence of SA algorithms which model the noise as martingale differences (see [15] and [48]).

2.3 Queuing theory

The results in this section are summarized from [12] and [11] to which the reader is referred to for more details. We briefly recall them here in order to make the document self-contained. We recall basic results in queuing theory such as Markov chains, Markov processes, M/G/1 queues and also discuss the queue model for a wireless network. The reader is referred to [44] for detailed discussion on queuing theory.

2.3.1 Markov chains and Markov processes

A stochastic process is a collection of random variables describing the evolution of a system over time. A simple example of a stochastic process is the number of customers in a queue of say a postal service. The randomness in this case comes from the arrival instants of customers in the post office and also the duration of their service. Stochastic processes can be classified according to whether the time and the state space are discrete or continuous.

Let us denote by $X_t, t \in \mathcal{T}$ a collection of random variables (also known as states) taking values in a set \mathcal{X} . We also denote by $(X_t)_{t \in \mathcal{T}}$ the stochastic process describing this collection of random variables. $(X_t)_{t \in \mathcal{T}}$ is said to be in discrete-time if \mathcal{T} is countable, otherwise it is continuous-time. $(X_t)_{t \in \mathcal{T}}$ is said to be discrete-space if \mathcal{X} is countable, otherwise it is continuous-space.

In the following, we focus on discrete-space stochastic processes. Stochastic processes can also be classified according to the dependency between its states. We have either completely independent states, Martingale states, Markov states, deterministic states.

In this section, we focus on Markov processes. The Markov property named after the Russian mathematician Andrey Markov (1856-1922) refers to the memoryless property of a stochastic process.

The collection of random variables $X_n, n \geq 0$ is a Markov chain if $\forall n \geq 0$ and $\forall x, y \in \mathcal{X}$, $x_0, x_1, \dots, x_{n-1} \in \mathcal{X}$, we have

$$\mathbb{P}(X_{n+1} = y | X_n = x; X_0 = x_0, \dots, X_{n-1} = x_{n-1}) = \mathbb{P}(X_{n+1} = y | X_n = x). \quad (2.8)$$

In other words, the $(n+1)$ th state depends only on the n th state. If the equality (2.8) does not depend on n then the Markov chain is homogeneous. In this case, the Markov chain is completely

determined by its transition probabilities

$$\forall x, y \in \mathcal{X}, p(x, y) = \mathbb{P}(X_1 = y | X_0 = x). \quad (2.9)$$

A probability measure π over \mathcal{X} is a stationary distribution of the Markov chain if $\mathbb{P}(X_n = x) = \pi(x)$, $\forall x \in \mathcal{X}$ whenever $\mathbb{P}(X_0 = x) = \pi(x)$, $\forall x \in \mathcal{X}$. In this case, the stationary distribution satisfies the balance equations

$$\forall y \in \mathcal{X}, \pi(y) = \sum_{x \in \mathcal{X}} \pi(x) p(x, y). \quad (2.10)$$

This stationary distribution does not always exist, and when it does, it is unique and the Markov chain is said to be stable. For a stable Markov chain with stationary distribution π , the mean time that the system takes to return to a state x starting at x is $\frac{1}{\pi(x)}$.

A particular example of Markov chain is the discrete time birth-death process on $\mathcal{X} = \mathbb{N}$. The transition probabilities in this case are null between non consecutive integers. We denote by

$$\begin{cases} a(x) = p(x, x+1) \\ b(x) = p(x, x-1) \end{cases} \quad \forall x \neq 0 \quad (2.11)$$

the transition probabilities from a state x to one of its neighbours such that $a(x) + b(x) = 1$. It is noted that from state 0, the system transitions to state 1 with probability 1. The stationary distribution of this Markov chain is given by

$$\pi(x) = \pi(0) \frac{a(1)a(2)\dots a(x-1)}{b(1)b(2)\dots b(x)}, \quad \forall x \neq 0. \quad (2.12)$$

Assuming that $\lim_{x \rightarrow \infty} \frac{a(x)}{b(x)} = \bar{\rho}$ exist, then the birth-death process is stable if and only if $\bar{\rho} < 1$.

The collection of random variables $X(t)$, $t \in \mathbb{R}_+$ is a Markov process if $\forall t, s \in \mathbb{R}_+$, $\forall x, y \in \mathcal{X}$ and $\forall t_1, \dots, t_l < t$ and $x_1, \dots, x_l \in \mathcal{X}$, we have

$$\mathbb{P}(X(t+s) = y | X(t) = x; X(t_1) = x_1, \dots, X(t_l) = x_l) = \mathbb{P}(X(t+s) = y | X(t) = x). \quad (2.13)$$

In other words, $X(t)$ is memoryless.

For time-invariant Markov processes, we define transition rates as follows

$$q(x, y) = \lim_{t \rightarrow 0} \frac{1}{t} \mathbb{P}(X(t) = y | X(0) = x) \quad (2.14)$$

for all $x, y \in \mathcal{X}$. $q(x, y)$ is the transition rate from state x to state y . In addition, the departure rate from state x is defined as

$$q(x) = \sum_{y \neq x} q(x, y). \quad (2.15)$$

As for the Markov chains, a probability measure π over \mathcal{X} is a stationary distribution of the process $X(t)$ if $\forall t \in \mathbb{R}_+$, we have $\mathbb{P}(X(t) = x) = \pi(x)$, $\forall x \in \mathcal{X}$ whenever π is the distribution of $X(0)$. In this case, π satisfies the balance equations for Markov processes:

$$\forall y \in \mathcal{X}, \pi(y) \sum_{x \neq y} q(y, x) = \sum_{x \neq y} \pi(x) q(x, y). \quad (2.16)$$

If the Markov process admits a stationary distribution, it is said to be stable and the mean return time to a state is the inverse of the frequency of passage through state x which is $\pi(x)q(x)$.

The equivalent example to random walk for Markov processes is the birth-death process for which $\mathcal{X} = \mathbb{N}$. The transitions probabilities $a(x)$ and $b(x)$ are replaced by the transition rates $\lambda(x) = q(x, x+1)$ and $\mu(x) = q(x, x-1)$, $\forall x \neq 0$ respectively. The birth-death process is stable if and only if $\lim_{x \rightarrow \infty} \lambda(x) = \bar{\lambda}$ and $\lim_{x \rightarrow \infty} \mu(x) = \bar{\mu}$ exist and $\bar{\lambda} < \bar{\mu}$. In this case its stationary distribution is given by

$$\pi(x) = \pi(0) \frac{\lambda(0)\lambda(1)\dots\lambda(x-1)}{\mu(1)\mu(2)\dots\mu(x)}. \quad (2.17)$$

2.3.2 Performance evaluation for queues of interest

A real-life queue can be described by a Markov process whose parameters depend on the characteristics of the queue such as the arrival process, the number of servers, the capacity of the queue, the service times and the service discipline (First In First Out (FIFO), Last In First Out (LIFO) or Processor Sharing (PS)).

In this chapter, we focus on PS discipline and consider Poisson arrivals. We present some performance metrics for M/G/1/n PS queues as well as their application to wireless communications.

In a M/G/1/n PS queue, the inter-arrival times follow an exponential distribution of parameter λ , the service times follow a general distribution, there is only one server and the queue can take in n users at most. Also the users are served with the PS discipline meaning that the resources are shared among all users present in the system (there is no waiting for another user to finish being service before beginning service for an arriving user).

If we assume that all users that arrive in the queue have the same service rate μ , then the resources are always equally shared among the users present in the system. The queue is then stable if and only if the load ρ satisfies

$$\rho = \frac{\lambda}{\mu} < 1. \quad (2.18)$$

In this case, the queue admits a stationary distribution given by

$$\pi(x) = (1 - \rho)\rho^x \quad (2.19)$$

The service rate can represent the inverse of the mean duration of a client service at a post office or the file transfer time of a mobile device. In case the stability of the system is not ensured,

one can enforce an admission control in the form of a maximum number of users in the system above which all new users are rejected. If we denote by m this number, the stationary distribution becomes

$$\pi(x) = \frac{\rho^x}{1 + \rho + \dots + \rho^m}, \quad \forall x = 0 \dots m. \quad (2.20)$$

The blocking probability can also be derived as

$$B = \pi(m) = \frac{\rho^m}{1 + \rho + \dots + \rho^m}. \quad (2.21)$$

One can also deduce the blocking threshold m when the blocking probability is given by solving the above equation with known load ρ . The solution is as follows

$$m = \text{floor} \left(\frac{\log(B) - \log(1 - \rho + B\rho)}{\log \rho} \right). \quad (2.22)$$

The results above can be generalized to a case where we have N classes of users in which class i users arrive at rate λ_i and are served at rate μ_i . In this case, the load is redefined as

$$\rho = \sum_i \rho_i = \sum_i \frac{\lambda_i}{\mu_i}. \quad (2.23)$$

The condition for stability is the same ($\rho < 1$) and the stationary distribution is given by

$$\pi(x) = (1 - \rho) \frac{(\sum_{i=1}^N x_i)!}{\prod_{i=1}^N x_i!} \prod_{i=1}^N \rho_i^{x_i}. \quad (2.24)$$

In any case, the mean number of users in the system is given by

$$\mathbb{E}(x) = \frac{\rho}{1 - \rho} \times \frac{1 - (m+1)\rho^m + m\rho^{m+1}}{1 - \rho^{m+1}} \quad (2.25)$$

so that when $m \rightarrow \infty$ (no admission control), it is reduced to

$$\mathbb{E}(x) = \frac{\rho}{1 - \rho} \quad (2.26)$$

Knowing the mean number of users in the system, the Little's law

$$\mathbb{E}(x) = \lambda T \quad (2.27)$$

can be used to derive the mean service time T knowing the total arrival rate λ .

2.3.3 Wireless downlink performance

In a wireless cell, the arrival process can be described by a Poisson process distributed spatially. As such we denote by $\lambda(r)$ the arrival rate at position r . The user at position r requests to download

a file of size σ and depending on his position, he can be served with a maximum data rate of $R(r)$. We only consider long term average data rates so the effect of fast fading is averaged out.

The load of such a system can be defined in a similar way as in the previous section, namely

$$\rho = \int_A \frac{\lambda(r)\mathbb{E}(\sigma)}{R(r)} dr \quad (2.28)$$

where A is the area of the cell. The only difference now is that the service times $(\mathbb{E}(\sigma)/R(r))$ take continuous values so the sum is replaced by an integral. In real systems though, the data rates can only take discrete values so the sum can be used.

Since in this case PS is also used, the insensitivity property implies that the number of users in the system does not depend on the service time distribution, thus we also have

$$\pi(x) = (1 - \rho)\rho^x \quad (2.29)$$

and the mean number of users is also given by (2.26). The mean throughput that a user arriving at a given position r would get is then written as

$$\gamma(r) = (1 - \rho)R(r) \quad (2.30)$$

It is noted that PS is implemented here with the Round-Robin scheme.

If admission control is enforced with a maximum number of users of m , the stationary distribution is given by (2.20) and the blocking probability by (2.21). The mean number of active users is also given by (2.25) and the throughput at position r becomes

$$\gamma(r) = \frac{(1 - \rho)(1 - \rho^m)}{1 - (m + 1)\rho^m + m\rho^{m+1}} R(r). \quad (2.31)$$

2.4 Convex optimization

2.4.1 Notion of convexity

A subset \mathcal{C} of a vector space is convex if for any $x_1, x_2 \in \mathcal{C}$ and any $0 \leq \tau \leq 1$, we have

$$\tau x_1 + (1 - \tau)x_2 \in \mathcal{C}. \quad (2.32)$$

In other words, a line between any two points of the set lies entirely in the set. The convex hull of a set denoted $\mathbf{conv} \mathcal{C}$ is the set of all convex combinations of points in \mathcal{C} :

$$\mathbf{conv} \mathcal{C} = \left\{ \sum_{i=1}^k \tau_i x_i \mid x_i \in \mathcal{C}, \tau_i \geq 0, i = 1 \dots k, \sum_{i=1}^k \tau_i = 1 \right\}. \quad (2.33)$$

The convex hull of a set \mathcal{C} is thus the smallest convex set containing all the elements of \mathcal{C} .

The intersection of convex sets is a convex set. The image of a convex set under an affine function is a convex set.

A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if its domain ($\mathbf{dom}f$) is convex and if for all $x, y \in \mathbf{dom}f$, and τ with $0 \leq \tau \leq 1$, we have

$$f(\tau x + (1 - \tau)y) \leq \tau f(x) + (1 - \tau)f(y). \quad (2.34)$$

Geometrically, convexity means that the line segment between the points $(x, f(x))$ and $(y, f(y))$ lies above the graph of f .

The function f is concave if $-f$ is convex.

A differentiable function f is convex if and only if $\mathbf{dom}f$ is convex and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall x, y \in \mathbf{dom}f. \quad (2.35)$$

A twice-differentiable function f is convex if and only if $\mathbf{dom}f$ is convex and its Hessian $H(x)$ is positive semi-definite for all $x \in \mathbf{dom}f$:

$$H(x) = \nabla^2 f(x) \succeq 0. \quad (2.36)$$

Examples of convex functions

- $f(x) = a \log(bx + c)$ where a, b , and c are positive constants and $(bx + c) \in \mathbb{R}_+$.
- $f(x) = -x \log(1 + \frac{a}{bx+c})$ where a, b , and c are positive constants and $x \in [\tau, 1 - \tau]$ and τ a small positive constant.

2.4.2 Convex optimization problem

An optimization problem is formulated as follows

$$\begin{aligned} & \text{minimize}_{x \in \mathcal{X}} && f(x) \\ & \text{subject to} && g_i(x) \leq 0, i = 1, \dots, m \\ & && h_i(x) = 0, i = 1, \dots, p \end{aligned} \quad (2.37)$$

where x is the optimization variable, $f(x)$ is the objective function, $g_i, i = 1, \dots, m$ and $h_i, i = 1, \dots, p$ are real functions of x . The last two lines of Equation (2.37) are respectively the inequality and equality constraints.

The problem (2.37) is convex if $f, g_i, i = 1, \dots, m$ are convex, and $h_i, i = 1, \dots, p$ are affine. In this case, the problem is called a convex program. If f is quadratic and $g_i, i = 1, \dots, m, h_i, i = 1, \dots, p$ are affine, it is a quadratic program. Finally if $f, g_i, i = 1, \dots, m, h_i, i = 1, \dots, p$ are all affine it is a linear program.

The Lagrangian is defined as

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x) \quad (2.38)$$

where $\lambda = [\lambda_1, \dots, \lambda_m]^T$ and $\nu = [\nu_1, \dots, \nu_p]^T$ are the Lagrange multipliers associated respectively with the inequality and equality constraints. The Lagrange dual function is subsequently defined as

$$l(\lambda, \nu) = \inf_{x \in \mathcal{X}} L(x, \lambda, \nu). \quad (2.39)$$

The Lagrange dual problem is then defined as

$$\begin{aligned} & \text{maximize}_{\lambda, \nu} \quad l(\lambda, \nu) \\ & \text{subject to} \quad \lambda_i \geq 0, i = 1, \dots, m \end{aligned} \quad (2.40)$$

2.4.3 KKT optimality conditions

Theorem 1. Suppose that \mathcal{X} is a convex set, f and g are convex functions and h is an affine function. Moreover suppose that the constraints are strictly feasible (Slater's condition), i.e. $\exists x \in \mathcal{X}$ such that $g_i(x) < 0, i = 1, \dots, m$ and $h_i(x) = 0, i = 1, \dots, p$. Let x^* be an optimal point of the primal problem (2.37) and (λ^*, ν^*) be an optimal point for the dual problem (2.40), then

$$\begin{aligned} g_i(x^*) &\leq 0, i = 1, \dots, m \\ h_i(x^*) &= 0, i = 1, \dots, p \\ \lambda_i^* &\geq 0, i = 1, \dots, m \\ \lambda_i^* g_i(x^*) &\geq 0, i = 1, \dots, m \\ \nabla L(x^*, \lambda^*, \nu^*) &= 0, \end{aligned} \quad (2.41)$$

which are called the Karush-Kuhn-Tucker (KKT) conditions.

It is recalled that the KKT conditions are necessary but not sufficient in general. Sufficient conditions are the second-order conditions (see Equation 2.36). They can nonetheless be used to derive the optimal solution for most convex optimization problems.

2.4.4 Algorithms

We describe here only two algorithms for solving unconstrained convex programs: the gradient descent and the conjugate gradient descent. These algorithms are based on the KKT conditions for optimality where they start at any feasible point and advance towards the optimal point by taking steps in the direction of the gradient of the Lagrangian.

The two algorithms below solve the following problem

$$\text{minimize}_{x \in \mathcal{X}} f(x) \quad (2.42)$$

where f is convex and twice continuously differentiable. In this case, a necessary and sufficient condition for some $x^* \in \text{int}\mathcal{C}$ to be optimal is that

$$\nabla f(x^*) = 0 \quad (2.43)$$

(see [17, §4.2.3] for a detailed proof). So finding the optimal solution to (2.42) is equivalent to solving (2.43) which can be done analytically if f allows it or iteratively otherwise. The two Algorithms 1 and 2 are iterative methods for solving (2.43).

Algorithm 1 Gradient Descent (with backtracking line search)

```

1: Initialization:
2: Choose  $\eta > 0$ ,  $\alpha \in (0, 0.5)$  and  $\beta \in (0, 1)$ 
3:  $x \leftarrow x_0 \in \mathcal{X}$ 
4: loop:
5: while  $k < k_{\max}$  and  $\|\nabla f(x)\|_2 \geq \eta$  do
6:    $\Delta x \leftarrow -\nabla f(x)$ 
7:    $\epsilon_k = 1$ ,
8:   while  $f(x + \epsilon_k \Delta x) > f(x) + \alpha \epsilon_k \nabla f(x)^T \Delta x$  do
9:      $\epsilon_k \leftarrow \beta \epsilon_k$ 
10:   $x \leftarrow x + \epsilon_k \Delta x$ 
11:   $k \leftarrow k + 1$ 

```

Algorithm 2 Newton's method (Conjugate gradient descent)

```

1: Initialization:
2: Choose  $\eta > 0$ ,  $\alpha \in (0, 0.5)$  and  $\beta \in (0, 1)$ 
3:  $x \leftarrow x_0 \in \mathcal{X}$ 
4: loop:
5: while  $k < k_{\max}$  and  $\frac{1}{2} \nabla f(x)^T (\nabla^2 f(x))^{-1} \nabla f(x) \geq \eta$  do
6:    $\Delta x \leftarrow -(\nabla^2 f(x))^{-1} \nabla f(x)$ 
7:    $\epsilon_k = 1$ ,
8:   while  $f(x + \epsilon_k \Delta x) > f(x) + \alpha \epsilon_k \nabla f(x)^T \Delta x$  do
9:      $\epsilon_k \leftarrow \beta \epsilon_k$ 
10:   $x \leftarrow x + \epsilon_k \Delta x$ 
11:   $k \leftarrow k + 1$ 

```

The reader is referred to [17] for more details on these algorithms and other algorithms. The reader can also refer to [55] for extended discussion on interior-point algorithms which are used to solve more general convex optimization problems.

2.5 Stochastic Approximation

In real-life optimization problems, the objective function is seldom analytically defined. Instead the value of the function at a given point is obtained via measurements and these measurements

are in general noisy. For example in wireless networks, when the objective function to be optimized depends on the load of the cell, such a performance indicator can only be obtained from observing the cell during a certain time interval and estimating the load from these observations.

The problem of optimizing a function for which only costly and noisy measurements are available has been largely addressed in the literature. A class of algorithms for this problem is known as Stochastic Approximation (SA). A SA algorithm is an iterative algorithm of the following form

$$x(k+1) = x(k) - \epsilon_k(f(x(k)) + m_k) \quad (2.44)$$

where k is the iteration counter, x - the optimization variable, ϵ_k - a step size, f - the descent direction towards the optimal point and m_k - the noise resulting from estimating $f(x(k))$.

2.5.1 Robbins-Monro algorithm

The first recursive stochastic algorithms were developed by Herbert Robbins and Sutton Monro for finding the root of a real-valued function g . In this case, only noisy observations of $g(x)$ are available for a given parameter x . The algorithm proposed in [61] is the following

$$x(k+1) = x(k) - \epsilon_k Y_k \quad (2.45)$$

where ϵ_k satisfies

$$\epsilon_k > 0, \epsilon_k \rightarrow 0, \sum_k \epsilon_k = \infty, \sum_k \epsilon_k^2 < \infty \quad (2.46)$$

and Y_k is the estimate of g at $x(k)$. In this case, the authors in [61] have shown that if $Y_k - g(x(k))$ is a Martingale difference sequence and g is strictly increasing then $x(k)$ converges almost surely to a set in which $g(x) = 0$.

2.5.2 Kiefer-Wolfowitz algorithm

Kiefer and Wolfowitz [42] later proposed the stochastic version of the gradient descent algorithm where they assume that the gradient can only be estimated through finite differences. The algorithm proposed is the same as (2.45) but Y_k is defined as

$$Y_k = \frac{g(x(k) + c_k) - g(x(k) - c_k)}{2c_k} \quad (2.47)$$

where g is the convex function whose minimum is sought and $c_k \rightarrow 0$ a finite difference interval. As in the Robbins-Monro algorithm, if condition (2.46) is verified and the sequence $\frac{\epsilon_k}{c_k}$ is square summable, then the Kiefer-Wolfowitz algorithm converges almost surely towards x^* such that $g(x) \geq g(x^*)$ for all $x \in \mathcal{X}$.

2.5.3 A sketch of the convergence proof

The convergence proof of (2.44) can be found in [48] and [15]. It mainly relies on showing that the algorithm (2.44) has the same asymptotic behavior as the solutions to the Ordinary Differential Equation (ODE)

$$\dot{x} = -f(x) \quad (2.48)$$

and this ODE is asymptotically stable. The convergence proofs use the following assumptions:

- the function f is Lipschitz,
- the sequence of step sizes ϵ_k is non-summable but square-summable

$$\sum_k \epsilon_k = \infty, \quad \sum_k \epsilon_k^2 < \infty, \quad (2.49)$$

- $\{m_k\}$ is a Martingale difference sequence with respect to the increasing family of σ -fields

$$\mathcal{F}_k = \sigma(x(0), m_1, \dots, m_k), \quad k \geq 0, \quad (2.50)$$

- The iterates of (2.45) remain bounded almost surely (a.s), i.e.,

$$\sup_k \|x(k)\| < \infty, \quad \text{a.s.} \quad (2.51)$$

An intuitive explanation for the equivalence between (2.48) and (2.44) is the Euler scheme. If the step sizes are taken small enough, we can assume that the effect of m_k will be averaged out of (2.44) which is rewritten as

$$x(k+1) = x(k) - \epsilon_k f(x(k)) \quad (2.52)$$

which is equivalent to

$$\frac{x(k+1) - x(k)}{\epsilon_k} = -f(x(k)). \quad (2.53)$$

Now since k represents the time dimension, for ϵ_k small enough, the Left Hand Side (LHS) of Equation (2.53) is approximately equal to \dot{x} which gives us Equation (2.48).

2.5.4 Constant step size selection

Instead of decreasing step sizes, a small constant step size can be used. The convergence is weaker (weak convergence [15, 48]) since the algorithm will fluctuate around the optimal solution. However, a constant step size allows to track a non-stationary optimization environment. It also provides a simpler algorithm to implement. The constant step size must be chosen so as to minimize the variance of the updates but also keep a good convergence rate.

2.5.5 Projected SA

When the parameters $x[k]$ are restricted to a compact convex set S , the SA algorithm (2.44) becomes a Projected SA algorithm of the form

$$x(k+1) = [x(k) - \epsilon_k(f(x(k)) + m_k)]_S^+ \quad (2.54)$$

where $[\cdot]_S^+$ is the projection on the set S . In this case, if the projection verifies some assumptions discussed in [15, Section 5.4], the convergence proof remains the same as in the no projection case. Otherwise, differential inclusion tools (see [15, Section 5]) can be used to assess the stability of the system.

One advantage of projections is that they ensure that the iterates of the SA algorithm remain bounded which is essential for its stability.

2.6 Diagonal strict concavity

Diagonal strict concavity is a property introduced in [63] for analyzing equilibrium of n-person games. Consider L functions $\theta \rightarrow g_l(\theta)$ defined on a convex closed bounded set $S \subset \mathbb{R}^L$, and $w_l, l = 1, \dots, L$ some real positive constants. And define

$$JG(\theta) = \begin{bmatrix} w_1 \nabla_1 g_1(\theta) \\ \vdots \\ w_L \nabla_L g_L(\theta) \end{bmatrix}. \quad (2.55)$$

We say that $G(\theta) = \sum_{l=1}^L w_l g_l(\theta)$ is diagonally strictly concave for $\theta \in S$ if for every $\theta_0, \theta_1 \in S$ we have

$$(\theta_0 - \theta_1)^T JG(\theta_0) + (\theta_0 - \theta_1)^T JG(\theta_1) > 0 \quad (2.56)$$

Chapter 3

Self-optimizing strategies for heterogeneous networks

“Evolution is definable as a change from an incoherent homogeneity to a coherent heterogeneity, accompanying the dissipation of motion and integration of matter.”

– Herbert Spencer

3.1 Introduction

The ever-growing traffic demand has called for the densification of mobile networks in order to increase their capacity. Small cells provide an attractive solution for this densification because of their low cost and their ease of deployment.

With their low transmission power, the small cells bring heterogeneity in the mobile network in terms of cell coverage and interference distribution. This heterogeneity can be the cause of new specific challenges that need to be addressed.

Small cells have a transmission power of around 30dBm while the macro BSs transmit at around 46dBm, which represents a 40 times decrease in the transmit power for the small cells. As a result, the small cells coverage is much smaller and their offloading capability is limited. Also, the coverage area of the small cells should be adapted to the traffic distribution in their vicinity. Small cells users are also vulnerable to macro cells interference.

In order to remedy these problems, new offloading and interference management mechanisms have to be devised. Different types of solutions are presented in [51]. In 3GPP, Cell Range Extension (CRE) has been introduced as a means to increase the coverage of the low power nodes by adjusting their coverage parameters. CRE enables load balancing between macro cells and small cells, but in return more handover could occur in the network and some users in the CRE area may experience radio link failures.

In order to mitigate interference, 3GPP introduced the ABS mechanism [4, Section 16.1.5] as an eICIC solution. This is a time domain interference mitigation scheme between the macro BSs layer and the small cells layer. In a nutshell, the macro BSs stop transmitting any data signals during some subframes (time intervals of length 1ms), and since certain control signals are still transmitted there is a residual interference thus the prefix 'Almost' in ABS. The use of the ABS mechanism reduces the macro BSs capacity, so care must be taken on the choice of the ABSr (proportion of muted subframes) in order to preserve the global network performance.

The performance evaluation of these two mechanisms (CRE and ABS) has received much attention lately. The achievable performance gains for static settings of the coverage and ABS parameters have been studied in [83], [58] and [66]. Optimization algorithms are provided in [57] and [8] for static network and traffic configurations. In [79], the ABS mechanism is optimized for a dynamic environment with fixed coverage. A joint optimization of CRE and ABS in dynamic environment is investigated in [28] and [41] with a centralized architecture.

As an alternative to time-domain interference mitigation as proposed by 3GPP, a frequency domain interference mitigation can be used. An extensive study of the achievable performance for both time and frequency domain interference coordination along with different user associations schemes is investigated in [31]. The authors propose optimization algorithms for both user association and interference mitigation parameters for a fixed deployment scenario (fixed traffic and BSs configurations) and full buffer traffic type. In particular they compare the benefits of a co-channel deployment with ABS mechanism and an orthogonal deployment where macro and small cells layers operate on disjoint bands.

In this chapter we propose self-optimizing algorithms for load balancing and interference mitigation both in time domain and frequency domain for a HetNet scenario with dynamic traffic. In Section 3.2, we first discuss the existing literature on load balancing and present the proposed algorithms for HetNets in this thesis taking also backhaul limitations into account. We next describe the algorithms we designed for ABS ratio optimization (ABSrO) (Section 3.3) and frequency split optimization (Section 3.4). Section 3.5 presents the performance results obtained by applying combinations of load balancing and interference coordination algorithms in a HetNet simulated at the flow level. Finally, Section 3.6 briefly describes the integration of some of the proposed algorithms into a SDN framework that provides mobile network programmability.

Throughout this chapter, we assume that the data rate of a user is given by a modified Shannon formula [3] as follows

$$R(\text{SINR}_u) = \eta W \min[4.4, 0.6 \log_2(1 + \text{SINR}_u)] \quad (3.1)$$

where SINR_u is user u Signal to Interference plus Noise Ratio (SINR), η - the proportion of time-frequency resources allocated to him by the scheduler, and W - the total bandwidth available. If Round-Robin scheduling is used for example for elastic traffic, we would have $\eta = \frac{1}{N_u}$ where N is the number of users served by the serving cell of user u . The SINR of a user will depend on the interfering base stations and on the power/bandwidth used.

The following assumption is also used throughout the chapter.

Assumption 1. ϵ_k are some positive step sizes that are non-summable ($\sum_{k=0}^{\infty} \epsilon_k = \infty$) but square-summable ($\sum_{k=0}^{\infty} \epsilon_k^2 < \infty$).

3.2 Load balancing

3.2.1 Literature overview

The problem of balancing loads is present in many areas of computer science and communication systems. A cluster of web servers delivering web pages to users needs a load balancing algorithm in order to distribute the traffic among the servers. The same goes for a network of WiFi access points or a network of mobile BSs. In all those cases, load balancing allows to increase the network capacity (number of users that can be served simultaneously) and reduce response times (delay before start of service).

The load balancing problem has been largely addressed in the literature and the proposed algorithms are as diverse as the specific setting for which they are designed. In [21, 27, 84], the authors propose load balancers for computer systems (web servers or parallel processors). For Wireless Local Area Networks (WLANs), load balancing schemes have been proposed in [9, 10, 20, 32, 46]. A more generic approach is studied in [52] where the authors study the performance of the following load balancing model: each customer waits for service in the shortest

queue among d queues that are pre-selected independently and uniformly at random among the n available servers. This model can be applied to application servers or supermarket cashiers.

Load balancing is also one of the main use cases identified in 3GPP for SON. As such, a vast literature also exists for load balancing in mobile networks using either distributed schemes [23, 43, 54, 62] or centralized ones [40, 70, 82]. We focus on distributed load balancing algorithms and present here two instances from the literature that we later use for the simulation results.

The first approach introduced in [43] allows to optimize a broad range of KPIs. The load balancing strategy here is to use a User Association (UA) scheme (denoted as α -Fair User Association (AFUA)) in which the attachment decision is taken by the mobile. Each user u that arrives in the network chooses the best serving BS according to the following rule

$$s_u^* = \operatorname{argmax}_s R_{u,s} (1 - \rho_s)^\alpha \quad (3.2)$$

where $R_{u,s}$ is the maximum data rate achievable by user u if served by BS s , ρ_s is the load of BS s , and α is a parameter impacting the performance objective. Note that this is an iterative process where the loads of the cells are updated after each user association until convergence.

The authors in [43] show that this scheme minimizes the quantity $\sum_s \frac{(1-\rho_s)^{1-\alpha}}{\alpha-1}$. For example if $\alpha = 0$, applying (3.2) is the same as the classic UA scheme, i.e. the user attaches to the BS which provides him with the best peak rate. If $\alpha = 2$, the scheme optimizes the mean delay in the network.

In order to have realistic user association, we implement a constrained version of the AFUA in the performance evaluation. BSs can only serve users that they can cover with a minimum received signal strength and/or a minimum SINR, so their range can be extended up to a certain maximum value. This is to prevent the users to attach to distant BSs because the control signals would otherwise be very weak and strongly interfered.

The second approach is introduced in [23] and consists in adapting the pilot powers of a cell according to the difference between its load and the load of its most loaded neighbour. The update equation at iteration $k \in \mathbb{N} \setminus \{0\}$ is as follows

$$P_s(k+1) = P_s(k)(1 + \epsilon_k(\hat{\rho}_1(k) - \hat{\rho}_s(k))) \quad (3.3)$$

where P_s is the pilot power of BS s , $\hat{\rho}_s$ - its estimated average load, $\hat{\rho}_1$ - the estimated average load of the reference cell (most loaded neighbouring cell) and ϵ_k - a step size. The authors in [23] have shown that provided that ϵ_k satisfies (2.46), the algorithm (3.3) succeeds in balancing the loads and the proof is based on SA results (see Section 2.5).

We present below (Section 3.2.2) an adaptation of this algorithm to heterogeneous networks where we replace the pilot powers by CIOs and redefine the reference cell to be the nearest macro cell. Moreover the load balancing algorithm is implemented only on small cells. We also modify the load definition later in Section 3.2.3 in order to take into consideration possible backhaul limitations.

3.2.2 Infinite-capacity backhaul heterogeneous networks scenario

UE attachment to a BS is determined by comparing the received pilot powers from all surrounding BSs plus a certain offset which is denoted as CIO. The attachment rule for UE u can be formulated as follows

$$s^* = \operatorname{argmax}_s \text{CIO}_s h_s^u P_s \quad (3.4)$$

where s^* is the chosen serving cell, CIO_s - the CIO of cell s , P_s - its pilot power and h_s^u - the pathloss from BS s to UE u .

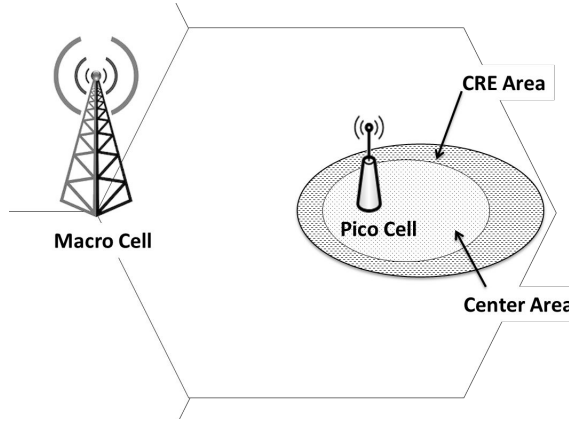


Figure 3.1: Illustration of Cell Range Extension in a HetNet

CRE can be performed by increasing the CIO of the small cell. By setting a nonnegative CIO_{dB} (CIO in dB) for the small cell, we force the macro users to attach to a small cell which is not their best serving cell as shown in Figure 3.1. We say that these users are in the CRE area. Hence the CRE allows to increase the offloading of the macro cell. However, the CRE users now experience more interference from the macro BS which is their best serving cell. The macro cell interference reduces the SINR at the CRE area thus limiting the extent of small cell offloading. The ABS mechanism (presented in Section 3.3.1) has been proposed in 3GPP to reduce the interference experienced by CRE users.

LB SON adjusts the small cells CIOs in order to balance the loads between a macro BS and the small cells. The idea is to increase the small cell area whenever its load is lower than that of the macro BS in the coverage area of which it is located (and vice versa).

Consider the downlink of a mobile network (e.g. LTE). We suppose that the backhaul has an infinite capacity or at least greater than the capacity of the BS. Two equivalent definitions can be used for the load of the BS. The first is the occupation rate of its resources. The second is the ratio between the traffic demand and the cell capacity which is valid for both elastic or Guaranteed Bit-Rate (GBR) traffic.

In the case of a BS serving elastic traffic, users arrive randomly according to a Poisson process of intensity $\lambda(r)$ at position r , download a file of random size σ with mean $\mathbb{E}(\sigma)$ and leave the

network when their download is complete. The load is written as [12]

$$\rho_e = \min \left(1, \int_A \frac{\lambda(r)\mathbb{E}(\sigma)}{R(r)} dr \right), \quad (3.5)$$

where A is the area of the considered cell, and $R(r)$ (in Mbps) is the peak data rate (i.e. when the user is alone in the cell) at position r averaged over fading.

If the BS also serves GBR traffic with fixed amount of resources allocated to the users, priority is given to the GBR users. We consider that GBR traffic varies slowly with respect to elastic traffic (which is bursty) and is considered constant in (3.6). The load is redefined as

$$\rho = \min \left(1, \rho_{GBR} + \int_A \frac{\lambda(r)\mathbb{E}(\sigma)}{\bar{R}(r)} dr \right), \quad (3.6)$$

where $\bar{R}(r)$ is the peak data rate achievable at position r when using only the resources left after scheduling all GBR users, and ρ_{GBR} is the proportion of resources occupied by the GBR traffic at the radio access (in this case $\bar{R}(r) = (1 - \rho_{GBR})R(r)$). Consider for example a case where the total available bandwidth is 10MHz, and with two GBR users each requesting 7.5Mbps of data rate. We suppose that the spectral efficiency for both users is 3.75 bit/Hz/s, then 4MHz of bandwidth (or the equivalent amount of resource blocks) is required for the GBR users, and $\bar{R}(r)$ is the peak data rate achievable with the remaining 6MHz. If the available resources cannot satisfy all GBR users then the load is 1. It is noted that this definition of the load does not depend on the scheduling algorithm which only impacts the user performance.

In practice, the load is estimated by the proportion of time-frequency resources that are occupied by the scheduler over a certain time period. We denote by K the total number of resource blocks available at a LTE BS, by K_t the number of resource blocks used at time slot t , and by T the total number of time slots over which the load is estimated. The load estimator is then given as

$$\hat{\rho} = \frac{1}{K \cdot T} \sum_{t=1}^T K_t. \quad (3.7)$$

If we consider traffic offloading from a macro BS m to a small cell s , the SA update equation defining the SON algorithm reads

$$CIO_{dBs}(k+1) = CIO_{dBs}(k) + \epsilon_k(\hat{\rho}_m(k) - \hat{\rho}_s(k)) \quad (3.8)$$

where $\hat{\rho}_m$ and $\hat{\rho}_s$ are the estimators of ρ_m and ρ_s obtained by using (3.7). ϵ_k are positive decreasing step sizes which are non-summable but square-summable. This SON has a similar behavior as (3.3) and also converges to a set on which all loads are equal as shown in [23, Theorem 5].

In practice, a projected SA algorithm (see Section 2.5) will be used instead of (3.8) because the CIOs are restricted to a maximum value. The restriction is mainly due to the fact that even with a eICIC mechanism, offloaded users suffer from residual interference caused by remaining control channels during ABSs. So, in order to limit the impact on control signals quality, users

must not be too far from their BS. However, the stability of the SA remains valid even in this case (see [15, §5.4]).

A small constant step size ϵ can also be used instead of ϵ_k . The convergence is then weaker as discussed in Section 2.5.3.

3.2.3 Backhaul-constrained heterogeneous networks scenario

Research on SON in general and LB in particular has mainly focused on the Radio Access Network (RAN) with the assumption of infinite backhaul capacity. However, finite backhaul capacity may impact the RAN performance in general and the operation of SON functions in particular.

Network operators carefully dimension the backhaul to avoid capacity bottlenecks and to ensure good end-to-end performance, while avoiding over-dimensioning due to both equipment and deployment costs. In existing networks, and particularly for low power nodes such as small cells but not only, performance issues due to finite backhaul capacity can be encountered in ADSL [69], or in wireless backhaul. In 5G networks, the high rate requirements for bandwidth intensive services [56] may cause saturation even in optical backhaul, unless very costly investment in the transport network are made. So backhaul must be considered when designing load balancing algorithms.

In order to take into account the impact of backhaul capacity on LB algorithms, the BS load definition should be modified to include the backhaul occupancy. Indeed, when the backhaul is saturated while the BS capacity remains sufficient, KPIs such as outage probability or File Transfer Time (FTT) may drastically deteriorate. In this case, the buffer of the BS may be empty since the radio link traffic flows faster than the backhaul traffic feeding the buffer.

3.2.3.1 Analytical load definitions

The load (3.5) for elastic traffic is rewritten taking into account the state of the backhaul as follows

$$\rho_{e,g} = \min \left(1, \int_A \frac{\lambda(r)\mathbb{E}(\sigma)}{\min(C_{BH}, R(r))} dr \right) \quad (3.9)$$

where C_{BH} is the capacity of the backhaul reserved for the RAN traffic. The subscript g stands for *global*, taking into account both BS and backhaul, as opposed to *local*.

The rationale behind Equation (3.9) is that the limited backhaul capacity may limit the peak data rate of a UE when alone in the cell. Hence Equation (3.5) should be modified by replacing $R(r)$ with $\min(C_{BH}, R(r))$. This modification is validated through simulation results in Section 3.2.4 where we compare the adjusted load formula (3.9) (in dashed lines in Figures 3.4 and 3.5) with the actual loads observed in the simulated system (plain lines in the Figures).

The load definition in (3.9) is modified if GBR traffic is also considered as follows

$$\rho_g = \min \left(1, \rho_{GBR,g} + \int_A \frac{\lambda(r)\mathbb{E}(\sigma)}{\min(\bar{C}_{BH}, \bar{R}(r))} dr \right). \quad (3.10)$$

\bar{C}_{BH} is the remaining backhaul capacity after backhaul resources have been allocated to GBR traffic. If we denote by D_{GBR} the total traffic demand of GBR users then $\bar{C}_{BH} = \max(0, C_{BH} - D_{GBR})$. $\rho_{GBR,g} = \max(\frac{D_{GBR}}{C_{BH}}, \rho_{GBR})$ is the global load of GBR traffic.

3.2.3.2 Load estimators

In practice, a simple load estimator can be derived, based on scheduler measurements. Consider first only elastic traffic and assume that all the resources are occupied even if only one user is present. Then the load can be estimated by the proportion of time that at least one user is present in the cell:

$$\rho_e = \sum_{t=1}^T \frac{\mathbf{1}_{\{x_e(t)>0\}}}{T} \quad (3.11)$$

where $x_e(t)$ is the number of users at time slot t .

If we consider a mixed traffic scenario with priority given to GBR traffic, we get a general load estimator $\hat{\rho}_g$ that reads

$$\hat{\rho}_g = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\{x_e(t)>0\}} + \mathbf{1}_{\{x_e(t)=0\}} \max(\gamma(t), \rho_{BH}(t)) \quad (3.12)$$

where $\gamma(t)$ is the proportion of resources used by the GBR traffic at time slot t in the RAN, and $\rho_{BH}(t)$ is the occupancy of the backhaul at time slot t .

In summary, Equation (3.12) is based on the basic load definition given by the average ratio between the resources used and the total resources in the BS (the ratio in each time step corresponds to a term in the sum). If at least one user with elastic traffic is present (first term in the sum), all the resources are used, whether or not GBR users are present. If there are only GBR users, one needs to consider the fraction of resources used at each time step, which is given by the maximum between the GBR load at the backhaul and at the radio access (the second term in the sum).

The LB algorithm is then rewritten as follows

$$CIO_{dBs}(k+1) = CIO_{dBs}(k) + \epsilon_k(\hat{\rho}_{g,0}(k) - \hat{\rho}_{g,s}(k)) \quad (3.13)$$

where $\hat{\rho}_{g,s}$ and $\hat{\rho}_{g,0}$ are the global loads of cell s and the reference cell 0 respectively evaluated using Eq. (3.12). It is noted that the convergence proof of (3.13) is the same as in [23].

3.2.4 Validation through simulation results

We present in this section some preliminary simulation results validating the new load definitions proposed in the previous subsection. Consider a LTE network comprising a trisector macro BS surrounded by 6 interfering macro BSs. We select one sector and place in it 4 small cells (see Figure 3.2). We consider only elastic traffic and evaluate the performance for the selected macro sector and the small cells inside its coverage area.

Two layers of traffic are superposed: the first one has a uniform arrival rate of λ users/s in the entire area (grey area in Figure 3.2). The second one has a uniform arrival rate of λ_h users/s in the initial area covered by the small cells (with all CIOs set to 0dB), namely the small cells are deployed to serve the users in the hotspot areas.

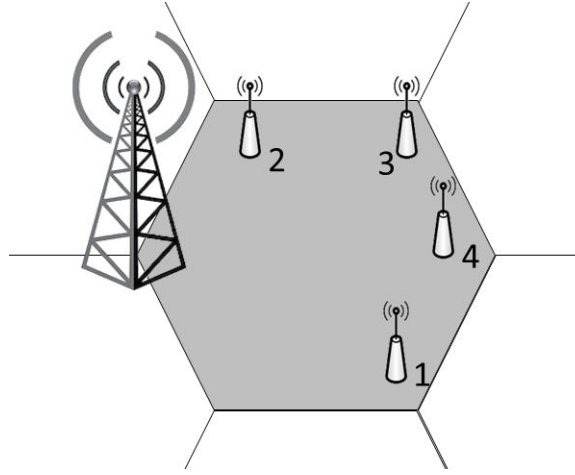


Figure 3.2: Network Layout

To illustrate the impact of a bottleneck at the backhaul, we assume a low backhaul capacity of 10 Mbps. The propagation models for the macro BSs and the small cells (following [1, Page 61]) are presented in Table 3.1 which also summarizes the simulation parameters.

We simulate the system during 3 hours and compare side-by-side the performance obtained using Algorithms (3.8) and (3.13) denoted respectively as *Local SON* and *Global SON*. We present the results for the macro sector and for two of the small cells. We plot the analytical and the estimated loads in dashed and plain lines respectively, using the different definitions (see Figures 3.4 and 3.5). We also plot the CIOs (Figure 3.3) set by the algorithms over time and the corresponding FTTs (Figure 3.6).

The local SON balances the BSs' scheduler loads (see Figure 3.4a) while it is unable to balance the real loads (see Figure 3.4b). In particular, small cell 1 increases excessively its coverage area (see red curve in Figure 3.3a) which causes its FTT to explode (red curve in Figure 3.6a).

On the other hand, the Global SON balances the real loads (see Figure 3.5b) by limiting the increase in small cells' CIOs (see Figure 3.3b). As a consequence, the small cells' FTT remain low while the macro cell's FTT is decreased (see Figure 3.6b).

It is noted that the size of small cell 2 is initially small because of its proximity to the macro

Table 3.1: Network and Traffic Parameters

Network parameters	
Number of macro sectors	1
Number of small cells	4
Number of interfering macros	6×3 sectors
Macro Cell layout	hexagonal trisector
Small Cell layout	omni
Intersite distance	500 m
Bandwidth	20MHz
Channel characteristics	
Thermal noise	-174 dBm/Hz
Macro Path loss (d in km)	$128 + 36.4 \log_{10}(d)$ dB
small cell Path loss (d in km)	$140.7 + 36.7 \log_{10}(d)$ dB
Traffic characteristics	
Traffic spatial distribution	uniform
λ	8 users/s
λ_h	4 users/s
Service type	FTP
Average file size	4 Mbits

cell. The increase in its CIO does not increase too much its size which is why its performance remains good even with local SON as shown in Figure 3.6a. Figures 3.4 and 3.5 also show the accuracy of the proposed load estimators (plain lines) compared to the analytical loads (dashed lines).

The overall user performance in terms of Mean User Throughput (MUT) and Cell-Edge throughput (CET) (see Figure 3.7) also shows the superior performance of Global SON. At the beginning of the simulation period, the MUT is driven by the macro users which are more numerous. With the activation of the LB algorithms, the macro cell is progressively offloaded by the small cells, thus the MUT improves for both the local SON and the Global SON. When the real loads are balanced, the global SON stops increasing the small cells coverage thus ensuring that the MUT remains good. The local SON on the other hand continues to increase the small cells coverage in order to balance the scheduler loads. This leads to the backhaul saturation of certain small cells which see their performance degrade drastically and consequently, to the degradation of the overall MUT. The same behavior is observed for the CET but this time the performance degradation for the local SON occurs earlier because cell edge users are generally more impacted by an overload in the system. It is noted that the proposed algorithm (3.13) has been proven robust to non-stationary traffic demands through extensive simulations.

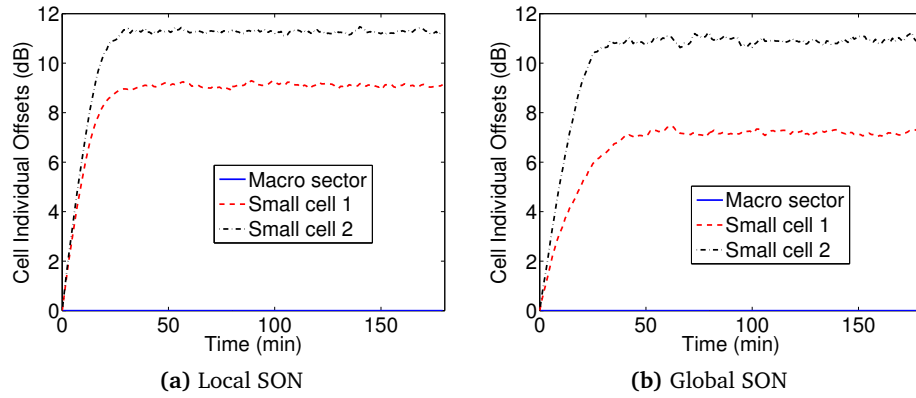


Figure 3.3: Cell Individual Offsets

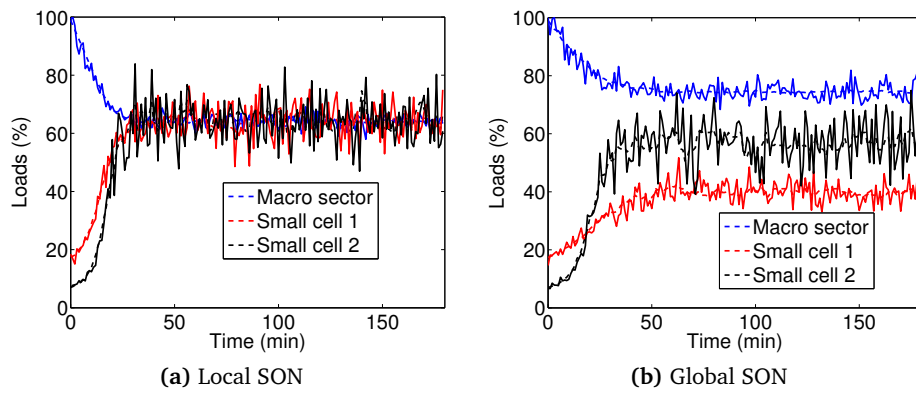


Figure 3.4: Local loads using Eq. (3.6) (dashed lines) and Eq. (3.7) (plain lines)

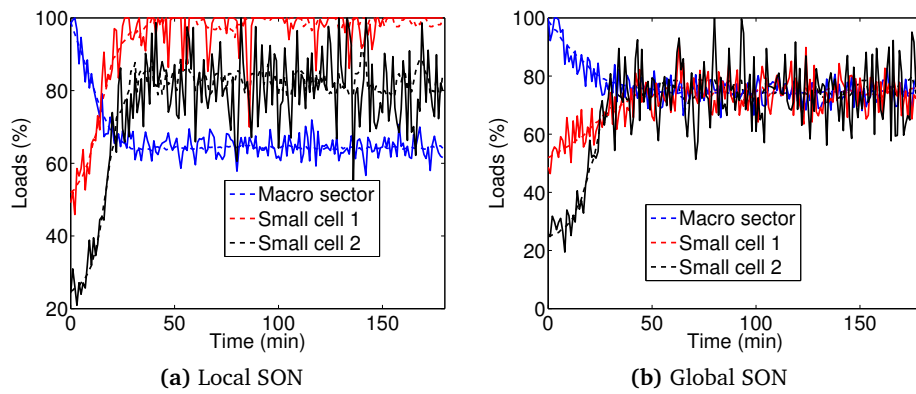


Figure 3.5: Global loads using Eq. (3.9) (dashed lines) and Eq. (3.11) (plain lines)

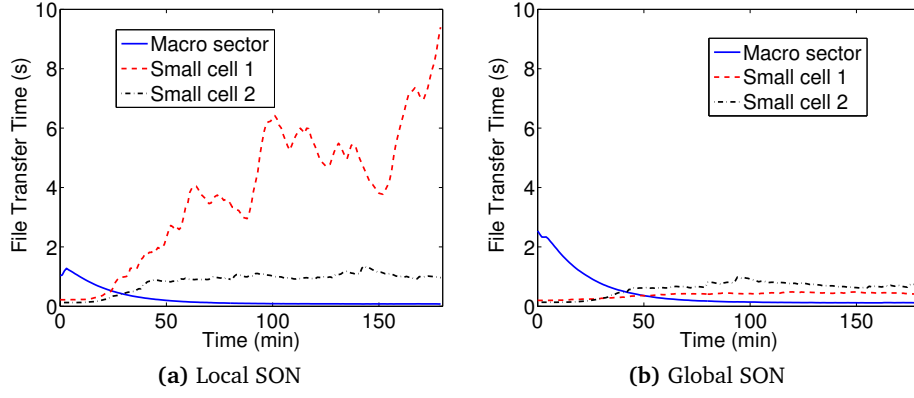


Figure 3.6: File Transfer Times comparison with Local SON or Global SON

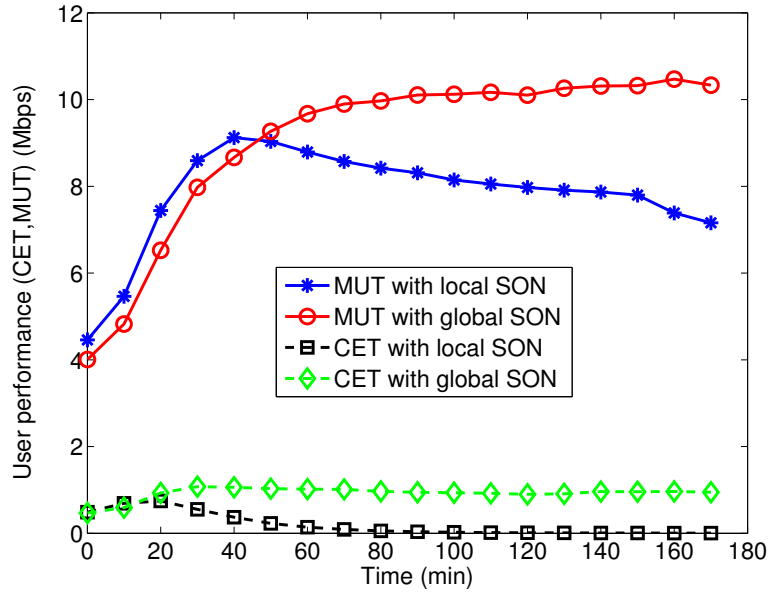


Figure 3.7: Time evolution of MUT and CET for the cluster of the macro BS and the 4 small cells

3.3 ABS ratio optimization

3.3.1 ABS mechanism

The ABS mitigation method consists in a time-domain interference avoidance. The goal is to reduce the interference from an aggressor cell (the cell causing the interference, in our case - the offloaded macro BS) by almost blanking out some of its sub-frames as shown in Figure 3.8.

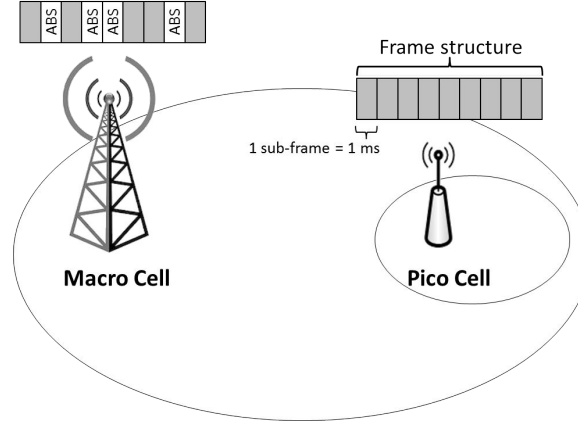


Figure 3.8: Illustration of Almost Blank Sub-Frames in a HetNet

During the ABSs, the aggressor cell mutes all of its traffic channels, leaving only some control channels which are transmitted with reduced power. The ABSs allow victim cells (namely the interfered small cells) to serve their users with almost no interference from the aggressor cell. The residual interference caused by the remaining control signals limits the gain from the ABSs, and can be further mitigated using additional interference cancellation techniques introduced in 3GPP release 11 [4, Section 16.1.5].

It is advisable to schedule cell edge users during ABSs because they are the most affected by interference. However, all the small cell users can benefit from ABSs since their strongest interferer is generally a macro BS. The use of ABSs decreases the available resources of the macro BS. Hence a trade-off should be found between the capacity gain of the small cells brought about by the ABSs and the corresponding capacity losses of the macro BS.

One may choose to schedule small cell UEs in the CRE area exclusively during the ABSs, and schedule the other small cell UEs only during non-ABSs. Alternatively, one may schedule all small cells' UEs during both ABSs and non-ABSs. We now discuss these two possible implementations.

3.3.2 Implementation alternatives

3.3.2.1 Protection of offloaded users

The first implementation for eICIC aims at protecting only the offloaded users at the CRE area of the small cells. Small cell users are divided into two groups: CRE users who are attached to the small cell due to the CIO, and center cell users who are attached to the small cell even when the

CIO is set to 0dB. The CRE users will be served by the small cell during the macro cell ABSs. Thus a strictly positive CIO must be accompanied by a strictly positive ABSr, otherwise the users in the CRE area will never be served. We say that the two parameters (CIO and ABSr) are coupled.

Consider a CRE user u and determine the SINR gain when he is offloaded from the macro cell to the small cell and is scheduled during ABSs. Denote by the subscript m the macro BS and by p - the small cell. When attached to the macro BS m , user u has a SINR equal to

$$S_{u,m} = \frac{h_m(u)P_m}{h_p(u)P_p + C_0(u)} \quad (3.14)$$

where $h_m(u)$ and $h_p(u)$ are the pathlosses from the macro cell m to user u and from the small cell p to user u , respectively. P_m and P_p are the macro cell and small cell transmit powers, respectively. $C_0(u)$ is the total interference generated by the other nodes in the network (other macro BSs and small cells) plus the thermal noise at the receiver of user u . If user u is offloaded to the small cell and is served only during the ABSs of macro cell m , then its SINR becomes

$$S_{u,p} = \frac{h_p(u)P_p}{C_0(u)} \quad (3.15)$$

The SINR gain for this user (ratio between $S_{u,p}$ and $S_{u,m}$) can then be written as

$$\begin{aligned} SG_u &= \frac{h_p(u)P_p(h_p(u)P_p + C_0(u))}{h_m(u)P_m C_0(u)} \\ &= \frac{h_p(u)P_p}{h_m(u)P_m} + \frac{(h_p(u)P_p)^2}{h_m(u)P_m C_0(u)} \end{aligned} \quad (3.16)$$

From equation (3.16) we can deduce the following simple condition on the received signals from the different BSs at every position for obtaining an offloading gain using ABSs

$$\frac{(h_p(u)P_p)^2}{h_m(u)P_m - h_p(u)P_p} > C_0(u) \quad (3.17)$$

If the condition (3.17) is not satisfied, offloading will result in performance degradation. Furthermore, there is a maximum CIO above which there is no gain for certain small cell users since as one gets further away from the small cell, $C_0(u)$ increases.

Condition (3.17) considers that only one macro cell implements ABSs. If M macro cells synchronously implement ABSs, their interference will be removed from $C_0(u)$ so that (3.17) becomes

$$\frac{h_p(u)P_p \left(h_p(u)P_p + \sum_{k=1, k \neq m}^M h_k(u)P_k \right)}{h_m(u)P_m - h_p(u)P_p} > C_0(u) \quad (3.18)$$

which is satisfied for a larger number of CRE users.

We now give a simple example in which we can find to which extent a CRE can be performed, i.e. where offloading provides SINR gains. We consider a trisector macro cell site with one small

cell in the coverage area of one of the macro sectors. To take into account neighbours' interference, we add a tier of trisector macro sites to this cluster.

We focus on the SINR gains for users in the CRE area. By varying the number of macro BSs applying ABSs for the small cell users, we can see the evolution of the maximum CIO above which there is a SINR degradation at the edge of the small cells. The macros considered for muting are the most interfering ones.

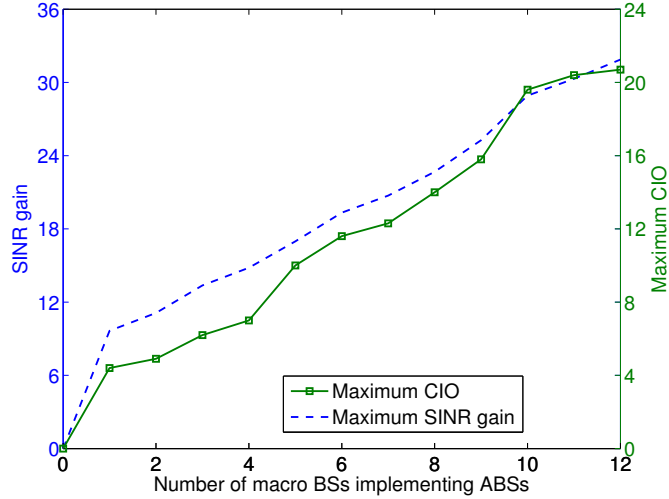


Figure 3.9: Maximum CIO for SINR improvement and Maximum SINR gain as a function of the number of macro BSs applying ABSs

Figure 3.9 presents the maximum CIO as a function of the number of macro BSs providing ABSs (green line with squares). It also shows the mean SINR gain obtained by the CRE users when setting the CIO to its best value i.e. the CIO providing the highest SINR gain. The Figure illustrates condition (3.18), namely that muting more macro BSs allows us to further increase the CIO. We can also see that the SINR gain obtained by muting the most interfering macro cell is more important than the additional gain from muting more macro cells. Starting from two muted macro cells, the SINR gain becomes almost linear with the number of muted macro cells. From this curve, it is clear that a trade-off between complexity (number of macro cells muted) and SINR gain must be found.

It is noted that if the small cells' load is low, offloading is still possible without applying the ABS thus preserving the macro cell resources. We present in the next section an implementation that allows to achieve this goal.

3.3.2.2 Protection of all small cell users

Instead of allowing only the CRE users to profit from ABS, in this implementation all the small cells' users can benefit from the ABS, namely they can be scheduled during ABS or during normal sub-frames. In this case, offloading by increasing the CIOs of the small cells without applying ABSs is also allowed. We can say that the two parameters (CIO and ABSr) are decoupled. However if

needed, the macro cell can provide ABS in order to enhance the offloading capability of the small cells. The SINR during ABSs period is the same as that in the previous implementation. Only now, a small cell user will benefit part of the time from ABSs and the rest of the time will experience normal SINR. The mean throughput of a macro user can then be written as

$$\bar{T}_{u,m} = (1 - \theta)\bar{R}_{u,m} \quad (3.19)$$

where θ is the ABSr of macro m and $\bar{R}_{u,m}$ - the mean data rate of user u when it is served by macro cell m . The mean throughput of a small cell user is

$$\bar{T}_{u,p} = (1 - \theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}} \quad (3.20)$$

where θ is the ABSr available to small cell p , $\bar{R}_{u,p}^{\text{no ABS}}$ and $\bar{R}_{u,p}^{\text{ABS}}$ are the average data rates of user u over time (i.e. averaged over fast-fading and scheduling) when served by small cell p outside and during the ABS periods respectively.

An efficient setting of the ABSr is needed to optimize the system performance. A condition for achieving an offloading gain using ABSs could be the increase of the total throughput of the considered cluster (macro cells and small cells)

$$\begin{aligned} \sum_{m \in \mathcal{M}} \sum_{u \in m} (1 - \theta)\bar{R}_{u,m} + \sum_{p \in \mathcal{P}} \sum_{u \in p} (1 - \theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}} \\ \geq \sum_{m \in \mathcal{M}} \sum_{u \in m} \bar{R}_{u,m} + \sum_{p \in \mathcal{P}} \sum_{u \in p} \bar{R}_{u,p}^{\text{no ABS}} \end{aligned} \quad (3.21)$$

where \mathcal{M} and \mathcal{P} are the set of all macro and small cells involved in the mechanisms (CRE and ABS).

Condition (3.21) may be too restrictive as we may want to increase the CET at the expense of a decrease in total throughput. In the next section, we propose self-optimization algorithms based on the two implementations described in this section, using a PF utility of UE throughputs as objective function.

3.3.3 Exact Proportional-Fair algorithm

We choose to implement the ABSrO algorithm at the small cell which then requests appropriate ABSr from its interfering macro BSs. The macro BSs receives ABSr requests from the small cells it interferes and then applies the maximum ABSr among these requests. The ABSr optimization algorithm should then take into account load or traffic conditions (i.e. number of users present in the cell) of all the macro cells from which the considered small cell will be requesting ABSs.

The cluster of BSs considered for a single ABSrO algorithm comprises a small cell p on which the algorithm is implemented, and the most interfering M macro cells with small cell p . We assume that all macro cells implementing ABS are synchronized. Typically $M = 1$ is sufficient if the small cell is in the center of the macro cell, namely by periodically muting only one macro cell,

we increase significantly the SINR of the small cell users. When the small cell is located at the cell edge, choosing $M = 3$ provides better results.

3.3.3.1 Only CRE users are protected

Using CRE, the small cells are able to offload traffic from the macro BS. The offloaded users at the small cell edge are highly interfered by the macro that previously served them. ABSs are used to mitigate the interference enabling the small cells to offload even more the macro cell traffic.

The objective function considered in this implementation is the Proportional Fair (PF) defined as follows

$$U_{PF1}(\theta) = \sum_{m=1}^M \sum_{u \in m} \log((1-\theta)\bar{R}_{u,m}) + \sum_{u \in \text{center of } p} \log((1-\theta)\bar{R}_{u,p}^{\text{no ABS}}) + \sum_{u \in \text{CRE of } p} \log(\theta\bar{R}_{u,p}^{\text{ABS}}) \quad (3.22)$$

where we considered the M most interfering macro BSs users' throughputs, the small cell center users' throughputs and the small cell CRE area users' throughputs. The PF utility enables us to maximize the users throughput and to enforce fairness among them. The SON algorithm applied to optimize this PF utility is given in the following theorem

Theorem 2. *Given Assumption 1 and the update equation*

$$\theta_{k+1} = \theta_k + \epsilon_k \left(\frac{N_{p,CRE}}{\theta} - \frac{N_{p,CEN} + \sum_{m=1}^M N_m}{1-\theta} \right) \quad (3.23)$$

where N_m , $N_{p,CEN}$ and $N_{p,CRE}$ are the numbers of active users in cell m , the center of cell p and the CRE area of cell p , respectively,

then θ_k converges to a set on which $U_{PF1}()$ is maximal.

Proof. See Appendix A.1. □

Note that the optimal θ can be directly derived in this case using (A.2) [79, Eq. 8], and is equal to

$$\theta^* = \frac{N_{p,CRE}}{N_{p,CRE} + N_{p,CEN} + \sum_{m=1}^M N_m} \quad (3.24)$$

The reason we may use a SA algorithm instead of setting optimal θ is to keep a certain stability in the parameter configuration which can be quite critical in real networks. Figure 3.10 illustrates this statement. Setting optimal ABSr at each event in the network (arrival or departure of a user) yields an extremely fluctuating parameter whereas the SA approach allows us to freeze the parameter at convergence when the traffic is stationary. However in our numerical results, we implemented the more dynamic approach in order to have an unbiased comparison between the different algorithms.

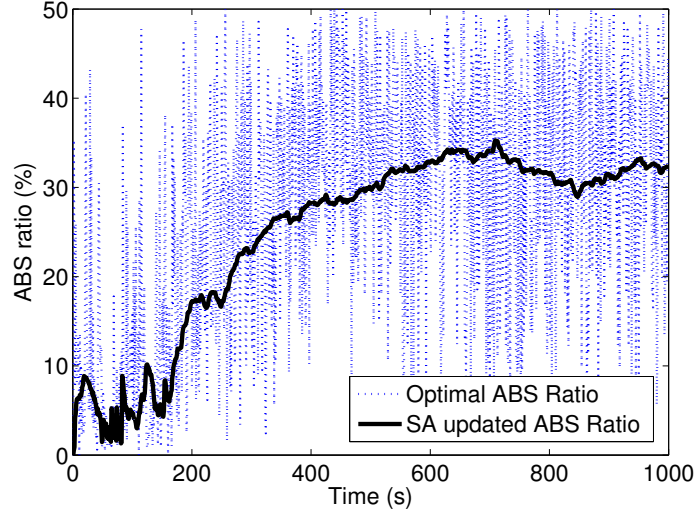


Figure 3.10: Time evolution of ABS ratio using stochastic approximation (solid line) and optimal solution (dashed line)

3.3.3.2 All small cell users benefit from ABSs

In this case, the user throughputs are modified according to Equations (3.19) and (3.20). The PF utility of the user throughputs is redefined as

$$U_{\text{PF2_exact}}(\theta) = \sum_{m=1}^M \sum_{u \in m} \log((1 - \theta)\bar{R}_{u,m}) + \sum_{u \in p} \log((1 - \theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}}) \quad (3.25)$$

since we make no distinction between CRE users of the small cell and its normal users. The SON algorithm for optimizing the utility function (3.25) is presented in the following theorem.

Theorem 3. *Given Assumption 1 and the update equation*

$$\theta_{k+1} = \theta_k + \epsilon_k \left(\sum_{u \in p} \frac{1}{\theta_k + \frac{\bar{R}_{u,p}^{\text{no ABS}}}{\bar{R}_{u,p}^{\text{ABS}} - \bar{R}_{u,p}^{\text{no ABS}}}} - \frac{\sum_{m=1}^M N_m}{1 - \theta_k} \right) \quad (3.26)$$

where N_m is the number of active users in cell m ,

then θ_k converges to a set on which $U_{\text{PF2_exact}}()$ is maximal.

Proof. See Appendix A.2. □

3.3.4 Lower-bound approximated PF algorithm

The update equation (3.26), requires the knowledge of the average data rates of all pico users, rendering practical implementation of the algorithm more complex. To simplify (3.26), we choose

to maximize a lower bound of (3.25) instead. The new objective function is written as

$$\begin{aligned}
 U_{PF2}(\theta) = & \sum_{m=1}^M \sum_{u \in m} \log((1-\theta)\bar{R}_{u,m}) \\
 & + \sum_{u \in p} \frac{1}{2} \log(2(1-\theta)\bar{R}_{u,p}^{\text{no ABS}}) \\
 & + \sum_{u \in p} \frac{1}{2} \log(2\theta\bar{R}_{u,p}^{\text{ABS}})
 \end{aligned} \tag{3.27}$$

Lemma 1. *Let us consider M macro cells and one small cell indexed p .*

Then $\forall \theta \in]0, 1[$ we have

$$U_{PF2}(\theta) \leq U_{PF2_exact}(\theta) \tag{3.28}$$

Proof. For the proof it suffices to show that for a small cell user u ,

$$\begin{aligned}
 & \log(2(1-\theta)\bar{R}_{u,p}^{\text{no ABS}}) + \log(2\theta\bar{R}_{u,p}^{\text{ABS}}) \\
 & \leq 2\log((1-\theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}})
 \end{aligned} \tag{3.29}$$

For ease of notation, denote $a = (1-\theta)\bar{R}_{u,p}^{\text{no ABS}}$ and $b = \theta\bar{R}_{u,p}^{\text{ABS}}$. We want to show that $\log 2a + \log 2b \leq 2\log(a+b)$. Using Jensen's inequality [39] for the function $-\log$ which is convex we have

$$-\log\left(\frac{1}{2}(2a) + \frac{1}{2}(2b)\right) \leq -\frac{1}{2}\log(2a) - \frac{1}{2}\log(2b) \tag{3.30}$$

By taking the negative of (3.30), we obtain the desired result. \square

The SON algorithm optimizing (3.27) is presented in the following theorem.

Theorem 4. *Given Assumption 1 and the update equation*

$$\theta_{k+1} = \theta_k + \epsilon_k \left(\frac{N_p}{2\theta_k} - \frac{\frac{N_p}{2} + \sum_{m=1}^M N_m}{1-\theta_k} \right) \tag{3.31}$$

*where N_m and N_p are the numbers of active users in cells m and p respectively,
then θ_k converges to a set on which $U_{PF2}()$ is maximal.*

Proof. See Appendix A.3. \square

The reasons for implementing a SA algorithm instead of setting the optimal ABS ratio (which can be easily derived) are the same as those stated in the previous section.

The SON algorithm (3.31) (when all small cell users are protected) presents certain advantages over the one in (3.23) (when only CRE users are protected). The first one is that we do not need to keep two counters for the numbers of active users of the small cell, but only one for the total number of users. The second one is that if the small cell has a very low load compared to those of its surrounding macro BSs, the ABSr provided by those macro BSs can be also low, thus preserving

the resources of the macro BSs. Furthermore, (3.31) is completely decoupled from the load balancing SON, whereas in (3.23), a positive CIO requires a corresponding ABSr.

3.3.5 General α -fair algorithm

In this subsection, we extend the algorithms from the previous subsections to the general α -fair utility. These algorithms provide good trade-offs between the CET of pico users and the MUT of macro users. The α -fair utility function for this case can be redefined as follows;

- for $\alpha = 1$:

$$U_\alpha(\theta) = \sum_{m=1}^M \sum_{u=1}^{N_m} \log((1-\theta)\bar{R}_{u,m}) + \sum_{p=1}^P \sum_{u=1}^{N_p} \log((1-\theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}}) \quad (3.32)$$

- and for $\alpha \neq 1$:

$$U_\alpha(\theta) = \sum_{m=1}^M \sum_{u=1}^{N_m} \frac{((1-\theta)\bar{R}_{u,m})^{1-\alpha}}{1-\alpha} + \sum_{p=1}^P \sum_{u=1}^{N_p} \frac{((1-\theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}})^{1-\alpha}}{1-\alpha} \quad (3.33)$$

The optimization algorithm for this class of utility functions is presented in the following theorem.

Theorem 5. *Given Assumption 1 and the ABSr θ updated using Algorithm 3, then θ converges to θ^* such that $U_\alpha(\theta^*) \geq U_\alpha(\theta) \forall \theta \in [0, 1]$.*

Proof. See Appendix A.4. □

3.3.6 Load minimization algorithm

In this subsection, we present an ABSrO algorithm for which the objective is to minimize the maximum of the loads. This algorithm has been implemented together with the load balancing algorithm (3.8) in an autonomic-SDN demonstrator that is described in Section 3.6. We assume here that only CRE users are protected.

In terms of network stability, the load of a cluster of BSs is obtained by taking the maximum of all loads in this cluster.

$$\rho(\theta) = \max\{\rho_{p_i}(\theta), \rho_{m_j}(\theta), i = 1 \dots P, j = 1 \dots M\} \quad (3.36)$$

Algorithm 3 α -Fair ABSrO

-
- 1: *Initialization:*
 - 2: $\theta \leftarrow \theta_0$ where $\theta_0 < 1$ and $\theta_0 \geq 0$
 - 3: *loop:*
 - 4: **for** $k \in \mathbb{N}, k > 0$ **do**
 - 5: $\theta \leftarrow \theta + \epsilon_k \frac{\partial U_\alpha(\theta)}{\partial \theta}$
-

The derivatives are given as follows

- For $\alpha = 1$

$$\frac{\partial U_\alpha(\theta)}{\partial \theta} = \sum_{p=1}^P \sum_{u \in p} \frac{1}{\theta + \frac{\bar{R}_{u,p}^{\text{no ABS}}}{\bar{R}_{u,p}^{\text{ABS}} - \bar{R}_{u,p}^{\text{no ABS}}}} - \frac{\sum_{m=1}^M N_m}{1 - \theta} \quad (3.34)$$

- For $\alpha \neq 1$

$$\begin{aligned} \frac{\partial U_\alpha(\theta)}{\partial \theta} = & - \sum_{m=1}^M \sum_{u \in m} \bar{R}_{u,m}^{1-\alpha} (1-\theta)^{-\alpha} + \sum_{p=1}^P \sum_{u \in p} \left[(\bar{R}_{u,p}^{\text{ABS}} \right. \\ & \left. - \bar{R}_{u,p}^{\text{no ABS}}) \left((1-\theta) \bar{R}_{u,p}^{\text{no ABS}} + \theta \bar{R}_{u,p}^{\text{ABS}} \right)^{-\alpha} \right] \end{aligned} \quad (3.35)$$

where θ is the ABSr set synchronously by macros $1 \dots M$ for picos $1 \dots P$. Let us first characterize this load in terms of the ABSr. BSs are considered as processor sharing queues in which users arrive according to a Poisson process, download a file of size σ and leave the network. The BSs' load can be analytically expressed as in Equation (3.5). Coming back to our use case, the different loads mentioned in (3.36) can be expressed as

$$\rho_{m_j}(\theta) = \int_{A_{m_j}} \frac{\lambda(r) \mathbb{E}[\sigma]}{(1-\theta) \bar{R}_{m_j}(r)} dr \quad (3.37)$$

$$\begin{aligned} \rho_{p_i}(\theta) = & \int_{\text{CRE}_{p_i}} \frac{\lambda(r) \mathbb{E}[\sigma]}{\theta \bar{R}_{p_i}(r)} dr \\ & + \int_{A_{p_i} - \text{CRE}_{p_i}} \frac{\lambda(r) \mathbb{E}[\sigma]}{(1-\theta) \bar{R}_{p_i}(r)} dr \end{aligned} \quad (3.38)$$

Since ρ_{m_j} and ρ_{p_i} are convex in θ , we have that the objective function ρ defined in (3.36) is convex as it is the maximum of convex functions. The ABSrO algorithm is given in the following theorem.

Theorem 6. *Given Assumption 1 and the update equation*

$$\theta_{k+1} = \theta_k - \epsilon_k \frac{\partial \rho_{b^*}(\theta_k)}{\partial \theta}, \quad (3.39)$$

where $b^* = \arg \max_{b \text{ in the cluster}} \rho_b$, we have

$$\lim_{k \rightarrow \infty} \theta_k^{BEST} - \theta^* = 0 \quad (3.40)$$

where θ^* is the minimizer of $\rho_{b^*}(\theta)$ (i.e. $\rho_{b^*}(\theta^*) \leq \rho_{b^*}(\theta) \forall \theta$, and θ_k^{BEST} is the best parameter obtained so far (i.e. $\rho_{b^*}(\theta_k^{BEST}) \leq \rho_{b^*}(\theta_j) \forall j = 0 \dots k$),

Proof. See Appendix A.5. □

If a constant step size is used instead, we only get bounds on the maximal distance from optimal point at convergence. We will now discuss how to evaluate $\frac{\partial \rho_{b^*}(\theta)}{\partial \theta}$ relying on the *Envelope Theorem*.

Let us denote by $\rho_{\text{CRE},p}$ and $\rho_{\text{NOR},p}$ the loads of BS p generated by the users in the CRE area, and the normal users (center area) respectively. So we will have

$$\rho_{\text{CRE},p} = \int_{\text{CRE}_{p_i}} \frac{\lambda(r) \mathbb{E}[\sigma]}{\theta \bar{R}_{p_i}(r)} dr \quad (3.41)$$

and

$$\rho_{\text{NOR},p} = \int_{A_{p_i} - \text{CRE}_{p_i}} \frac{\lambda(r) \mathbb{E}[\sigma]}{(1 - \theta) \bar{R}_{p_i}(r)} dr \quad (3.42)$$

From these expressions we can easily derive derivatives of (3.37) and (3.38) as follows

$$\begin{aligned} \frac{\partial \rho_{m_j}(\theta)}{\partial \theta} &= \frac{\rho_{m_j}(\theta)}{1 - \theta} \\ \frac{\partial \rho_{p_i}(\theta)}{\partial \theta} &= \frac{\rho_{\text{NOR},p}(\theta)}{1 - \theta} - \frac{\rho_{\text{CRE},p}(\theta)}{1 - \theta} \end{aligned} \quad (3.43)$$

The sub-gradient used in (3.39) is then evaluated using (3.43) where the loads are estimated by the resource utilization in the BS. The estimators of the loads are then noisy and so is the sub-gradient. But SA allows us to get almost sure convergence results as discussed in Section 2.5.

3.4 Frequency splitting optimization

The second deployment scenario considered for HetNets is orthogonal deployment. The small cells and macro cells operate on disjoint frequency bands. This solution has been extensively studied in [31] for a static traffic scenario. The authors in [31] show that Orthogonal Deployment (OD) improves cell capacity compared to Co-Channel Deployment (CCD) if a proper user association mechanism is used. Our focus in this paper is the dynamic optimization case. Let us denote the split factor by δ , so the bandwidth available will be $(1 - \delta)W$ for macro cells, and δW for small cells. W is the total bandwidth of the system.

We suppose that all the power available is reused on the reduced band. So, the total power budget remains constant, but the power per Hertz increases. The SINR for a macro user becomes

$$\Gamma_{u,m}(r) = \frac{P_m h_m(r)}{(1-\delta)N_0 + \sum_{k=1, k \neq m}^M P_k h_k(r)}. \quad (3.44)$$

The SINR for a pico user will now be

$$\Gamma_{u,p}(r) = \frac{P_p h_p(r)}{\delta N_0 + \sum_{j=1, j \neq p}^P P_j h_j(r)}. \quad (3.45)$$

We can see that the SINR is further increased with OD, and this is due to the fact that the BSs received signal strength will increase relatively to the thermal noise (see Equations (3.44), (3.45)). This is particularly true for the small cells' users when the fraction of bandwidth allocated to the small cells is small so that they transmit with higher power on a smaller bandwidth.

However, compared to CCD, the available bandwidth for OD is reduced. As in the case of the ABS mechanism, the frequency split factor δ must be optimized. We present in the next section various self-optimizing algorithms to maximize α -fair utilities of users throughputs.

3.4.1 Exact α -fair algorithms

For the case of OD, the α -fair utility functions have a more complex dependence with the split factor since the SINR of each user now depends on δ as shown in equations (3.44) and (3.45). The utility functions can be rewritten as follows:

- For $\alpha = 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \log((1-\delta)R(\Gamma_{u,m}(\delta))) + \sum_{p=1}^P \sum_{u=1}^{N_p} \log(\delta R(\Gamma_{u,p}(\delta))) \quad (3.46)$$

where $\Gamma_{u,m}$ and $\Gamma_{u,p}$ are the users' SINRs given respectively by (3.44) and (3.45), and $R(\cdot)$ is the average data rate function defined in (3.1).

- For $\alpha \neq 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \frac{[(1-\delta)R(\Gamma_{u,m}(\delta))]^{1-\alpha}}{1-\alpha} + \sum_{p=1}^P \sum_{u=1}^{N_p} \frac{[\delta R(\Gamma_{u,p}(\delta))]^{1-\alpha}}{1-\alpha} \quad (3.47)$$

It is noted here that the SINRs are formulated in terms of δ as follows

$$\Gamma(\delta) = \frac{\xi}{\beta \delta + \kappa} \quad (3.48)$$

where $\xi > 0$ is proportional to the received signal, $\beta = N_0 > 0$ for pico users and $\beta = -N_0 < 0$ for macro users, κ is a real constant independent of δ . The function $U_\alpha(\delta)$ is continuous, differentiable and concave (see Appendix A.6) on $\delta \in (0, 1)$. By the KKT conditions for optimality, its maximum is attained at δ^* such that $\frac{\partial U_\alpha(\delta^*)}{\partial \delta} = 0$. The derivatives of $U_\alpha(\delta)$ with respect to δ are given as follows for different values of α :

- For $\alpha = 1$

$$\begin{aligned} \frac{\partial U_\alpha(\delta)}{\partial \delta} = & \sum_{p=1}^P \left[\frac{N_p}{\delta_k} - \sum_{u \in p} \frac{\Gamma_{u,p}^2 N_0}{\xi_{u,p}(1 + \Gamma_{u,p}) \log(1 + \Gamma_{u,p})} \right] \\ & - \sum_{m=1}^M \left[\frac{N_m}{1 - \delta_k} - \sum_{u \in m} \frac{\Gamma_{u,m}^2 N_0}{\xi_{u,m}(1 + \Gamma_{u,m}) \log(1 + \Gamma_{u,m})} \right] \end{aligned} \quad (3.49)$$

- For $\alpha \neq 1, \alpha \in \mathbb{N}$

$$\begin{aligned} \frac{\partial U_\alpha(\delta)}{\partial \delta} = & \sum_{p=1}^P \sum_{u \in p} \delta^{-\alpha} \bar{R}_{u,p}^{-\alpha} \left[\bar{R}_{u,p} - \frac{\eta W \delta \Gamma_{u,p}^2 N_0}{\log(2) \xi_{u,p}(1 + \Gamma_{u,p}) \log(1 + \Gamma_{u,p})} \right] \\ & - \sum_{m=1}^M \sum_{u \in m} (1 - \delta)^{-\alpha} \bar{R}_{u,m}^{-\alpha} \left[\bar{R}_{u,m} - \frac{\eta W (1 - \delta) \Gamma_{u,m}^2 N_0}{\log(2) \xi_{u,m}(1 + \Gamma_{u,m}) \log(1 + \Gamma_{u,m})} \right] \end{aligned} \quad (3.50)$$

where N_m and N_p are the numbers of active users in cells m and p respectively, $\xi_{u,c}$ and $\Gamma_{u,c}$ are respectively the received signal strength and the SINR of user u served by BS c .

We can see that solving $\frac{\partial U_\alpha(\delta)}{\partial \delta} = 0$ can be rather complex for all values of α especially because of the dependence of $\Gamma_{u,c}$ on δ . Also the data rates appearing in the expressions strongly fluctuate in practice so their measurements are noisy. All of these reasons justify once again the use of a SA to optimize the α -fair utilities which is presented in the following theorem.

Theorem 7. *Given Assumption 1 and δ updated using Algorithm 4, then δ converges to a value at which $U_\alpha(\delta)$ is maximal.*

Proof. See Appendix A.6. □

Algorithm 4 α -Fair SfO

- 1: Initialization:
 - 2: $\delta \leftarrow \delta_0$ where $\delta_0 \in (0, 1)$
 - 3: loop:
 - 4: **for** $k \in \mathbb{N}, k > 0$ **do**
 - 5: $\delta \leftarrow \delta + \epsilon_k \frac{\partial U_\alpha(\delta)}{\partial \delta}$ where the derivatives are given in Equations (3.49) and (3.50)
-

3.4.2 Lower-bound α -fair algorithms

The implementation of Algorithm 4 is complex and requires many measurements. In order to provide simpler algorithms, we use a lower bound to the SINR that can be obtained with OD. Indeed, $\frac{a}{b\delta+c} < \frac{a}{b+c} \forall \delta \in [0, 1], a > 0, b > 0$ and $c > 0$. So the α -fair utilities for the OD case are bounded below by

- For $\alpha = 1$:

$$U_{\alpha}^{\text{approx}}(\delta) = \sum_{m=1}^M \sum_{u \in m} \log((1-\delta)R(\Gamma'_{u,m})) + \sum_{p=1}^P \sum_{u=1}^{N_p} \log(\delta R(\Gamma'_{u,p})) \quad (3.51)$$

where $\Gamma'_{u,m}$ and $\Gamma'_{u,p}$ are the lower bounds to the users' SINRs obtained by replacing respectively $(1-\delta)N_0$ by N_0 in (3.44) and δN_0 by N_0 in (3.45).

- For $\alpha \neq 1$:

$$U_{\alpha}^{\text{approx}}(\delta) = \sum_{m=1}^M \sum_{u \in m} \frac{[(1-\delta)R(\Gamma'_{u,m})]^{1-\alpha}}{1-\alpha} + \sum_{p=1}^P \sum_{u=1}^{N_p} \frac{[\delta R(\Gamma'_{u,p})]^{1-\alpha}}{1-\alpha} \quad (3.52)$$

Using these lower bounds utilities, we can obtain much simpler algorithms in which the SINRs, the received signal or the thermal noise values are not needed. Indeed the functions (3.51) and (3.52) are continuous, differentiable and concave in $\delta \in (0, 1)$. The concavity follows from that of the functions $g_1 : x \mapsto \log(x)$ and $g_{\alpha} : x \mapsto \frac{x^{1-\alpha}}{1-\alpha}$. So the KKT conditions imply that the maximum of the utilities $U_{\alpha}^{\text{approx}}(\delta)$, $\forall \alpha > 0$ is attained at δ^* such that $\frac{\partial U_{\alpha}^{\text{approx}}(\delta^*)}{\partial \delta} = 0$. The derivatives of the lower bound utilities are given as follows:

- For $\alpha = 1$

$$\frac{\partial U_{\alpha}^{\text{approx}}(\delta)}{\partial \delta} = \sum_{p=1}^P \frac{N_p}{\delta_k} - \sum_{m=1}^M \frac{N_m}{1-\delta_k} \quad (3.53)$$

- For $\alpha \neq 1$

$$\frac{\partial U_{\alpha}^{\text{approx}}(\delta)}{\partial \delta} = \sum_{p=1}^P \sum_{u \in p} \delta^{-\alpha} \bar{R}_{u,p}^{1-\alpha} - \sum_{m=1}^M \sum_{u \in m} (1-\delta)^{-\alpha} \bar{R}_{u,m}^{1-\alpha} \quad (3.54)$$

The stochastic optimization algorithm for the lower bound utilities is given by the following theorem.

Theorem 8. *Given Assumption 1 and the split factor δ updated using Algorithm 5, then δ converges to a value at which $U_{\alpha}^{\text{approx}}(\delta)$ is maximal.*

Proof. The proof is straightforward as the function $U_\alpha^{\text{approx}}(\delta)$ is continuous, differentiable and concave for $\alpha > 0$. So the equivalent ODE of the SA is

$$\dot{\delta} = \frac{\partial U_\alpha^{\text{approx}}(\delta)}{\partial \delta} \quad (3.55)$$

by the same arguments as in Section 2.5. This ODE converges towards the maximum of $U_\alpha^{\text{approx}}(\delta)$.

The case $\alpha = 0$ is trivial since $U_\alpha^{\text{approx}}(\delta)$ is then linear (see the first paragraph of Appendix A.4). \square

Algorithm 5 α -Fair SfO

```

1: Initialization:
2:  $\delta \leftarrow \delta_0$  where  $\delta_0 \in (0, 1)$ 
3: loop:
4: for  $k \in \mathbb{N}, k > 0$  do
5:    $\delta \leftarrow \delta_k + \epsilon_k \frac{\partial U_\alpha^{\text{approx}}(\delta_k)}{\partial \delta}$ 

```

The case $\alpha = 0$ is not interesting because it will assign all the bandwidth to whichever provides the highest throughput between small cells and macro cells.

We can see that the lower bound utilities provide much simpler algorithms based only on the throughputs of the users. For the case $\alpha = 1$, we can even get the analytical maximizer of the lower bound utility which is

$$\delta^* = \frac{\sum_{p=1}^P N_p}{\sum_{m=1}^M N_m + \sum_{p=1}^P N_p} \quad (3.56)$$

3.4.3 eICIC algorithms implementation: Centralized or Distributed

The optimal implementation (in terms of interference) of the proposed algorithms would be to centralize all the information for all the cells considered, and perform the optimization in a centralized fashion. This way all macro BSs would be synchronized, minimizing the interference on the small cells. However this scenario implies a huge amount of signaling, and also limits the reactivity of the SON algorithms since the algorithm will run at the pace of information update in the central entity.

A distributed solution is to optimize the parameters locally at each BS. So each macro BS will only consider the small cells inside its coverage area, its neighboring macro cells and the small cells inside their coverage area. The algorithms are then run at each BS but considering all neighboring BSs. Also, resource allocation in each macro BS should be such that there is maximum overlap between resources affected to pico BSs (frequency band for OD or subframes for ABS). This way, inter-layer interference is minimized.

This could be achieved by imposing that macro BSs occupy for example the lower part of the bandwidth in the case of OD, or the first subframes in each series of say 100 subframes in the case

of CCD (see Figure 3.11 for an example). Another possibility to reduce delay could be to interleave normal sub-frames and ABSs at the beginning of the series of 100 subframes and complete with either of them which is more numerous (see Figure 3.12 for an example).

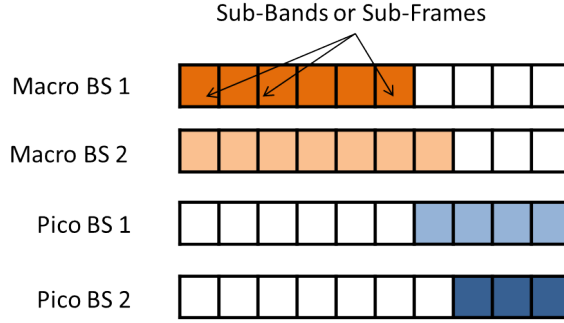


Figure 3.11: Asynchronous ABS/Frequency Splitting Example 1

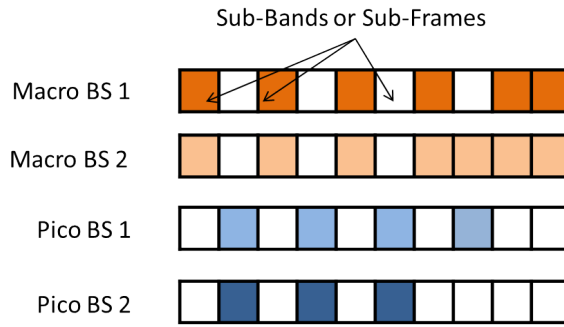


Figure 3.12: Asynchronous ABS/Frequency Splitting Example 2

3.5 Numerical results

We present in this section some numerical results allowing to compare the performances of CCD versus OD in static and dynamic settings with the self-optimizing algorithms presented in the previous sections. We focus on the case $\alpha = 1$ corresponding to the PF utility of user throughputs and we do not take backhaul limitations into consideration.

3.5.1 Deployment scenario

We evaluate the performance of the algorithms in a simple yet insightful scenario. We consider a trisector BS surrounded by 6 interfering trisector macro sites. The simulation is performed only on the central macro site (three sectors colored in blue in Figure 3.13). The surrounding macro BSs (colored in white in Figure 3.13) serve only for generating the interference. In each sector of the central macro BS, 4 small cells are deployed close to the cell edge as shown in Figure 3.13.

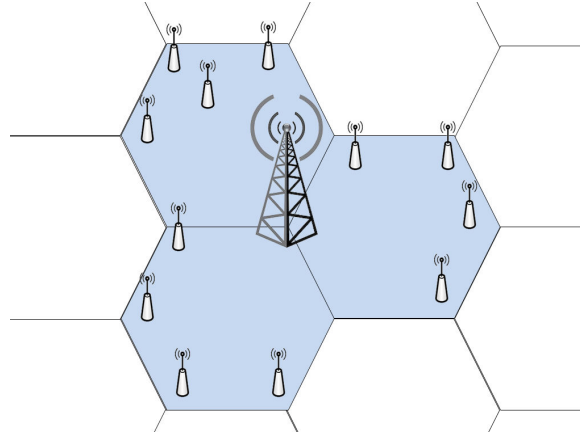


Figure 3.13: Network layout scenario

Placing the small cells close to the macro cell edge is expected to provide the best gains from densification since those are the areas where the macro users would have a reduced SINR that can be improved by the small cells. Also the offloaded macro users to the small cells will suffer less interference from the macro BSs at the cell edge. However, all the algorithms described in this paper do not rely on specific positions of the small cells. So they remain valid for small cells placed near the macro BS because of the presence of a traffic hotspot for example.

We use the propagation models for macro - and small cells (following [1, Page 61]) presented in Table 3.2 which also summarizes all the main simulation parameters.

3.5.2 Static Analysis - Full Buffer performance

We evaluate first the performance of our algorithms for a full buffer scenario, i.e. the number and positions of the users is constant throughout the simulation. The full buffer results serve to prove the optimality of the proposed algorithms for a static scenario especially that the lower bound utilities are less optimal than the exact ones. Indeed, this is no longer obvious in a dynamic scenario as discussed in Section 3.5.3. The traffic configuration is shown in Figure 3.14. The LB here simply consists in increasing the CIOs of the small cells to 12dB. Before applying the LB, each small cell serves two users while each macro cell serve 18 users. But with LB, each small cell now serves 4 users while the macro cells serve each 10 users. The performance results for the following six cases are presented:

- 1) NoSON: this is the baseline case where no mechanism is implemented and the small cells are deployed in CCD.
- 2) LBoNly: the load balancing alone is implemented on a CCD network i.e. all small cells have a CIO of 12dB.
- 3) LB-CCD: Algorithm 3 is implemented with $\alpha = 1$ along with LB.
- 4) LB-CCD-approx: LB is implemented and the ABSr is updated with Equation (3.31).

Table 3.2: Network and Traffic characteristics

Network parameters	
Number of macro BSs	3
Number of small BSs	12
Number of interfering macros	6×3 sectors
Macro Cell layout	hexagonal trisector
Small Cell layout	omni
Intersite distance	500 m
Bandwidth	10MHz
Scheduling	Round-Robin
Channel characteristics	
Thermal noise	-174 dBm/Hz
Macro Path loss (d in km)	$128.1 + 37.6 \log_{10}(d)$ dB
Small cell Path loss (d in km)	$140.7 + 36.7 \log_{10}(d)$ dB
Algorithms Parameters	
Maximum CIO (CIO_max)	12dB
SON update frequency	every event (arrival or departure)
Step size of ABSr optimization algorithm	10^{-4}
Step size of Frequency Splitting optimization algorithm	5.10^{-5}

- 5) *LB-OD*: OD is implemented with LB and the split factor is optimized using Algorithm 4.
- 6) *LB-OD-approx*: OD and LB are also implemented but the split factor is optimized using Equation (3.56).

Figures 3.15–3.17 present respectively the CET, the MUT and the Geometric Mean user Throughput (GMT) for the 6 cases. Regarding the GMT (Figure 3.17) which is the utility that is optimized here ($\alpha = 1$), the optimized resource allocation algorithms perform much better than the baseline (NoSON) or the implementation of only load balancing (LBonly). CCD gives a slightly higher gain than OD (see for example *LB-CCD* vs *LB-OD* in Figure 3.17). This is explained by the fact that in CCD more resources are available since small cells can transmit all the time (outside ABS too) while in OD small cells are only allowed to transmit on their dedicated band. The use of the lower bound on the PF utility in the CCD case degrades the GMT only a bit, so its reduced complexity makes it more attractive (the algorithm only needs the number of users in each cell). The results also show that Algorithms 4 and 5 provide the same performances, so because of its reduced complexity, Algorithm 5 is preferred.

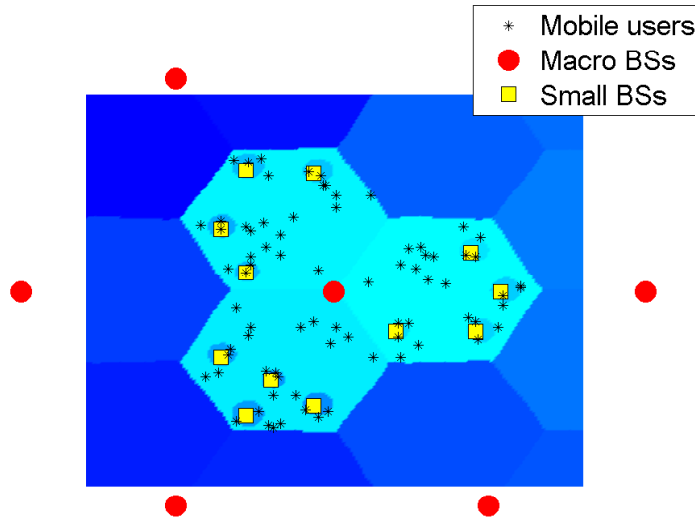


Figure 3.14: Traffic scenario for Full Buffer simulation

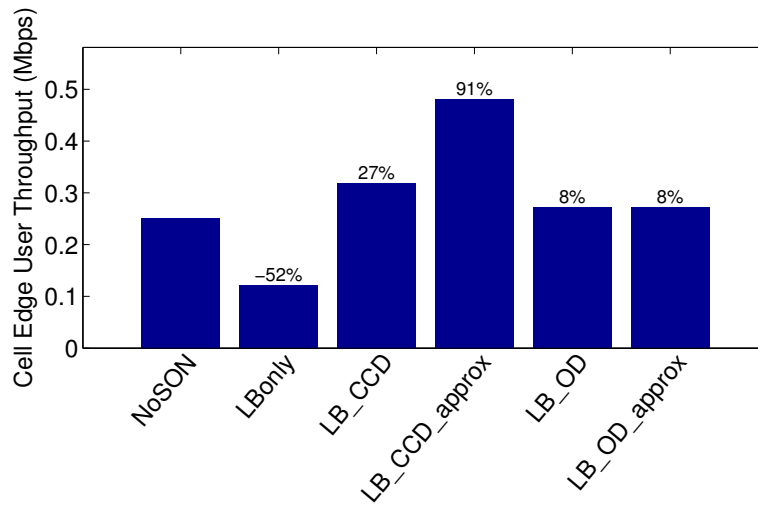


Figure 3.15: Full buffer Cell Edge Throughputs

The results in this section show that the algorithm 4 provides no additional performance gains compared to algorithm 5, so in the following we will restrict ourselves to Algorithm 5 in the OD case. Also, we have seen that the use of the lower bound of the utility in CCD case does not degrade much overall performance (MUT in Figure 3.16), so we will restrict ourselves also to it for the CCD case since its algorithm is much simpler (use of number of users only).

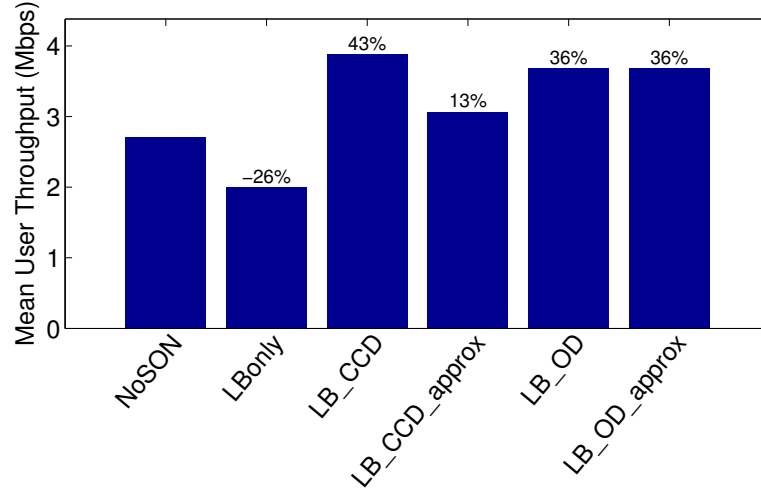


Figure 3.16: Full buffer Mean User Throughputs

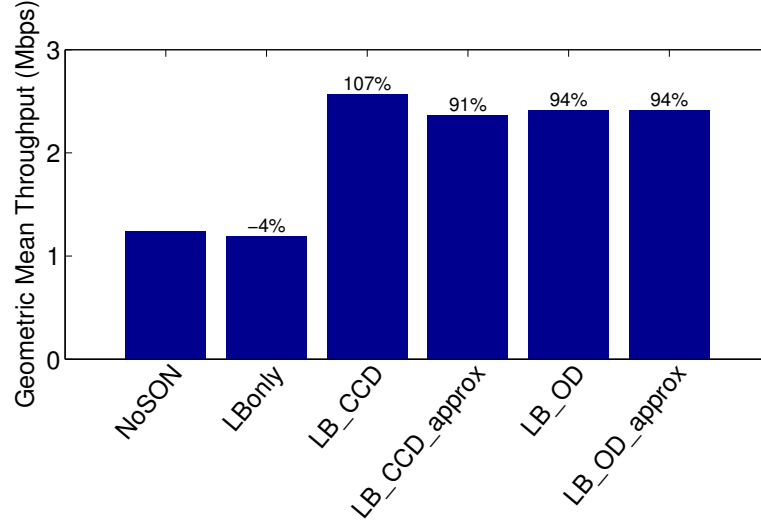


Figure 3.17: Full buffer Geometric Mean Throughputs

3.5.3 Dynamic analysis

In this section, we compare the performance of CCD and OD in a dynamic environment. We consider elastic traffic where users arrive in the network according to a Poisson process with arrival rate λ , download a file of exponential size with mean $\mathbb{E}[\sigma]$ and leave the network as soon as their download is complete. On top of the uniform traffic distribution, we superimpose a traffic around the small cells in order to simulate traffic hotspots with an arrival rate of λ_h . The simulation and traffic parameters are summarized in Table 3.3.

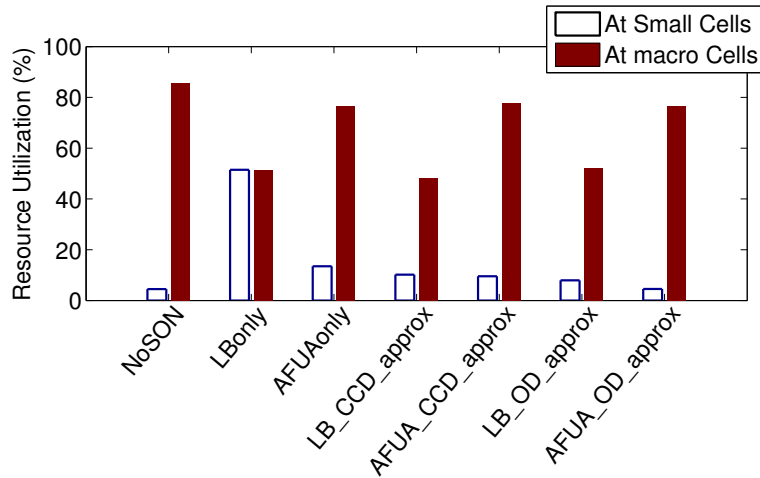
We compare the performance of CCD with ABSrO carried out with (3.31) and OD with SfO performed with (3.56). Both are implemented with the two UA schemes presented in Section 3.2.1 namely LB and AFUA.

Table 3.3: Traffic characteristics

Traffic spatial distribution	uniform
λ	14 users/s/km ²
λ_h	6 users/s/km ²
Service type	FTP
Average file size	6 Mbits

We plot the maximum loads (defined as the percentage of time/frequency resources used) for each layer of cells (macro, pico), the CETs, the MUTs, the FTTs and the GMTs obtained for the following cases:

- 1) NoSON: same as in the full buffer case but with dynamic traffic.
- 2) LBoonly: the load balancing algorithm (3.8) alone is implemented on a CCD network.
- 3) AFUAonly: AFUA using Equation (3.2) is implemented with no eICIC.
- 4) *LB-CCD-approx*: LB algorithm (3.8) is implemented and the ABSr is updated using Equation (3.31).
- 5) *AFUA-CCD-approx*: The ABSr is set using (3.31) and AFUA is implemented.
- 6) *LB-OD-approx*: OD and LB with algorithm (3.8) are implemented and the split factor is optimized using Equation (3.56).
- 7) *AFUA-OD-approx*: OD is implemented with AFUA and the split factor is optimized using (3.56).

**Figure 3.18:** Comparison of the maximum loads among the macro - and small cells

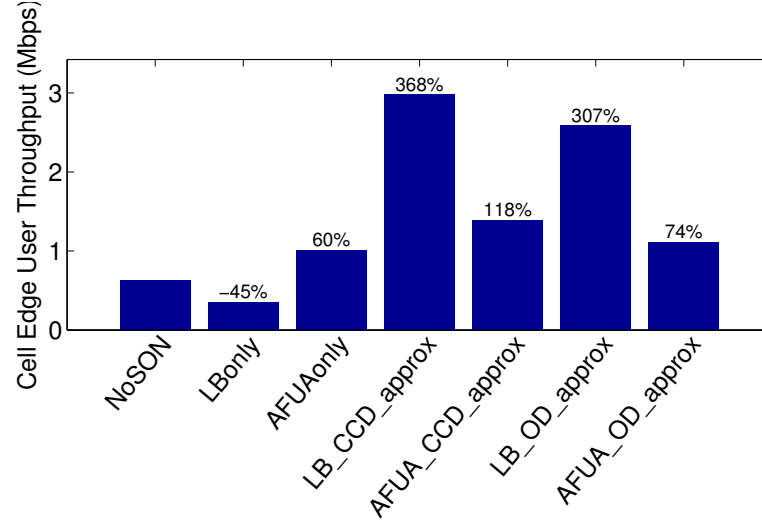


Figure 3.19: Cell Edge Throughputs with elastic traffic

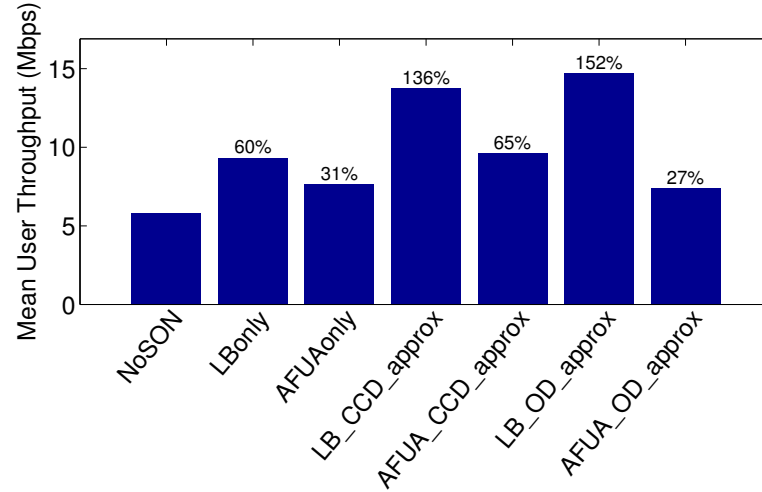


Figure 3.20: Mean User Throughputs with elastic traffic

The performance results show that balancing the loads between macro cells and small cells (see LBonly in Figure 3.18) allows in this scenario to increase the MUT at the expense of the CET and the GMT (see NoSON case vs LBonly case in Figures 3.19, 3.20 and 3.22). This is mainly due to the reduced capacity (worse SINR distribution due to coverage extension) of the pico cells which are offloading the macro cells. So the offloaded users will have a worse user experience because of the decreased SINR they get from low power nodes. At the same time, the remaining users at the macro cells see their throughput increase because of the decreased load of the macro cells (see NoSON and LBonly cases in Figure 3.18). But using AFUA, the GMT, MUT and CET are improved (see AFUAonly in Figures 3.22, 3.19 and 3.20) even if the macro cells are only slightly offloaded (see AFUAonly in Figure 3.18). Since AFUA targets directly the improvement of user Quality of Service (QoS), it allows to adjust the offloading so that the small cell users' performance

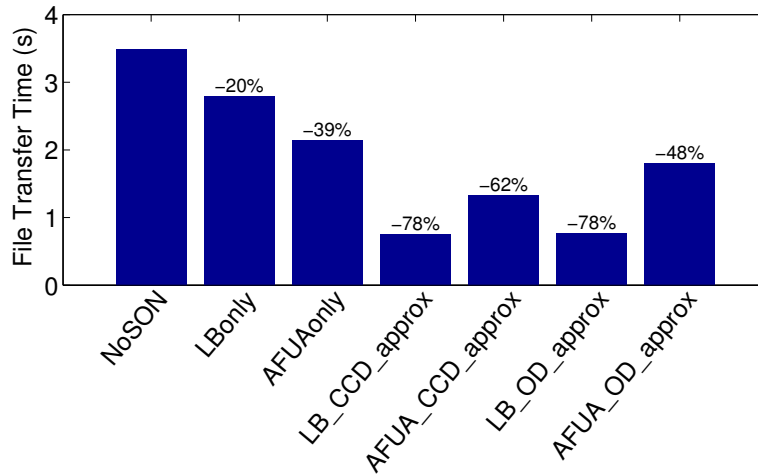


Figure 3.21: File Transfer Times with elastic traffic

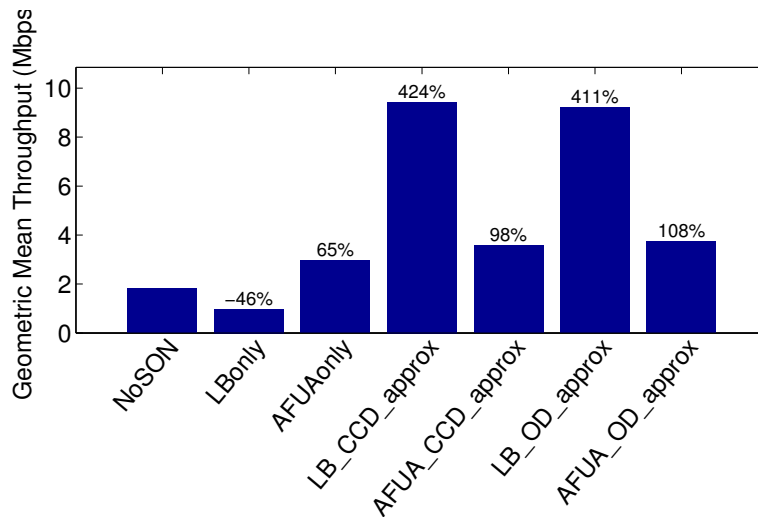


Figure 3.22: Geometric Mean Throughputs with elastic traffic

is not degraded. LBonly shows the need for an interference mitigation technique.

Indeed, adding interference mitigation schemes to LB allows to increase both MUT and CET (see *LB-CCD-approx* and *LB-OD-approx* in Figures 3.19 and 3.20), thus the GMT is also better (see Figure 3.22). The same holds for AFUA with interference mitigation as shown by *AFUA-CCD-approx* and *AFUA-OD-approx* in Figures 3.19, 3.20 and 3.22. We can also see that the loads are not balanced anymore between macro cells and small cells. This falls in line with the fact that macro cells and small cells do not have the same capacity (in terms of SINR distribution) if their loads are balanced using coverage extension. This load disparity allows to get an increased GMT (see Figure 3.22) which means proportional fairness of throughput among all users. These results show the importance of having an interference mitigation scheme with the deployment of small cells in addition to LB which is also necessary to increase the coverage of the low power nodes.

CCD provides better KPIs than OD as shown in Figures 3.19, 3.20, 3.21 and 3.22. This was

predictable and has already been observed in the full buffer case. The explanation is the same as in the full buffer case meaning that CCD provides small cells with more time/frequency resources than OD.

In general, from the CET, the FTT and the GMT of AFUAonly compared to LBoonly in Figures 3.19, 3.21 and 3.22, we can say that AFUA improves performance better than LB. This is mainly due to its α -fairness while LB balances the loads regardless of the SINR distribution of each cell. But this advantage disappears when the UA scheme is coupled with an eICIC scheme as shown by the performance of *LB-CCD-approx* and *LB-OD-approx* compared to that of *AFUA-CCD-approx* and *AFUA-OD-approx* in Figures 3.21 and 3.22. It is noted that much higher gains could have been obtained for AFUA scheme if there was no constraint on the allowed coverage zone of each cell (CIO_max) which strongly limits the approach.

We now evaluate the tightness of the lower bound used to optimize the ABSr in the CCD case for a dynamic traffic. We plot in Figure 3.23 the users throughputs c.d.fs when the ABSr optimizes the exact utility function using Algorithm 3 versus when it optimizes the lower bound utility using (3.31). The results are presented for both LB and AFUA cases.

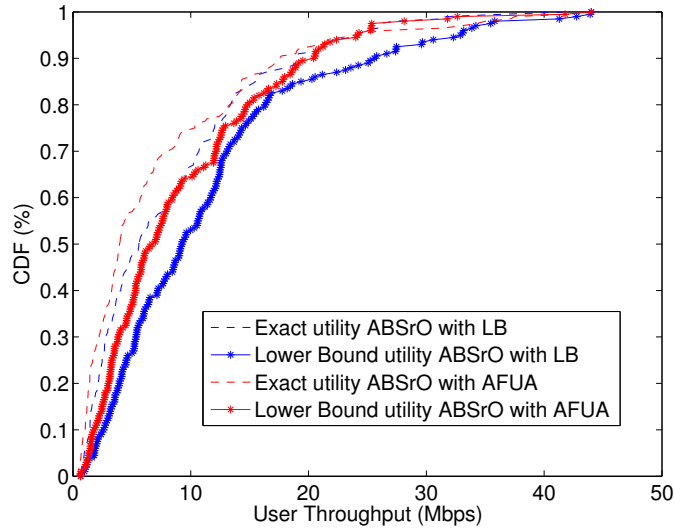


Figure 3.23: Users Throughputs CDF: Exact PF utility (3.25) vs Lower bound PF utility (3.27)

The results are quite surprising since they show that the lower bound utility performs better than the exact utility. This is mainly due to the impact of the optimization algorithm on the trajectory in the traffic configuration states (number of users, their positions, etc). Basically the choice of the ABSr, which is derived from the choice of the utility function, impacts on the file transfer times of the users. So they stay longer or shorter depending on that impact. In the end, the utility chosen can degrade or improve depending on whether the system goes through states where there are less users to serve. Because of this, an appropriate choice of utility function is needed.

The choice we have made in this paper allows to get very simple algorithms that require only the number of users in the system. If say the sum of the file transfer times was used, the geometric

mean throughput could have been improved, but that would require bookkeeping of all users throughputs, whose measurements are highly fluctuating and imprecise in practice. It is also noted that the optimality of the algorithms proposed is guaranteed between two consecutive events (arrivals or departures) when the system can be considered static so the same results as in the full buffer case (Section 3.5.2) can be obtained. For the dynamic case, the algorithms proposed will still perform optimally if the time scale of their updates is smaller than that of the traffic dynamics.

3.6 AUTOSDN framework

In order to implement the SON concept in mobile networks, the SDN paradigm can be used. It consists in providing an abstraction layer to the network elements and introduce programmable SON functions that make use of the abstraction layer to self-organize the network. This way, any third party can provide SON algorithms that will be implemented in the network in a vendor-agnostic way. The Autonomic SDN (AutoSDN) project that proposes such a framework was performed in collaboration with the University of Pireaus (Greece) and Orange Labs (France).

The objective was to propose an architecture and a proof of concept for a programmable framework for SON, denoted here as Software-Defined SON or SD-SON. Unlike programmable centralized SON, the aim here is to also allow highly reactive SON that can adapt the network to variations in traffic and propagation. As shown in Figure 3.24, the architecture is composed of the Unified Management Framework (UMF) core proposed in the Univerself project [78] and a SDN controller.

The UMF core provides a unified view of the network that allows to control its governance, coordination and manage its knowledge. The UMF core gives the operator the tools for managing the SON functions that are implemented as UMF Network Empowerment Mechanisms (NEMs). It allows for example to instantiate / delete / monitor / start / stop NEMs, assign them to managed elements, apply policies or even set their pace of execution.

The SDN controller provides a northbound interface to SON algorithms providers that allow SON algorithms to discover existing network elements and topology, retrieve and configure network elements parameters (programmability) and monitor KPIs or subscribe to events related to KPI value changes. These functionalities are possible with the SDN controller's southbound interface which is an Application Programming Interface (API) through which network elements, their metrics and parameters can be discovered and controlled. It is noted that the API is designed to support the various vendors' network elements.

A complete description of the AutoSDN framework can be found in [59] and [77]. Figure 3.25 presents the network view of the UMF dashboard. From this interface, SON algorithms can be deployed on specific nodes of a live network. We present here an instance of the AutoSDN framework as a proof-of-concept.

As shown in the UMF dashboard, the network instance is composed of a trisector macro site surrounded by 6 interfering macro sites. In each sector of the central macro site, 4 small cells

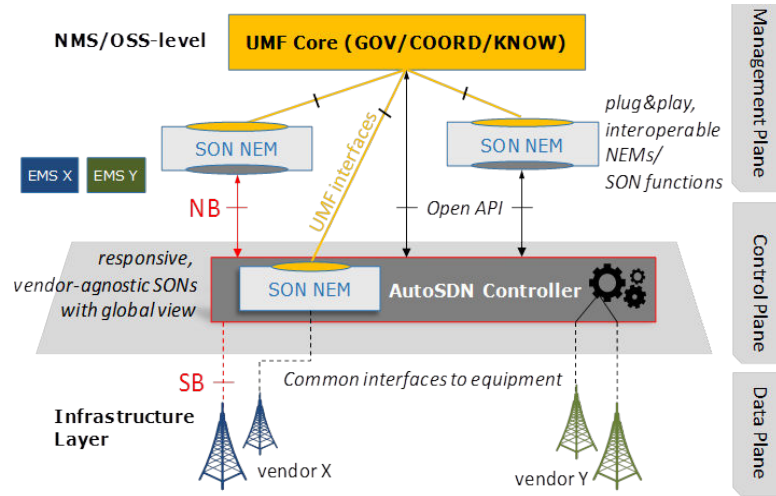


Figure 3.24: AutoSDN overall architecture.

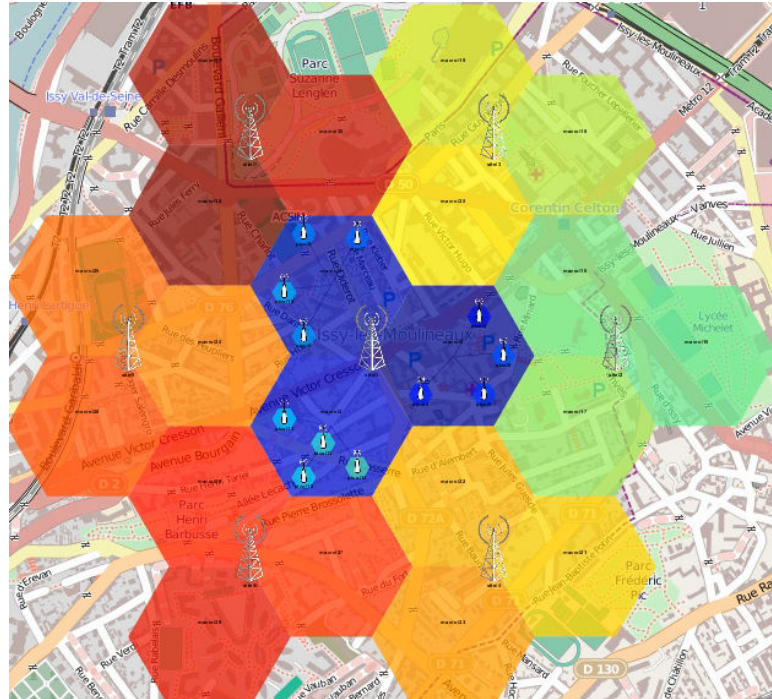


Figure 3.25: UMF dashboard view of the network

are deployed. We consider two SON algorithms in this proof-of-concept: the load balancing algorithm presented in Equation (3.8) and the ABSrO algorithm presented in Equation (3.39). Both algorithms are deployed in the central sectors. Elastic traffic is used in this case with Poisson arrivals.

Figure 3.26 presents the coverage maps (best server area) of the different cells before and after deploying the SON functions. Figure 3.27 presents a view of the network parameters (CIOs, mute ratio - ABSr) and KPIs per cell (loads) or aggregated (mean number of active users in the

system). This view is available through the SDN controller.

As shown in Figure 3.26, the small cells increase their coverage area in order to offload the macro cells. This coverage increase is performed via an increase of the CIOs as can be seen in Figure 3.27. As a consequence the load of the macro cell is decreased and that of the small cells is increased (see Figure 3.27). The ABSr of the macro cell is subsequently also increased (in Figure 3.27) in order to reduce the interference of the macro cells on the offloaded users. The joint modification of the CIOs and the ABSr allows to improve the overall efficiency of the system as demonstrated by the decrease in the mean number of active users in the system (see Figure 3.27).

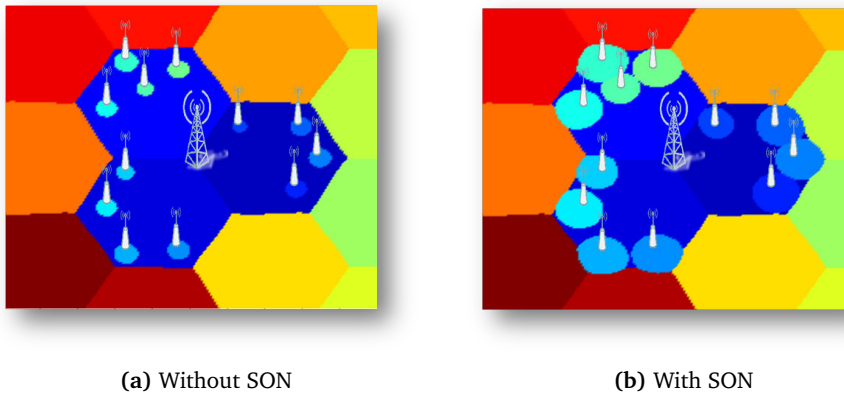


Figure 3.26: Coverage maps

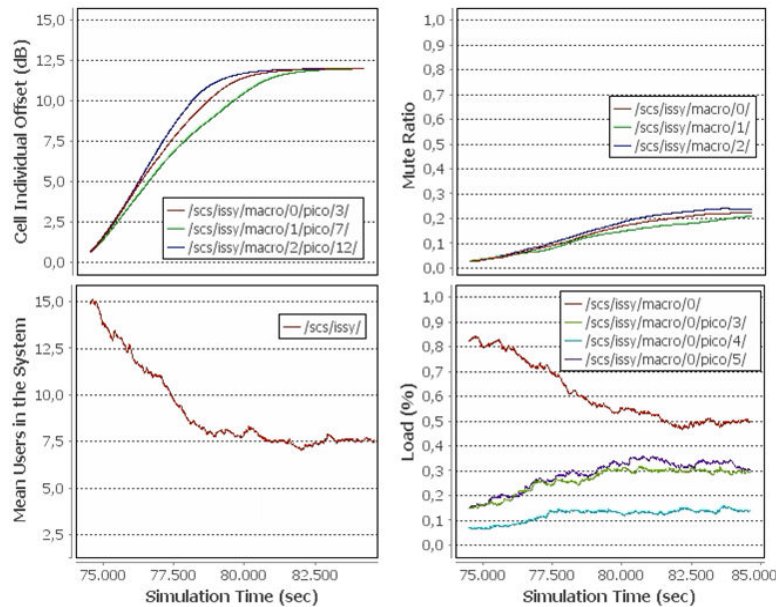


Figure 3.27: Parameters and KPIs view from the SDN controller

3.7 Conclusion

We have presented in this chapter two types of self-optimizing algorithms for HetNets. The first one are for the user association in a HetNet where the goal is to balance the loads between macro cells and small cells. To achieve this goal, we presented simple algorithms that use either the CIOs of the small cells (Equations (3.8) and (3.13)) or the ABSr of the macro cells (Equation (3.39)). For the load balancing algorithms, modifications to the load definitions were proposed in order to take into account possible backhaul limitations. The second type of algorithms aim at reducing the interference caused by the high power nodes (macro cells) on the low power nodes (small cells). Two approaches were studied in this case: time domain interference mitigation using the ABS mechanism, and frequency domain mitigation with OD. In both approaches, the algorithms proposed (Algorithms 3, 4) optimize the general family of α -fair utilities of users' throughputs. In order to further simplify the algorithms, lower bound for utilities were proposed as optimization objectives and the corresponding algorithms were provided (Algorithm 5 and Equation (3.31)). Extensive numerical results were provided to support the usefulness of the proposed algorithms and give insights into their impact on network performance. The results showed for example that load balancing must be used in conjunction with interference coordination in order to improve the performance of all users (cell edge and cell centre users). Considerable performance gains were also observed, e.g. up to 368% of gain in CET with LB and ABSrO in the case of CCD. The broad range of proposed algorithms (for the whole class of α -fair utilities) allow the operator to choose its performance objective according to its own policy. Finally we presented the AutoSDN framework which provides a flexible way to implement the proposed algorithms in the real network giving enhanced control to the operator.

Small cells provide a cheap way to densify the network but still require additional installations of BSs. The next chapter investigates solutions which also provide network densification (creation of new cells) without further deployments by using AAS.

Chapter 4

Self-optimization for active antenna systems

“Observation is a passive science, experimentation an active science.”

– Claude Bernard

4.1 Introduction

The architecture of BSs especially the location of their different components (Baseband Unit (BBU), Radio Frequency (RF)¹ components and antenna) has greatly evolved throughout the mobile network generations (see Figure 4.1). The traditional BS had a completely passive antenna which is fed by the BBU and the RF unit through a transmission line such as a coaxial cable. In this case the antennas had to be very close (typically several meters) to the processing units because of the huge losses in the cable. When fiber cables became available, it became possible to place the antenna and the RF unit at a greater distance from the BBU. In this case, the RF unit now denoted as Remote Radio Head (RRH) is responsible for transmit/receive signal processing and sends the RF signal through a short coaxial cable to the antenna which is nearby. The next evolution to AASs integrates the RF unit directly into the antenna completely avoiding the losses in the cable. The AASs contain a transceiver per antenna element allowing to reduce the size of RF components such as the amplifier, also to introduce redundancy in case any antenna elements fails.

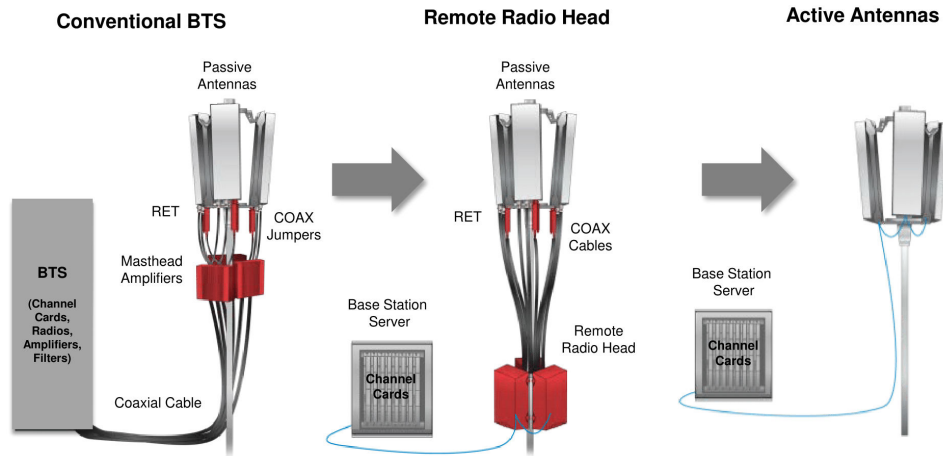


Figure 4.1: BS architecture evolution

Source: CommScope White paper [49].

AASs allow to dynamically change the beam pattern at the antenna by controlling each antenna element's phase and amplitude. This feature of AASs has opened new possibilities for cell planning. The first possibility has been the VeSn which allows to create vertically separated cells: the original sector denoted outer sector and an inner sector that is created near the BS using an electrically downtilted antenna. Since VeSn is enabled with a constant power budget, the question is raised as to whether or not it is useful to be activated depending on the traffic distribution.

VeSn is possible using a vertical array of antenna elements. When a matrix of antenna elements is available, a small sector can be generated anywhere in the original cell. The term Virtual Sector (ViS) is then used. As in the case of VeSn, the activation of ViS must take into account the traffic distribution. However since the ViS can be focused towards a traffic hotspot, it can be

¹RF: the frequency range used in radio communications

activated more often than VeSn.

VeSn and ViSn are long time scale solutions since the parameters defining the inner or virtual sector are generally fixed for a long time (typically in the order of days or even weeks), only the activation of the cells is dynamic. In the next evolution of AASs, more focused beams can be created thus the antenna beam can be geared towards each specific UE when it is scheduled, this is referred to as beamforming.

For each of the three applications of AASs, we propose SON algorithms for their optimal operation. For VeSn, we provide in Section 4.2 optimal activation algorithms as well as optimal frequency splitting in the case of orthogonal frequency operation of the inner and outer sectors. In this case an activation rule is provided to switch to full frequency reuse when needed. The optimal frequency splitting algorithms is also provided for ViSn in Section 4.3 along with numerical results giving insights on full reuse activation. Finally in Section 4.4, we present a multi-level beamforming strategy along with performance results that demonstrate the tremendous gains of this technology in terms of user performance and energy saving.

4.2 Vertical sectorization

4.2.1 Problem formulation

A vertically sectorized cell is split into an outer cell whose antenna is vertically downtilted by θ_o and an inner cell with vertical tilt θ_i . $\theta_i > \theta_o$, so that the inner focuses its beam towards the center of the cell as shown in Figure 4.2. The tilts are generally electrically controlled, allowing to set them dynamically as we activate/deactivate VeSn.

The inner and outer cells share the total transmit power available for the sector, but the available bandwidth is fully reused. Performance gains from VeSn are brought by the fact that we can transmit to two users simultaneously with the whole bandwidth. So VeSn has better gains compared to no VeSn when the loads are high both in the inner and the outer sectors.

The choice of the antenna tilts and transmit powers for the inner/outer cells is very critical to the performance of VeSn. They both have an impact on the additional interference brought about by VeSn, particularly on neighbouring cells.

Algorithms trying to get the best out of VeSn should optimize antenna tilts (θ_i, θ_o), proportion of total power allocated to inner sector (P_i) and a decision rule as to when to activate the inner (VeSn activation controller). We mainly focus here on the activation problem.

Let us consider elastic traffic in which users arrive in a cell according to a Poisson process of arrival rate λ users/s uniformly distributed in space, download a file of mean size $\mathbb{E}[\sigma]$ and leave the network as soon as their download finishes. The load of such cell has been derived in [12] as

$$\rho = \int_A \frac{\lambda(r)\mathbb{E}[\sigma]}{R(r)} dr = \frac{\lambda\mathbb{E}[\sigma]}{|A|} \int_A \frac{dr}{R(r)} \quad (4.1)$$

where A is the area of the cell, $R(r)$ - the peak data rate at position r and $\lambda(r)$ - the arrival rate

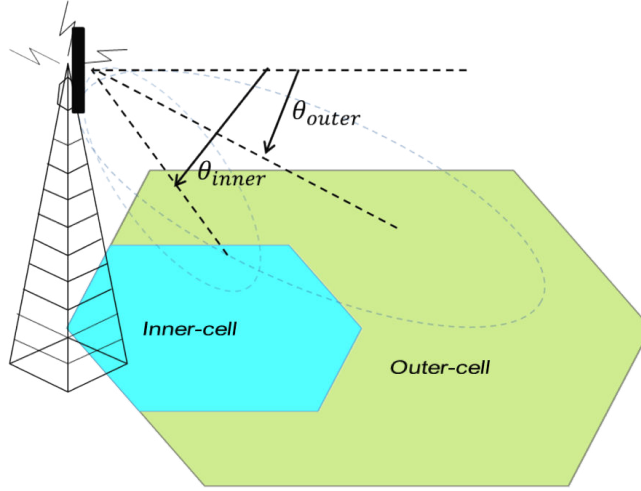


Figure 4.2: Vertical Sectorization illustrated

density which is also equal to $\frac{\lambda}{|A|}$ because the traffic is assumed to be spatially uniform. A user arriving at position r will have a flow throughput of $R(r)(1 - \rho)$. So the MUT will be

$$\mu = \int_A \frac{\lambda(r)}{\lambda} (1 - \rho) R(r) dr = \frac{1}{|A|} \int_A (1 - \rho) R(r) dr. \quad (4.2)$$

We will now calculate this load for two configurations:

- **Vertical Sectorization OFF (VSOFF):** we have only one sector where the vertical tilt is θ_o and transmit power is 46 dBm.
- **Vertical Sectorization ON (VSON):** we have two vertical sectors with the inner cell having vertical tilt of θ_i and transmit power of 43 dBm while the outer cell has vertical tilt of θ_o and transmit power of 43 dBm. Each vertical sector is a base station of its own so we apply a full frequency reuse.

The two cases above differ from each other by their SINR distribution and bandwidth reuse. We assume uniformly distributed traffic in the inner and outer parts of the cell respectively. We denote the total arrival rates by λ_i and λ_o for the inner and outer cells respectively as shown in Figure 4.2. The load of the sector in VSOFF can be written as

$$\begin{aligned} \rho^{\text{OFF}} &= \int_A \frac{\lambda(r) \mathbb{E}[\sigma]}{R(r)} dr \\ &= \frac{\lambda_i \mathbb{E}[\sigma]}{|A_i|} \int_{A_i} \frac{dr}{R(r)} + \frac{\lambda_o \mathbb{E}[\sigma]}{|A_o|} \int_{A_o} \frac{dr}{R(r)} \\ &= \rho_i^{\text{OFF}} + \rho_o^{\text{OFF}} \end{aligned}$$

The load of the inner and outer sectors in the VSON case are

$$\rho_i^{\text{ON}} = \frac{\lambda_i \mathbb{E}[\sigma]}{|A_i|} \int_{A_i} \frac{dr}{R_i(r)} \quad (4.3)$$

and

$$\rho_o^{\text{ON}} = \frac{\lambda_o \mathbb{E}[\sigma]}{|A_o|} \int_{A_o} \frac{dr}{R_o(r)} \quad (4.4)$$

The corresponding MUTs can be derived as

$$\begin{aligned} \mu^{\text{OFF}} &= \int_A \frac{\lambda(r)}{\lambda} (1 - \rho^{\text{OFF}}) R(r) dr \\ &= \int_{A_i} \frac{\lambda_i}{|A_i|(\lambda_i + \lambda_o)} (1 - \rho^{\text{OFF}}) R(r) dr + \int_{A_o} \frac{\lambda_o}{|A_o|(\lambda_i + \lambda_o)} (1 - \rho^{\text{OFF}}) R(r) dr \\ &= (1 - \rho_i^{\text{OFF}} - \rho_o^{\text{OFF}}) \left(\frac{\lambda_i}{\lambda_i + \lambda_o} \frac{1}{|A_i|} \int_{A_i} R(r) dr + \frac{\lambda_o}{\lambda_i + \lambda_o} \frac{1}{|A_o|} \int_{A_o} R(r) dr \right) \end{aligned} \quad (4.5)$$

$$\begin{aligned} \mu^{\text{ON}} &= \int_{A_i} \frac{\lambda(r)}{\lambda_i + \lambda_o} (1 - \rho_i^{\text{ON}}) R_i(r) dr + \int_{A_o} \frac{\lambda(r)}{\lambda_i + \lambda_o} (1 - \rho_o^{\text{ON}}) R_o(r) dr \\ &= (1 - \rho_i^{\text{ON}}) \frac{\lambda_i}{\lambda_i + \lambda_o} \frac{1}{|A_i|} \int_{A_i} R_i(r) dr + (1 - \rho_o^{\text{ON}}) \frac{\lambda_o}{\lambda_i + \lambda_o} \frac{1}{|A_o|} \int_{A_o} R_o(r) dr \end{aligned} \quad (4.6)$$

In order to further clarify the formulas, let us denote by:

$$\begin{aligned} a_i &= \frac{1}{|A_i|} \int_{A_i} R_i(r) dr & a_o &= \frac{1}{|A_o|} \int_{A_o} R_o(r) dr \\ b_i &= \frac{\mathbb{E}[\sigma]}{|A_i|} \int_{A_i} \frac{dr}{R_i(r)} & b_o &= \frac{\mathbb{E}[\sigma]}{|A_o|} \int_{A_o} \frac{dr}{R_o(r)} \\ \alpha_i &= \frac{1}{|A_i|} \int_{A_i} R(r) dr & \alpha_o &= \frac{1}{|A_o|} \int_{A_o} R(r) dr \\ \beta_i &= \frac{\mathbb{E}[\sigma]}{|A_i|} \int_{A_i} \frac{dr}{R(r)} & \beta_o &= \frac{\mathbb{E}[\sigma]}{|A_o|} \int_{A_o} \frac{dr}{R(r)} \end{aligned}$$

a_i and a_o represent the average peak data rates in the inner and outer cells respectively when VeSn is activated, α_i and α_o represent their counterparts when VeSn is deactivated. b_i , b_o , β_i and β_o similarly represent the ratio between the mean file size and the inner and outer cells' capacities with or without VeSn.

Now we can rewrite μ^{ON} and μ^{OFF} in terms of λ_i and λ_o as follows

$$\mu^{\text{OFF}} = \frac{1}{\lambda_i + \lambda_o} \left[-\alpha_i \beta_i \lambda_i^2 - (\alpha_i \beta_o + \alpha_o \beta_i) \lambda_i \lambda_o - \alpha_o \beta_o \lambda_o^2 + \alpha_i \lambda_i + \alpha_o \lambda_o \right] \quad (4.7)$$

$$\mu^{\text{ON}} = \frac{1}{\lambda_i + \lambda_o} \left[-a_i b_i \lambda_i^2 - a_o b_o \lambda_o^2 + a_i \lambda_i + a_o \lambda_o \right] \quad (4.8)$$

Case of non-uniform spatial distribution of the traffic: When the traffic is not distributed uniformly in the cell, some corrections are needed of the formulas. In particular the integrations over the area of the cells are modified by replacing dr with $p(r)dr$ where $p(r)$ is related to the density of traffic at position r .

$$p(r) = \begin{cases} \frac{\lambda(r)|A_i|}{\lambda_i} & \text{if } r \in \text{inner} \\ \frac{\lambda(r)|A_o|}{\lambda_o} & \text{if } r \in \text{outer} \end{cases} \quad (4.9)$$

For example, a_i will be rewritten as

$$a_i = \frac{1}{|A_i|} \int_{A_i} R_i(r) p(r) dr.$$

4.2.2 Analytical approach to VeSn activation and calibration with realistic measurements

We proceed to design a controller which activates or deactivates the inner cell based on inner/outer loads, with the objective of always maximizing the MUT. We choose to restrain the domain of decision within the stability conditions, i.e. the set of traffic demand values for which $\rho_i < 1$ and $\rho_o < 1$.

The decision for VSON or VSOFF is performed by comparing the MUT as a function of (λ_i, λ_o) for the two cases. More formally

$$\text{Action} = \begin{cases} \mathbf{VSON} & \text{if } \mu^{\text{ON}}(\lambda_i, \lambda_o) > \mu^{\text{OFF}}(\lambda_i, \lambda_o) \\ \mathbf{VSOFF} & \text{otherwise.} \end{cases} \quad (4.10)$$

In other terms, we activate VeSn only when the MUT can be improved this way. It appears from this definition that the decision boundary will be the equation

$$\mu^{\text{ON}}(\lambda_i, \lambda_o) = \mu^{\text{OFF}}(\lambda_i, \lambda_o) \quad (4.11)$$

Using the relationship between the loads and the traffic demand, we can reformulate this

equation for both **VSOFF** and **VSON** cases. So we have

$$\lambda_i = \frac{\rho_i^{\text{ON}}}{b_i} = \frac{\rho_i^{\text{OFF}}}{\beta_i} \quad (4.12)$$

$$\lambda_o = \frac{\rho_o^{\text{ON}}}{b_o} = \frac{\rho_o^{\text{OFF}}}{\beta_o} \quad (4.13)$$

Replacing (4.12) and (4.13) in (4.11), we get the following two equations

$$(\text{VSOFF}) : \mu^{\text{ON}} \left(\frac{\rho_i^{\text{OFF}}}{\beta_i}, \frac{\rho_o^{\text{OFF}}}{\beta_o} \right) = \mu^{\text{OFF}} \left(\frac{\rho_i^{\text{OFF}}}{\beta_i}, \frac{\rho_o^{\text{OFF}}}{\beta_o} \right) \quad (4.14)$$

$$(\text{VSON}) : \mu^{\text{ON}} \left(\frac{\rho_i^{\text{ON}}}{b_i}, \frac{\rho_o^{\text{ON}}}{b_o} \right) = \mu^{\text{OFF}} \left(\frac{\rho_i^{\text{ON}}}{b_i}, \frac{\rho_o^{\text{ON}}}{b_o} \right) \quad (4.15)$$

Now let us rewrite those equations in closed form, in terms of the variables (either the traffic demands or the loads). We get

$$(\text{General}) : A\lambda_i^2 + B\lambda_i\lambda_o + C\lambda_o^2 + D\lambda_i + E\lambda_o = 0 \quad (4.16)$$

$$(\text{VSOFF}) : \frac{A}{\beta_i^2}\rho_i^2 + \frac{B}{\beta_i\beta_o}\rho_i\rho_o + \frac{C}{\beta_o^2}\rho_o^2 + \frac{D}{\beta_i}\rho_i + \frac{E}{\beta_o}\rho_o = 0 \quad (4.17)$$

$$(\text{VSON}) : \frac{A}{b_i^2}\rho_i^2 + \frac{B}{b_ib_o}\rho_i\rho_o + \frac{C}{b_o^2}\rho_o^2 + \frac{D}{b_i}\rho_i + \frac{E}{b_o}\rho_o = 0 \quad (4.18)$$

where

$$A = \alpha_i\beta_i - a_ib_i$$

$$B = \alpha_i\beta_o + \alpha_o\beta_i$$

$$C = \alpha_o\beta_o - a_ob_o$$

$$D = a_i - \alpha_i$$

$$E = a_o - \alpha_o$$

In the $\lambda_o - \lambda_i$ plane, Equation (4.16) defines a parabola which constitutes a decision boundary. This boundary separates the plane into two regions: one in which it is better to turn VeSn on and the other - off. In a similar manner, two decision boundaries can be defined in the inner and outer load plane.

4.2.2.0.1 Discussion on how to evaluate ρ_i and ρ_o in VSOFF case The idea here is to use the proportion of users that would be in the inner cell in order to deduce which of part of the total load would come from the inner cell. In order to classify a user as being in the inner cell, we can use a threshold on the Reference Signal Received Power (RSRP) since in general it tends

to decrease as we get farther from the base station. However this does not work when there is shadowing, so a more sophisticated classifier taking also into account the latency for example could be used.

In the calibration step, we adjust the model using the realistic network simulator described in [75] for scenarios with $\theta_i = 8^\circ$, $\theta_o = 0^\circ$ of electrical tilts to which 4° of mechanical tilt is added and variable loading. We then statistically evaluate the coefficients of Equations (4.17) and (4.18). For different traffic demands, we evaluate the loads and the MUT for VSON/VSOFF cases and plot in the load plane the different activation decisions obtained by choosing the action (VSON, VSOFF) that gives the highest MUT, as shown in Figures 4.3 and 4.4 respectively.

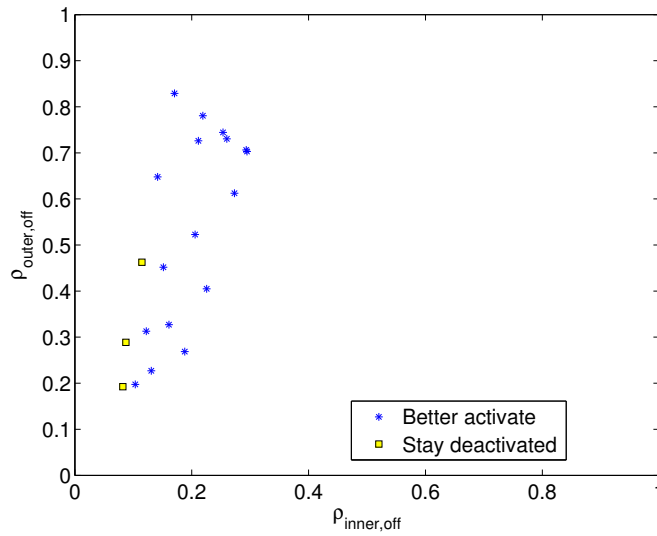


Figure 4.3: Inner activation decision in the inner/outer sector load plane.

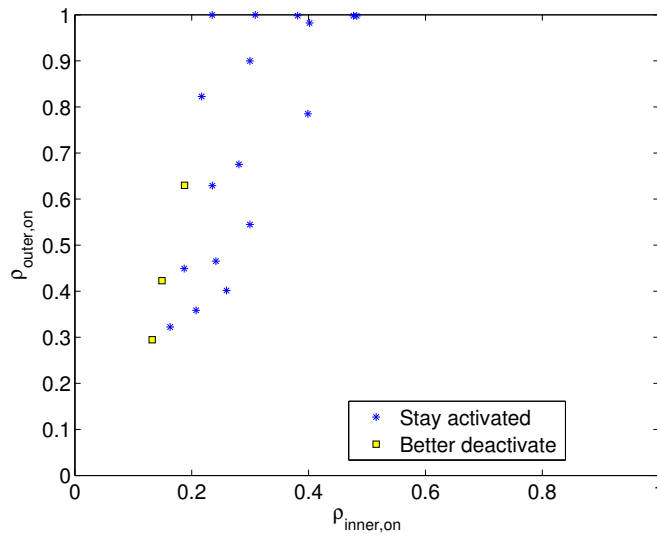


Figure 4.4: Inner deactivation decision in the inner/ outer sector load plane.

In order to find the decision boundaries, we fit a logistic regression with the functional form

obtained in Equations (4.17) and (4.18). The problem is formulated as follows

$$\text{minimize}_{a,b,c,d,e} \sum_n \log(1 + \exp(-y_n(a\rho_i^2 + b\rho_o^2 + c\rho_i\rho_o + d\rho_i + e\rho_o))) \quad (4.19)$$

where n is the index of a point on the graphs, $y_n = 1$ for blue stars in Figures 4.3 and 4.4, $y_n = -1$ for the yellow squares in these Figures. The resulting decision boundaries are shown in Figure 4.5.

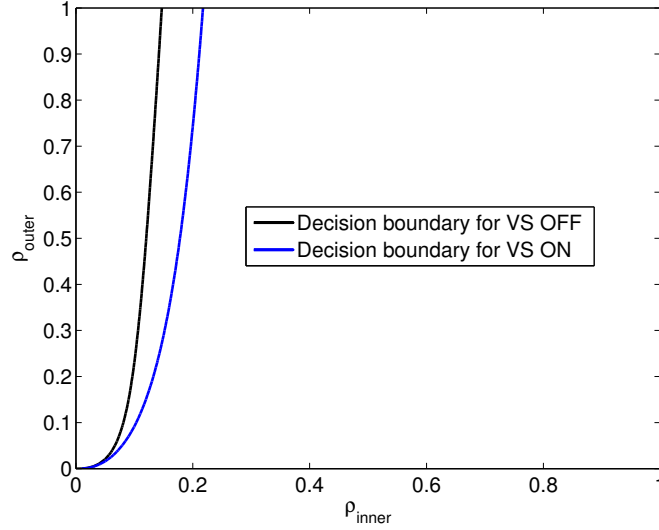


Figure 4.5: VeSn (de)activation decision boundaries in the inner/outer sector load plane.

We evaluate the performance (in terms of MUT) of the VeSn activation controller presented in Figure 4.5 and denoted as AAS SON with a Matlab network simulator (described in Appendix B). The baselines are the fixed activation scenarios in which either we enable VeSn always (VSON), or it is always OFF (VSOFF). The study is done on one BS site only and we suppose that the neighboring sites do not implement VeSn. We simulate a varying traffic distribution with the traffic decreasing smoothly from the inner cell area and increasing in the outer cell area (see Figure 4.6). We plot the activation decisions of each scenario in Figure 4.7. We can see that the VeSn activation controller tracks the variation of the traffic by enabling the VeSn only when there is enough traffic in the inner (blue curve in Figure 4.6). The controller also allows to get the best MUT over time by choosing the activation decision that provides the best MUT (see Figure 4.8). These results also show that when there is no traffic in the inner area, VSOFF performs better than VSON, which highlights the need for the VeSn activation controller.

4.2.3 Frequency splitting approach

The classical implementation of VeSn (as in the previous section) is to fully reuse the bandwidth, so the inner and outer cells interfere with each other. The SINR of a user u served by the inner cell is then

$$S_u = \frac{P^i h_u^i}{N_0 + P^o h_u^o + \sum_{c \neq s} P^c h_u^c} \quad (4.20)$$

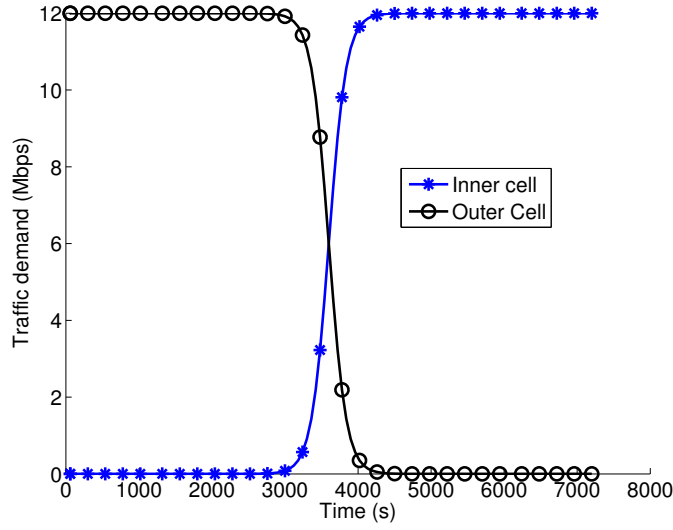


Figure 4.6: Evolution of the traffic distribution in inner/outer cells' areas.

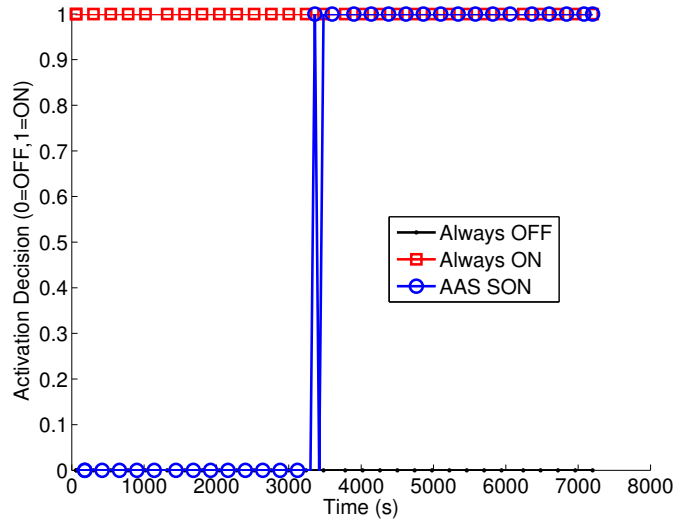


Figure 4.7: VeSn activation Decisions over time.

where P^i, P^o are respectively the transmit powers of inner and outer cells, and h_u^i, h_u^o - the pathlosses from respectively the inner and outer cell antennas to user u . The sum over $c \neq s$ represents the interfering signals from the neighboring BSs. It is noted that a similar expression can be given for a user served by the outer cell. Since the bandwidth is fully reused, the total transmit power has to be shared between inner and outer cells, e.g. using equal split $P^i = P^o = P^s/2$.

The transmit power for the inner and outer cells is reduced compared to the case where VeSn is not implemented. As a consequence, SINR degradation may be observed due to reduced useful signal and increased interference for all the users. A better power split of the transmit power between the inner and outer cells is challenging because of the complex dependencies between the transmit powers and the average data rates of the users.

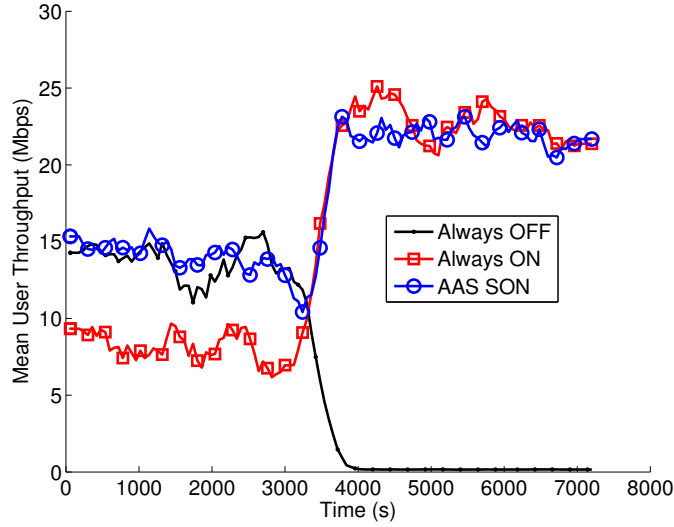


Figure 4.8: Mean User Throughput over time.

It has been shown in the previous section's numerical results (especially in Figure 4.8) that the full bandwidth reuse implementation of VeSn only improves performance over no VeSn when there is enough traffic in the inner cell area. So we proposed a SON controller which activates the VeSn feature only when needed. However, even when the traffic demand in the inner region is low, the users that are served in the inner region can still benefit from the stronger signal transmitted by the downtilted inner antenna.

The VeSn feature implementation considered here shares the total available bandwidth between inner and outer cells (no reuse). In this case, not only the inner and outer cells do not interfere each other, but also the transmit power per Hertz remains unchanged. Indeed, if we denote by $\delta \in [0, 1]$ the fraction of bandwidth allocated to the inner cell, the same fraction of the transmit power is also allocated to the inner cell, so that $P^i = \delta P^s$ and $P^i/W_\delta = (\delta P^s)/W_\delta = P^s/W$ where $W_\delta = \delta W$ is the bandwidth allocated to the inner cell.

The SINR of a user u served by the inner cell in the case of bandwidth sharing can be written as

$$S_u = \frac{P^s h_u^i}{N_0 + \sum_{c \neq s} P^c h_u^c} \quad (4.21)$$

It is noted that this SINR expression is given for the whole bandwidth W but it is also equivalent to the SINR for each Hertz of the bandwidth (both the numerator and the denominator divided by W).

Equation (4.21) shows how the SINR is improved over the full bandwidth reuse case (4.20); the interference is reduced because the outer cell does not interfere any more and the useful signal is increased because the power per bandwidth is not halved. This SINR improvement comes at the loss of bandwidth which in turn impacts the data rate of user u as follows

$$R_u = \eta \delta W \min(4.4, 0.6 \log_2(1 + S_u)) \quad (4.22)$$

Equation (4.22) shows that this approach allows to improve the SINR but reduces the available bandwidth so the choice of δ drives the performance of this approach. We formulate and solve the optimization problem for δ in the following. We denote by \bar{R}_u the data rate of user u when his serving cell is allocated the whole bandwidth so that his actual data rate is $R_u = \delta \bar{R}_u$. If the user u was served by the outer cell, his data rate would be $R_u = (1 - \delta) \bar{R}_u$.

4.2.3.1 α -fair bandwidth sharing

In the bandwidth sharing implementation of VeSn, the sharing proportions have to match the actual proportions of traffic served by the inner and outer cells. The class of α -fair utilities of user throughputs [13],[7] offer a wide range of criteria for choosing these sharing proportions.

In the following, the bandwidth split factor δ is dynamically optimized for any new user configuration, i.e. at every event (arrival or departure). The optimization problem is thus described for one instance of user configuration where the number and positions of users are fixed.

Let us denote by \mathcal{U}_i and \mathcal{U}_o the sets of users in inner and outer cells respectively. The α -fair utility of users' throughputs is given by [7]

$$U_\alpha(\delta) = \begin{cases} \sum_{u \in \mathcal{U}_i} \log(\delta \bar{R}_u) + \sum_{u \in \mathcal{U}_o} \log((1 - \delta) \bar{R}_u) & \alpha = 1 \\ \sum_{u \in \mathcal{U}_i} \frac{(\delta \bar{R}_u)^{1-\alpha}}{1-\alpha} + \sum_{u \in \mathcal{U}_o} \frac{((1-\delta) \bar{R}_u)^{1-\alpha}}{1-\alpha} & \alpha \neq 1 \end{cases} \quad (4.23)$$

When $\alpha = 0$, this utility reduces to the sum of users throughputs. This choice of $\alpha = 0$ is not interesting as no fairness is enforced among the users. The case $\alpha = 1$ corresponds to the well-known proportional fair utility. It can also be shown using queuing theory that the case $\alpha = 2$ corresponds to the sum of the file transfer times in the network.

We show that the utility functions in (4.23) are concave in δ for a given α . Indeed, since the concavity is preserved under linear transformation and non-negative sums [17], it suffices to prove that

$$f_\alpha(x) = \begin{cases} \log x & \alpha = 1 \\ \frac{x^{1-\alpha}}{1-\alpha} & \alpha \neq 1 \end{cases} \quad (4.24)$$

is concave. This function is twice differentiable in $x > 0$ and its second derivative with regard to x is

$$\frac{\partial^2 f_\alpha(x)}{\partial x^2} = \begin{cases} \frac{-1}{x^2} & \alpha = 1 \\ -\alpha x^{-\alpha-1} & \alpha \neq 1 \end{cases} \quad (4.25)$$

So $\frac{\partial^2 f_\alpha(x)}{\partial x^2} \leq 0$ for all $\alpha \geq 0$. As a consequence, $f_\alpha(x)$ is concave by the second order conditions on convexity (see Section 2.4).

Simple gradient descent algorithms [17] can be used to find the optimal δ efficiently.

For $\alpha = 0$ which is unfair, the solution is rather simple and no gradient descent is needed. The

utility function is linear in δ so its maximum is attained at one of the ends $\delta = 0$ or $\delta = 1$, so a simple evaluation of the utility at those ends gives the maximum.

For $\alpha = 1$ (Proportional Fair), a closed form expression of the optimal δ is also given below. Indeed, the KKT conditions are equivalent to finding δ satisfying the following equation

$$\frac{\partial U_\alpha(\delta)}{\partial \delta} = \frac{N_i}{\delta} - \frac{N_o}{1-\delta} = 0 \quad (4.26)$$

where $N_i = |\mathcal{U}_i|$ and $N_o = |\mathcal{U}_o|$ are the number of users in the inner and outer cells respectively. The solution to (4.26) can be easily derived and reads

$$\delta = \frac{N_i}{N_i + N_o} \quad (4.27)$$

when there is at least one user in the cell ($N_i + N_o > 0$). This simple solution does not depend on the particular channel quality of the users present in the cell rendering it particularly effective.

For $\alpha \notin \{0, 1\}$, the gradient descent algorithm is needed. However, in a real network the values of \bar{R}_u which are used by the algorithms are generally fluctuating because of the random nature of wireless channels. Since seeking a good estimate (e.g. average over a long time interval) of these values for each iteration of the optimization would slow down its convergence, a SA algorithm can be applied in which a new estimate of \bar{R}_u is used at every step until convergence of the algorithm. The SA algorithm is of the form

$$\delta[k+1] = \delta[k] + \epsilon \frac{\partial \hat{U}_\alpha(\delta[k])}{\partial \delta} \quad (4.28)$$

where k is the step index, ϵ a small step size and \hat{U}_α the estimate available at step k . If ϵ is sufficiently small and the consecutive estimates of the gradient form a Martingale Difference sequence, this algorithm converges to a neighborhood of the optimal δ (see Section 2.5 for more details on SA).

The proportional fair utility ($\alpha = 1$) brings both fairness and simple implementation. Indeed, δ can be updated in one step at every event using (4.27). Because of these advantages, the PF utility is adopted in the numerical results below.

The bandwidth sharing approach allows to significantly improve the SINR but at the price of reduced bandwidth reuse. It is thus expected to perform better than the full reuse case only when the traffic demand is low and where more bandwidth is not needed. With regard to this observation, a switching mechanism is also needed to enable the full bandwidth reuse when the traffic demand gets higher.

It is noted that a practical implementation of bandwidth-sharing between vertical sectors requires (frequency) synchronization between inner and outer sectors. This requirement is easily fulfilled since the two BSs are generally co-located. Also, in full reuse, reactive VeSn activation will considerably increase the number of HOs. Frequency splitting solves this HO problem.

4.2.3.2 Threshold-based SON Controller

The bandwidth sharing implementation of the VeSn feature indirectly allows to balance the loads between inner and outer cells by applying fairness among all users in the sector. So this solution is naturally robust to load disparity in the cell. The full bandwidth reuse implementation on the other hand is only efficient when the traffic demand in both inner and outer cells is high enough.

With these observations, we propose a simple SON controller based on a load threshold (ρ_{th}) to automatically switch between the two implementations of the VeSn feature according to the traffic demand in the cell. The load is defined as the average over time of the proportion of transmission resources (Physical Resource Blocks (PRBs) in LTE) used.

By denoting ρ_i and ρ_o the inner and outer cells loads respectively, we have for the bandwidth sharing implementation

$$\rho_{i/o}^{sharing} = \frac{\text{average number of PRBs used by inner/outer}}{\text{Total number of PRBs allocated to inner/outer}}, \quad (4.29)$$

and for the full reuse implementation we have

$$\rho_{i/o}^{reuse\ one} = \frac{\text{average number of PRBs used by inner/outer}}{\text{Total number of PRBs}}, \quad (4.30)$$

since each sector can use the whole bandwidth. If elastic traffic is considered and all the resources available are used whenever there is a user to serve, the loads correspond to the proportion of time there are users in the cell.

The SON controller for automatic selection of the implementation of the VeSn feature between bandwidth sharing and full reuse is presented in Algorithm 6.

Algorithm 6 SON controller algorithm

Initialization:

Activate VeSn feature with **bandwidth sharing**

loop:

for $k \in \mathbb{N}, k > 0$ **do**

 Estimate the inner and outer loads during time interval k using (4.29) or (4.30)

if VeSn uses **bandwidth sharing** and $\max(\rho_i^{no\ reuse}, \rho_o^{no\ reuse}) \geq \rho_{th}$ **then**

 Activate VeSn feature with **reuse one**

if VeSn uses **reuse one** and $\max(\rho_i^{reuse\ one}, \rho_o^{reuse\ one}) < \rho_{th}$ **then**

 Activate VeSn feature with **bandwidth sharing**

The maximum between the inner and outer loads is considered in Algorithm 6 in order to avoid attaining congestion in either cell before switching to full bandwidth reuse. A theoretical derivation of the thresholds is difficult because of their dependence on specific network environment (path-loss map) and neighboring interference which can be very fluctuating, so a learning algorithm is needed. Numerous approaches can be considered including reinforcement learning or stochastic

approximation. A simple learning algorithm would be to increase (resp. decrease) the threshold if switching to full reuse (resp. bandwidth sharing) results in a performance degradation.

4.2.3.3 Numerical results

We consider a trisector LTE network surrounded by six interfering macro sites as shown in Figure 4.9. Performance is evaluated only in the central sectors (colored blue in Figure 4.9). In the case where the VeSn feature is implemented, it is deployed only on those 3 central sectors.

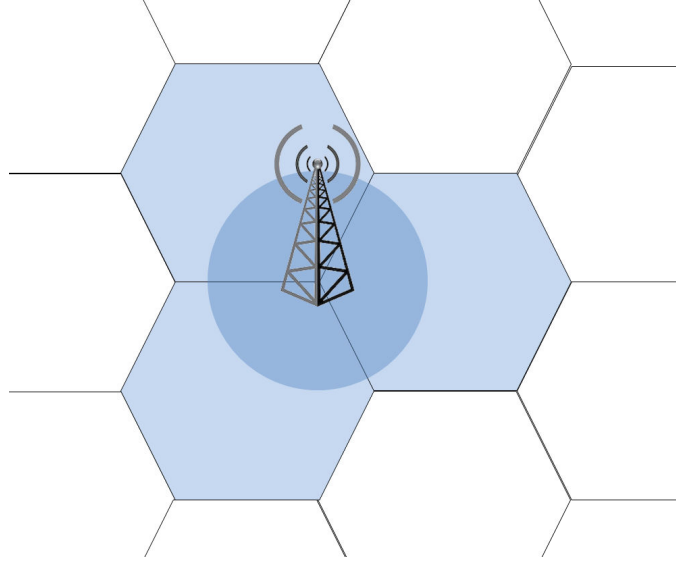


Figure 4.9: Simulation scenario network layout

We also consider elastic traffic in which users arrive in the network according to a Poisson process. The arrival rate is denoted λ so that the inter-arrival times are exponentially distributed with mean $1/\lambda$. Each user downloads a file of exponentially distributed size and leaves the network as soon as the download is complete. Simulation parameters are summarized in Table 4.1. The Matlab program used is event-based, so users' arrivals/departures are simulated. The simulator does not take into account fast-fading so that the data rate of a user is that obtained using round-robin scheduling, in the absence of fading.

We evaluate user performance (MUT and CET) as well as network performance (maximum cell loads) with varying traffic demand for the four following cases:

- **Baseline:** The VeSn feature is not implemented, the total transmit power for each sector is 46dBm. This case is colored black in the Figures.
- **VeSn reuse one:** The VeSn feature is implemented with full bandwidth reuse, the total transmit power is split equally among inner and outer sectors (43dBm each), and each cell uses the whole bandwidth. This case is colored red in the Figures.
- **VeSn bandwidth sharing:** The VeSn feature is also implemented but this time with no reuse, the bandwidth is shared between inner and outer cells according to Equation (4.27) which

Table 4.1: Network and Traffic characteristics

Network parameters	
Number of interfering macros	6×3 sectors
Macro Cell layout	hexagonal trisector
Intersite distance	500 m
Bandwidth	10MHz
Inner Cell Antenna Tilt	18°
Outer Cell Antenna Tilt	12°
Antenna Tilt Type	Electrical only
Channel characteristics	
Thermal noise	-174 dBm/Hz
Macro Path loss (d in km)	$128 + 36.8 \log_{10}(d)$ dB
Traffic characteristics	
Traffic spatial distribution	uniform
Service type	FTP
Average file size	6 Mbits

optimizes the proportional fair utility of users throughputs. This case is colored blue in the Figures.

- **SON controller:** The VeSn feature is implemented following Algorithm 6 which switches automatically between the two implementations according to the traffic demand. The value for ρ_{th} is set to 70% after observing the results from a first run of the three previous cases. This case is colored magenta in the Figures.

The results are presented for a global user arrival rate in the 3 sectors varying from 1 user/s to 10 users/s. Figure 4.10 shows the MUT, Figure 4.11 - the CET and Figure 4.12 - the maximum load observed in all the sectors (3 sectors for the baseline, 6 sectors when the VeSn feature is enabled).

The figures readily show that **VeSn reuse one** improves performance (MUT and CET) over Baseline only when the traffic demand is high enough (here $\lambda \geq 3$). **VeSn bandwidth sharing** on the other hand takes advantage of the higher inner cell signal strength to improve performance (MUT and CET) over the Baseline at all loads. **VeSn bandwidth sharing** also provides better MUT and CET than **VeSn reuse one** for arrival rates less than 7.5 users/s (low to medium load scenarios).

The stability region of a sector is defined as the maximum traffic demand that can be handled by that sector with a load strictly less than one. If the traffic demand is inside the stability region, the mean number of users simultaneously present in the cell remains bounded. As shown in Figure

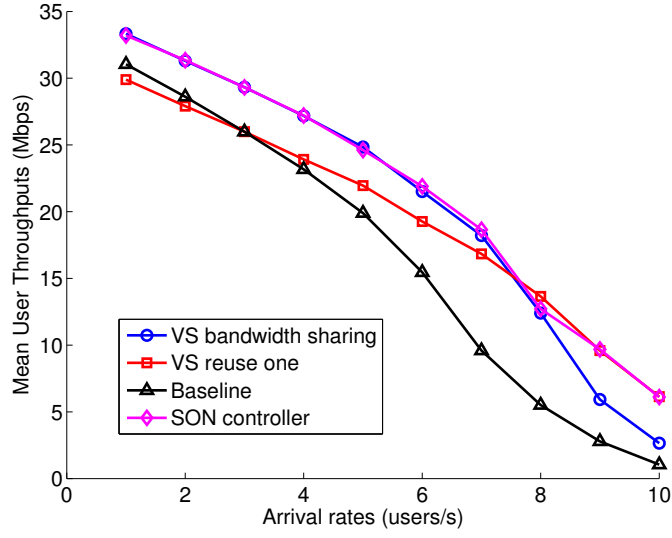


Figure 4.10: Mean user throughput for increasing arrival rates

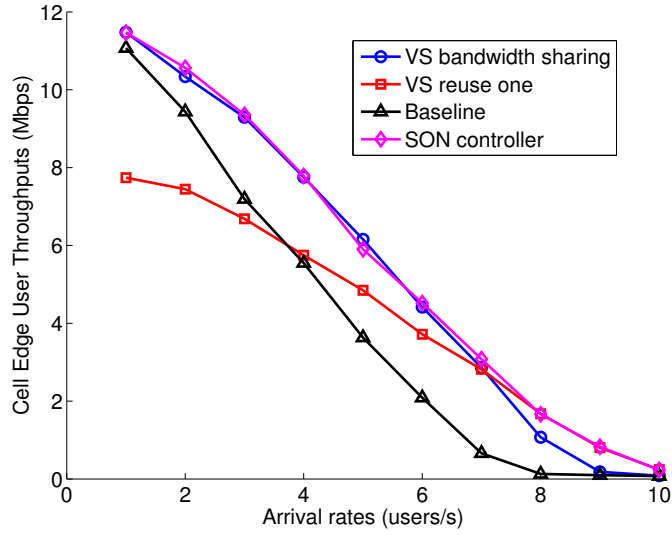


Figure 4.11: Cell edge user throughput for increasing arrival rates

4.12, **VeSn bandwidth sharing** improves the stability of the system over the Baseline. However at high load, full bandwidth reuse is needed. Thus **VeSn reuse one** provides a larger stability region than **VeSn bandwidth sharing**.

As shown by all the performance results (MUT, CET and stability region), the SON controller provides the best of both worlds: the interference gain of **VeSn bandwidth sharing** in the medium to low load traffic demand scenarios, and the increased bandwidth of **VeSn reuse one** for high traffic demands.

These numerical results show the improved performance of the bandwidth sharing implementation of VeSn feature, and how the SON controller improves performance by switching from VeSn with bandwidth sharing to VeSn with reuse one at high loads.

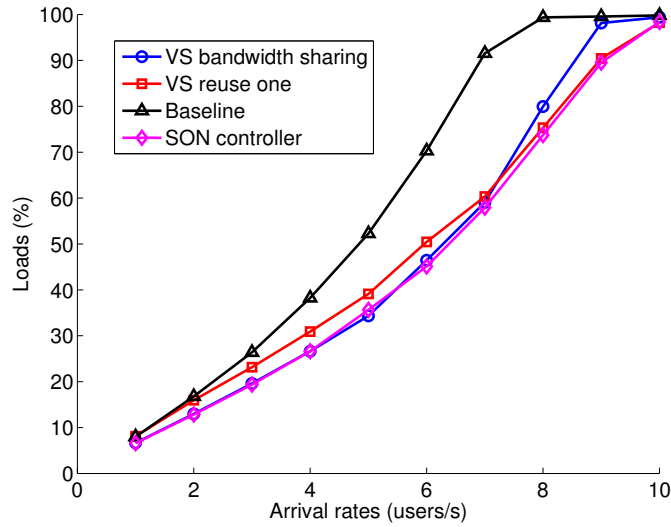


Figure 4.12: Maximum loads for increasing arrival rates

4.3 Virtual sectorization

When a traffic hot-spot is located in the cell but far from the BS, VeSn provides no advantages. In this case, one can deploy small cells at the hot-spot area. The efficiency of small cells grows when the hot-spot is located close to the cell edge. However, backhaul deployment can increase the overall cost of the small cell technology, particularly when optical backhaul is chosen. An alternative solution for small cell deployment is the use of a large antenna array for generating narrow beams for covering the hot-spot area in the cell as in Figure 4.13. A cell covered by a remote beam from an antenna located at- or near to the macro BS is denoted as a ViS.

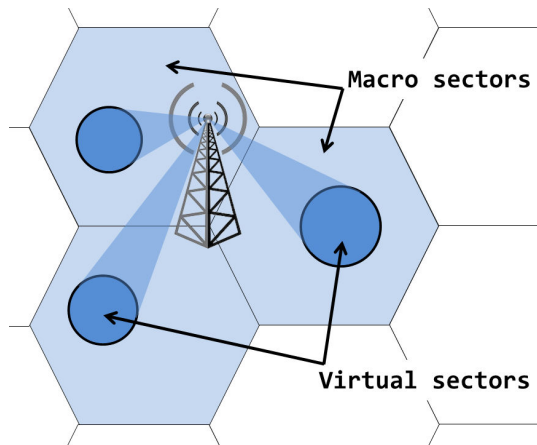


Figure 4.13: Network layout with ViSn enabled

4.3.1 Antenna model

We investigate in this section the design of the array of antenna elements for generating the more focused beams. Consider a $N_x \times N_z$ (sub) antenna array of vertical dipoles, at a distance of $\frac{\lambda}{4}$ from a square metallic conductor, with λ being the wavelength. The full antenna array (all available antenna elements) generates the highest level (narrowest) beams, whereas lower level beams correspond to smaller (rectangular) sub arrays sizes (a fraction of the antenna elements) (see Figure 4.21). If another type of radiating element is chosen, only its radiation pattern should be modified. To simplify the model, we approximate hereafter the reflector as an infinite Perfect Electrical Conductor (PEC). The N_x and N_z elements in each row and column are equally spaced with distances d_x and d_z , respectively (Figure 4.14).

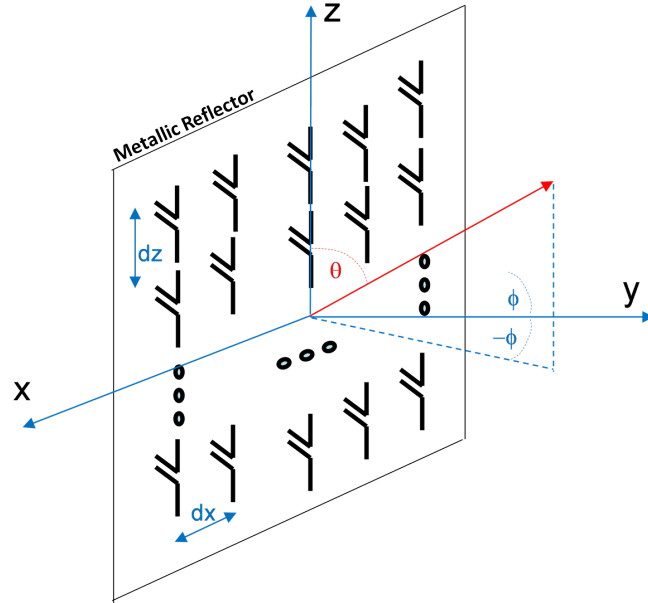


Figure 4.14: Antenna array with dipole radiating elements.

The direction of a beam is determined by the angle (θ_e, ϕ_e) in the spherical coordinates θ and ϕ . The antenna gain for a given beam defined by (θ_e, ϕ_e) in a given direction (θ, ϕ) is written as

$$G(\theta, \phi, \theta_e, \phi_e) = G_0 f(\theta, \phi, \theta_e, \phi_e) \quad (4.31)$$

where f is a normalized gain function and G_0 the maximum gain in the (θ_e, ϕ_e) direction. A separable excitation in the x and z directions is assumed, resulting in the following separable form of f :

$$f(\theta, \phi, \theta_e, \phi_e) = |AF_x^2(\theta, \phi, \theta_e, \phi_e) \cdot AF_y^2(\theta, \phi) \cdot AF_z^2(\theta, \theta_e)| \cdot G_d(\theta) \quad (4.32)$$

$AF_x(\theta, \theta_e, \phi, \phi_e)$ and $AF_z(\theta, \theta_e)$ are the array factors in the x and z directions and are given

by

$$AF_x(\theta, \theta_e, \phi, \phi_e) = \frac{1}{\sum_{m=1}^{N_x} w_m} \sum_{m=1}^{N_x} w_m \cdot a_m \quad (4.33)$$

and

$$AF_z(\theta, \theta_e) = \frac{1}{\sum_{n=1}^{N_z} v_n} \sum_{n=1}^{N_z} v_n \cdot b_n. \quad (4.34)$$

a_m and b_n are complex amplitude contributions of the radiating element located at $(m-1)d_x$ and $(n-1)d_z$, respectively:

$$a_m = \exp\left(-j2\pi \frac{(m-1)d_x}{\lambda} (\sin \theta \sin \phi - \sin \theta_e \sin \phi_e)\right), \quad (4.35)$$

$$b_n = \exp\left(-j2\pi \frac{(n-1)d_z}{\lambda} (\cos \theta - \cos \theta_e)\right). \quad (4.36)$$

The weights w_m and v_n for radiating elements in the m -th row and n -th columns define a Gaussian tapering function used to control the sidelobe level of the gain pattern

$$w_m = \exp\left(-\left(\frac{(m-1)L_x - \frac{L_x}{2}}{\sigma_x}\right)^2\right), \quad (4.37)$$

$$v_n = \exp\left(-\left(\frac{(n-1)L_z - \frac{L_z}{2}}{\sigma_z}\right)^2\right), \quad (4.38)$$

where L_x and L_z are the array size in the x and z directions respectively, with $L_x = (N_x - 1)d_x$ and $L_z = (N_z - 1)d_z$. The values for $\sigma_s, s \in \{x, z\}$, are defined by fixing the ratio between the extreme and center dipole amplitudes respectively to a given value of α_s :

$$\sigma_s^2 = -\left(\frac{L_s}{2}\right) \frac{1}{\log(\alpha_s)}; s \in \{x, z\} \quad (4.39)$$

The impact of the PEC can be modeled by replacing it with the images of the radiating elements it creates. The term $AF_y(\theta, \phi)$ takes into account the images and is written as:

$$AF_y(\theta, \phi) = \sin\left(\frac{\pi}{2} \sin \theta \cos \phi\right) \quad (4.40)$$

The normalized gain pattern of the dipoles, $G_d(\theta)$, is approximated as

$$G_d(\theta) = \sin^3 \theta. \quad (4.41)$$

The term G_0 is obtained from the power conservation equation:

$$G_0 = \frac{4\pi}{\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^\pi f(\theta, \phi) \sin \theta d\theta d\phi}. \quad (4.42)$$

A beam is defined by the (rectangular) sub-array size, and the couple (θ_e, ϕ_e) defines its direction.

4.3.2 System description

The SINR of a user u is modeled as follows

$$S_u = \frac{P^s h_u^s}{N_0 + \sum_{c \neq s} P^c h_u^c} \quad (4.43)$$

where P^c is the transmit power Per Hertz of BS c , h_u^c - the signal attenuation from BS c to user u , $s = \operatorname{argmax}_c P_c h_u^c$ - the best serving cell for user u and N_0 the thermal noise per Hertz. The sum over $c \neq s$ accounts for the interference from other BSs. The frequency diversity is not taken into consideration here.

The pathloss h_u^c comprises the signal attenuation over the air, the shadowing from the environment and the antenna gains at both the transmitter and the receiver. Fast fading is implicitly taken into account via quality tables which map SINR into data rates (averaged over fast fading). The antenna gain at the transmitter is evaluated using Equation (4.31) for a ViS. So a better antenna gain will result in a better SINR. Let us denote by m and v the indexes related respectively to the macro cell and the ViS. The total transmit power available at the macro BS P^0 is split between the macro cell (P^m) and the ViS (P^v), so $P^0 = P^v + P^m$. The SINR of a user served by the macro cell in the presence of a ViS which reuses the whole bandwidth is

$$S_u = \frac{P^m h_u^m}{N_0 + P^v h_u^v + \sum_{c \neq s} P^c h_u^c} \quad (4.44)$$

and the SINR of a user served by the ViS is

$$S_u = \frac{P^v h_u^v}{N_0 + P^m h_u^m + \sum_{c \neq s} P^c h_u^c}. \quad (4.45)$$

In the remainder of this section especially in the simulation results, we consider only the case where $P^v = P^m = \frac{P^0}{2}$. Equations (4.44) and (4.45) clearly show the SINR degradation (reduced useful signal, increased interference) when the ViS is activated with frequency bandwidth reuse one.

If instead, the macro cell and the ViS operate on disjoint frequencies, then the SINR of a macro

user is the same as (4.43) while the SINR of a user served by the ViS becomes

$$S_u = \frac{P^v h_u^v}{N_0 + \sum_{c \neq s} P^c h_u^c} \quad (4.46)$$

where $P^v = P^m = P^0$ since the power available per unit bandwidth does not change. An appropriate choice of the bandwidth sharing proportions is then needed in order to avoid performance degradation. We use the proportional fair sharing criteria which provides a good trade-off between throughput optimization and fairness in resource sharing [14],[47], [71].

4.3.3 Self-optimizing algorithms

We use here the same approach as in Section 4.2.3.1. Denote by δ the fraction of the frequency bandwidth dedicated to the ViS and \bar{R}_u the mean data rate of a user served by either the macro cell or the ViS when the other is switched off. The proportional fair utility is defined as

$$U_{PF}(\delta) = \sum_{u \in \text{ViS}} \log(\delta \bar{R}_u) + \sum_{u \in \text{macro}} \log((1 - \delta) \bar{R}_u) \quad (4.47)$$

Since the utility function (4.47) is concave, maximizing it is a convex optimization problem. Using KKT conditions for optimality [17], the optimal value of δ can be easily derived to be (see Section 4.2.3.1 for detailed derivation)

$$\delta = \frac{N_v}{N_v + N_m} \quad (4.48)$$

where N_v, N_m are respectively the number of users in the ViS and the macro sector.

Equation (4.48) constitutes the self-optimization algorithm used to update the bandwidth sharing proportions between the macro sector and the ViS and the update is performed at each event (arrival/departure). It is noted that a general α -fair utility [7] can be used and the optimization problem can be solved using a similar method as in Section 4.2.3.1.

4.3.4 Numerical results

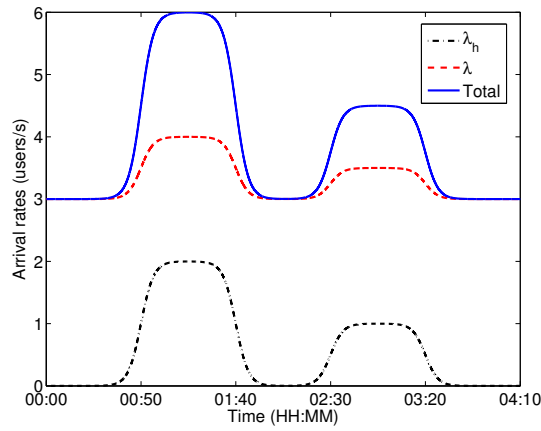
4.3.4.1 Simulation scenario

Consider a trisector BS surrounded by 2 rings of interfering macro sites as shown in Figure 4.13. In each macro sector, a ViS can be activated whenever needed. We consider elastic traffic where users arrive in the network according to a Poisson process, download a file and leave the network as soon as their download is complete. The considered area A is the initial area covered by the central macro BSs. In order to limit the complexity, slow and fast fading are not taken into account in these simulations and users are assumed static. However the users arrive at random locations in the network.

Table 4.2: Network and Traffic characteristics

Network parameters	
Number of macro sectors	3
Number of ViSs	3
Number of interfering macros	2 rings of macro sites
Macro Cell layout	hexagonal trisector
Intersite distance	500 m
Bandwidth (B)	10MHz
BS transmit power	40W (46dBm)
Scheduler	Round-Robin
Link adaptation model	$B \min(4.4, \log_2(1 + \text{SNR}))$ [3]
Channel characteristics	
Thermal noise	-174 dBm/Hz
Path loss (d in km)	$128.1 + 37.6 \log_{10}(d)$ dB
Traffic characteristics	
Traffic spatial distribution	uniform + hot-spots
Service type	FTP
Average file size	3 Mbits

Traffic is composed of two layers: the first one has a uniform arrival rate of λ users/s all over A , and the second - a uniform arrival rate of λ_h users/s in the ViSs coverage area. These arrival rates evolve over time as shown in Figure 4.15 in order to show the effect of the self-optimization algorithm. For example, between 00:50 and 01:40, the hot-spot traffic demand (λ_h) increases from 0 to 2 users/s. This is close to a realistic scenario where the ViSs' beams are set to point at the hot-spot areas by adjusting the θ_e , and ϕ_e angles (used to compute the antenna gain in Equation (4.31)).

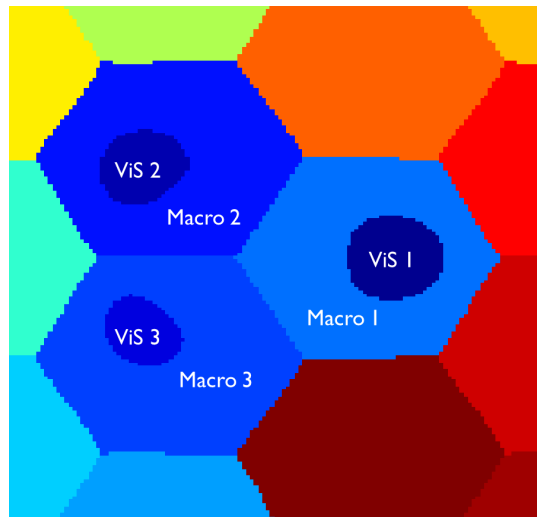
**Figure 4.15:** Traffic profile over time (HH:MM means hours:minutes)

We use the propagation models for all BSs following [1, Page 61] and presented in Table 4.2

Table 4.3: ViSs antenna configurations

	ViS 1	ViS 2	ViS 3
Vertical tilt	10°	11°	12°
Horizontal tilt	0°	10°	-15°
N_x	10		
N_z	40		

which also summarizes all the simulation parameters. The best server map obtained from these parameters is presented in Figure 4.16. The parameters used for each ViS are summarized in Table 4.3. The vertical tilt is defined with respect to the horizon and the horizontal tilt has the azimuth of the containing macro sector as reference (see Figure 4.14).

**Figure 4.16:** Best server map

4.3.4.2 Performance Evaluation

We evaluate the MUT (Figure 4.17), the CET (Figure 4.18), the maximum loads (Figure 4.19) and the FTT (Figure 4.20) for three different cases:

- Baseline (black in Figures): this is the reference case in which no ViS is present, so the macro sectors serve all the traffic as they would do traditionally.
- ViSn reuse one (red in Figures): in this case, the ViSs are deployed with a full reuse of the bandwidth. The macro and virtual sectors share equally the available transmit power.
- ViSn bandwidth sharing (blue in Figures): the ViSs are also enabled in this case but the total bandwidth is shared between the macro cell and the ViS in its coverage area. The bandwidth sharing proportions are dynamically optimized using (4.48).

It is noted that the CET refers to the 5th percentile throughput, so it will correspond generally to users at the macro cell edge in our scenario (no interference between macro cell and ViS).

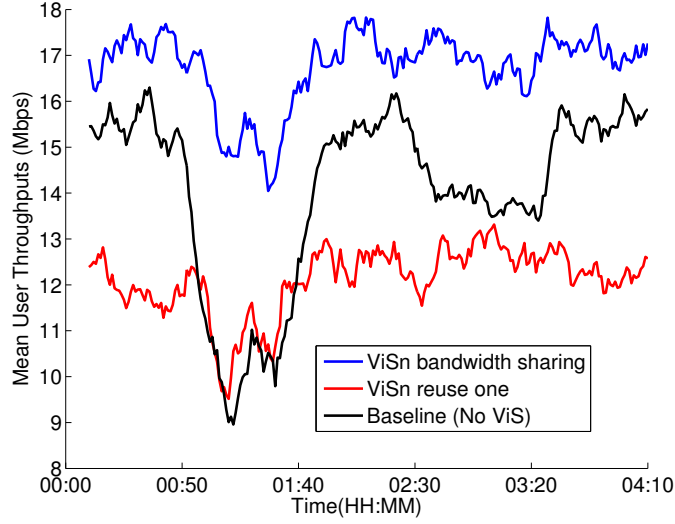


Figure 4.17: Mean user throughput evolution over time

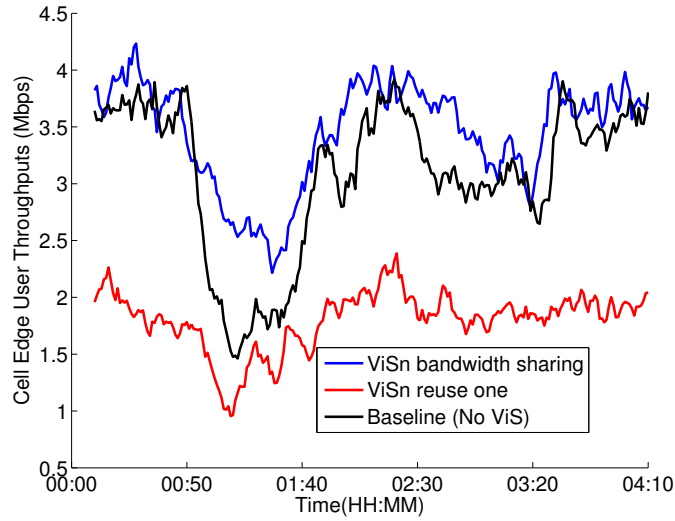


Figure 4.18: Cell edge user throughput evolution over time

The numerical results show that deploying the ViS with full reuse of the bandwidth degrades performance over the baseline (No ViS) except for sufficiently high loads. Indeed the CET and the FTT of the baseline are always the same or better than those of the reuse one case. It is only between 00:50 and 01:40 that the MUT of the baseline is slightly worse than that of reuse one (see Figure 4.17), and it can be seen in Figure 4.19 that the mean load at this time is over 75% for both baseline and reuse one.

The reuse one case degrades performance because of its worse SINR (reduced power due to its split between macro and virtual cells, and macro-virtual cells mutual interference). This scheme is

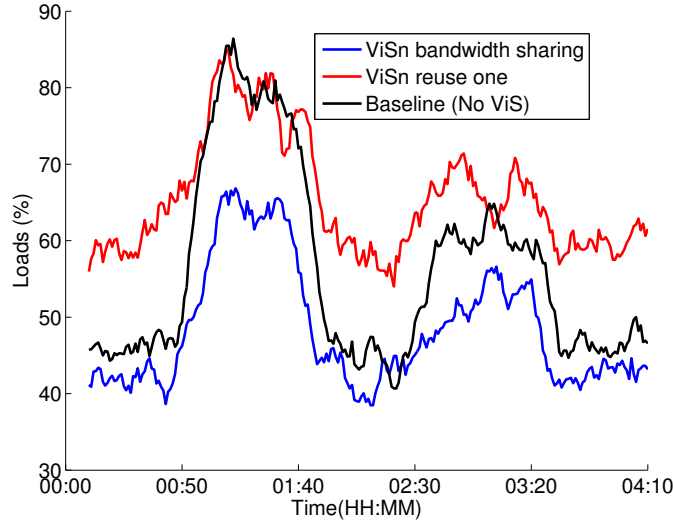


Figure 4.19: Maximum loads' (all cells, virtual and macro) evolution over time

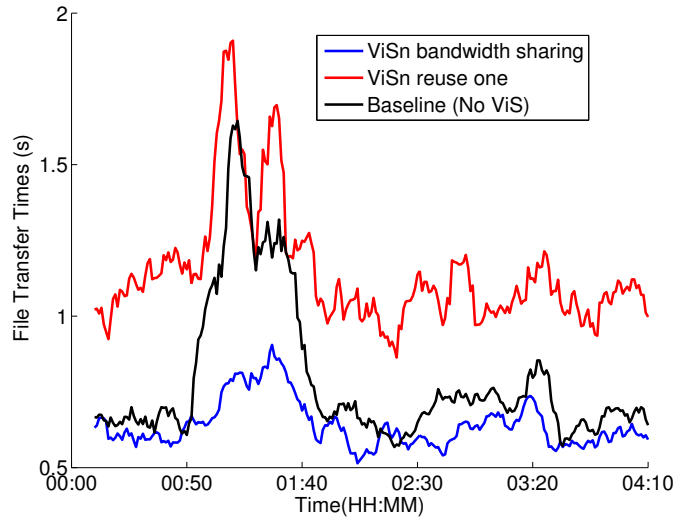


Figure 4.20: File transfer time's evolution over time

then only useful at very high loads (over 85%). It is noted that similar results have been obtained in Section 4.2 for VeSn.

Deploying the ViS with bandwidth sharing is shown to provide the best performance during the whole simulation period for different load conditions as shown by all performance indicators in Figures 4.17, 4.18 and 4.20. Even the loads (Figure 4.19) are lower suggesting that deploying ViS with bandwidth sharing provides a higher capacity.

The higher gain of the ViS antenna improves the SINR over the baseline case. Moreover, the bandwidth sharing enables the two cells (macro sector and ViS) to serve their traffic without mutual interference and a better SINR compared to a ViS deployed with full bandwidth reuse. It is noted that the bandwidth reuse one is expected to provide better performance when the loads approach 100% as for VeSn (see Section 4.2).

4.4 Hierarchical beamforming

A further step in AAS technology evolution is the dynamic adaptation of the beam to each user, often referred to in the literature as beamforming. The idea is to be able to reconfigure the antenna parameters (i.e. electrical tilts) online so that the beam used to serve a particular user is adjusted to get the maximum RSRP. The antenna gains brought by such technology increase with the number of antenna elements of the antenna array, but so does also the feedback overhead needed to adjust the antenna parameters.

In order to reduce this overhead, a codebook of antenna parameters can be used in which each code corresponds to a specific configuration of the antenna in terms of beam width and beam direction. The user feedback can then be used to select the best code. However, depending on the number of antenna elements in the array, a large number of codes can be obtained and a lot of feedback is still be required.

In this section, we propose to construct the beams (codes) offline in a hierarchical manner so that each user iteratively selects his best beam by narrowing the beam width at each iteration, the feedback needed is then logarithmic instead of linear in the number of possible codes. This technique has been used in [35] for fast configuration of narrow beams for millimeter wave wireless backhaul. We use the same technique for multi-user communications in LTE networks but for a 3D antenna model and show that the performance gains are significant for even relatively low number of antenna elements.

It is noted that the expressions "hierarchical beamforming" and "multilevel beamforming" are used interchangeably throughout this section.

4.4.1 Antenna design methodology

This Section presents the guidelines for the antenna array design supporting multilevel beamforming. The (sub) antenna array design constitutes an optimization problem with two objectives: maximizing the antenna gain (or conversely, minimizing the width of the main lobe) and minimizing the side-lobes' level, as a function of the parameters d_s and α_s , $s \in \{x, z\}$ (see Section 4.3.1).

The problem is written as a constrained optimization problem:

$$\underset{d_x, d_z, \alpha_x, \alpha_z}{\text{maximize}} G_0(N_{x,\max}, N_{z,\max}, d_x, d_z, \theta_e, \phi_e) \quad (4.49a)$$

$$\text{s.t.} \quad (4.49b)$$

$$\max_{N_x, N_z} \{SL(N_x, N_z, d_x, d_z, \theta_e, \phi_e)\} \leq Th; \quad (4.49c)$$

$$N_{s,\min} \leq N_s \leq N_{s,\max}; s \in \{x, z\}; \quad (4.49d)$$

$$0 < \alpha_s \leq 1; s \in \{x, z\}; \quad (4.49e)$$

$$0 < d_s \leq \frac{\lambda}{2}; s \in \{x, z\}; \quad (4.49f)$$

$$\theta_{\min} \leq \theta_e \leq \theta_{\max}; \quad (4.49g)$$

$$-\phi_{\max} \leq \phi_e \leq \phi_{\max}; \quad (4.49h)$$

where $N_{s,\min}$ and $N_{s,\max}$ are respectively the minimum and maximum number of antenna elements in the s direction, $\theta_{\min}, \theta_{\max}, \phi_{\min}, \phi_{\max}$ are respectively the minimum and maximum electrical elevation and azimuth angles of the antenna array. The constraint (4.49c) reads: the maximum side-lobe level for the whole range of sub array size should be below a predefined threshold Th .

It is noted that for small elevation electrical tilt values, the beams will likely cover larger areas. Special care should be taken when setting the $\theta_{\min}, \phi_{\min}, \phi_{\max}$ angles in order to avoid overshooting on neighboring cells. These angles will depend on the geometrical characteristics of the cell (original coverage area of the considered BS before deploying the antenna array), such as the cell shape, size, and antenna height. One can consider optimizing the antenna for a wide range of elevation and azimuth angles, and then, according to the cell geometry, construct a codebook of beams for a desired angular range. Furthermore, a database with a set of codebooks can be pre-optimized for a set of cell geometries and then, according to the specific cell deployment, the most suitable codebook can be selected.

4.4.2 Multilevel beamforming algorithm

4.4.2.1 Beams structure

Consider a multilevel codebook as shown in Figure 4.21, with L levels and J_l beams at level l , $l = 0, \dots, L$. The j th beam at level l is written as $B_{l,j}(\theta, \theta_{l,j}, \phi, \phi_{l,j})$, $j = 1, \dots, J_l$, and for brevity of notation - as $B_l(j)$. It is noted that the angles $(\theta_{l,j}, \phi_{l,j})$ correspond to (θ_e, ϕ_e) defined in Section 4.3.1.

The beam of the first level, namely level 0 in Figure 4.21, $B_0(1)$, covers the entire cell. Beams at level l are generated by the same sub-array, i.e. with the same number of array elements, and differ from each other by the angles $(\theta_{l,j}, \phi_{l,j})$. Denote by \hat{C} a set-valued function which receives as argument a beam, and outputs its coverage area (often denoted as the best server area), where the beam provides the strongest signal with respect to other cell or beam coverage. The coverage

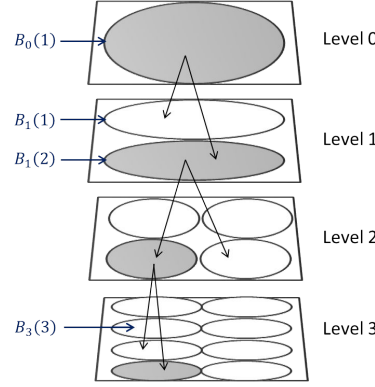


Figure 4.21: Example of beam hierarchy

operator can be obtained through a network simulator. By construction we assume that for a given level l , the beams' coverage do not overlap:

$$\hat{C}(B_l(i)) \cap \hat{C}(B_l(j)) = \emptyset, \forall i \neq j. \quad (4.50)$$

The multilevel structure of the beams in the codebook implies that a given beam $B_l(j)$ at level l where $l < L$ has two children beams $B_{l+1}(2j-1)$ and $B_{l+1}(2j)$ with

$$\hat{C}(B_{l+1}(2j-1)) \cup \hat{C}(B_{l+1}(2j)) \subset \hat{C}(B_l(j)) \quad (4.51)$$

The beams at level L are the narrowest that can be obtained given the $N_{x,max} \times N_{z,max}$ antenna array.

4.4.2.2 Beam selection algorithm

The beam selection algorithm consists in finding the best beam available by navigating through the multilevel codebook. It starts with $B_0(1)$ which covers the entire cell and keeps track of the overall best beam up till now (B^*). Assuming the beam selection algorithm is at level l with $l < L$, the best beam is updated as follows

$$B^* = \arg \max_{B \in \{B^*, B_{l+1}(j), j=(2j_{l+1}-1, 2j_{l+1})\}} S(B, u) \quad (4.52)$$

where $S(B, u)$ is the SINR of user u served by the beam B , and j_l denotes index of the beam at level l . It is noted that the RSRP can be used instead of the SINR in (4.52).

The algorithm stops when the best beam does not change in a given iteration, i.e. the parent beam provides better SINR than the children beams, or the highest level of beams L is reached. The complexity of such an algorithm is $\log(N)$, N being the total number of beams, hence convergence is obtained in a very small number of iterations.

It is noted that condition (4.51) can be relaxed so that narrower beams can cover regions not

covered by their parent beams. In this case the beam selection algorithm should continue until level L . The multilevel beam structure presented in this section is just one illustration of how a multilevel codebook structure can be designed. Other approaches can be adopted regarding the relation between parent and children beams in terms of coverage. The only requirement is to be able to easily navigate through the codebook in an iterative manner.

The beam selection algorithm runs in parallel with the scheduling algorithm. A user is first scheduled based on its SINR obtained with the level 0 beam. During the scheduling period, the user's best beam is updated based on the received feedback using equation (4.52). If the scheduling period is long enough, the beam selection algorithm can converge. Otherwise the beam selection resumes at the next scheduling period based on the SINR of the best beam tested so far. It is noted that other scheduling strategies can be considered.

4.4.2.3 Implementation issues

The beam codebook can be precalculated for a given cell and stored in a database of a self-configuration server at the management plane. Upon deployment of multilevel beamforming feature, the multilevel codebook is downloaded from the server to the BS. As mentioned previously, the codebook selection can be based on geometrical characteristics of the cell.

The antenna array can be dynamically configured using different approaches. The classical approach would be to feed each antenna element by a distinct amplifier which receives the appropriate input signal necessary to excite the selected beam from the codebook. More recently, the *load modulated massive MIMO* approach has been reported [53] in which the baseband input signal is used to adapt a load behind each antenna element which controls its complex input impedance. This approach which utilizes a single amplifier aims at further reducing the antenna size and cost, and could be a candidate technology for the multilayer beamforming (further studies are still necessary).

The size of the antenna array depends on the number of antenna elements and the spacing between them. In Section 4.4.3.1, we use $N_{x,max} = 12$ and $N_{z,max} = 32$ antenna elements with spacings of $d_x = 0.5\lambda$ and $d_z = 0.7\lambda$. For the LTE technology with a 2.6 GHz carrier, the antenna size is of 0.69 m \times 2.58 m. Multilevel beamforming will be particularly attractive for 5G networks where higher frequency bands will be available, allowing moderate size of antenna array with a large number of radiating elements.

4.4.3 Numerical results

We present in this subsection numerical results for multilevel beamforming. Two scenarios are considered: a mass event type of scenario in an urban environment with a crowded open area, e.g. an esplanade, and a rural environment in which the users have a LoS path component from the BS. Multilevel beamforming is applied to one cell which is interfered by two rings of neighboring BSs.

A LTE event-based simulator coded in Matlab is used (see Appendix B for more details on

Table 4.4: Network and Traffic characteristics

Network parameters	
Number of sectors with multilevel beamforming	1
Number of interfering macros	2 rings, 20 sectors
Macro Cell layout	hexagonal trisector
Antenna height	30 m
Bandwidth	10MHz
Scheduling Type	Proportional Fair
Channel characteristics	
Thermal noise	-174 dBm/Hz
Path loss (d in km)	$128.1 + 37.6 \log_{10}(d)$ dB
Shadowing	Log-normal (6dB)
Traffic characteristics	
Service type	FTP
Average file size	4 Mbits

the simulator). Users (data sessions) arrive according to a Poisson process, download a file of exponential size with mean of 4 Mbits, and leave the network as soon as their download is complete. We focus on the case where there is no mobility. The channel coherence time of several milliseconds is assumed (which is typically the case for low mobility), and so the beam selection algorithm converges within this coherence time. Hence beam selection errors due to fast-fading are not considered.

The main simulation parameters used for the two scenarios are summarized in Table 4.4.

4.4.3.1 Mass event scenario

Consider a cell with a very large hotspot described by a truncated spatial Gaussian traffic distribution which is superimposed on a uniform traffic over the whole cell area as shown in Figure 4.22. This distribution represents the traffic intensity map in arrivals per second per km^2 . The hotspot can represent a crowd watching a live concert held on an esplanade for example. It is assumed that the users have a significant direct (LoS) path with the BS. It is recalled that in a rich scattering environment, other MIMO techniques are more appropriate. In order to take into account the residual multipaths due to reflections on neighboring buildings, we use the Nakagami-m distribution for the fast fading.

Illustration of multilevel beamforming: Figures 4.23, 4.24 and 4.25 represent respectively the coverage maps of the beams in each level, the best beam chosen at each level for a user located

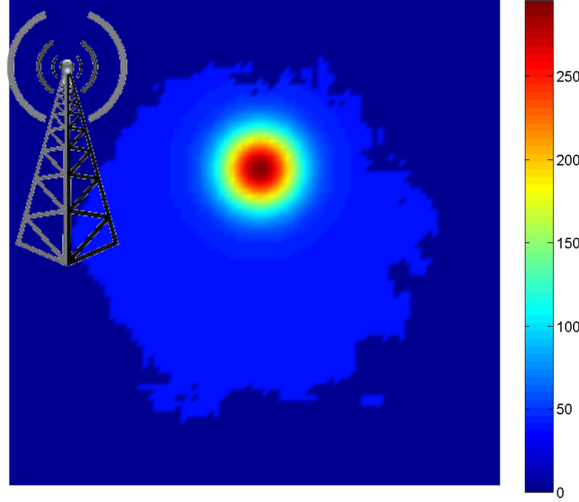


Figure 4.22: Traffic intensity map (in users/s/km²).

at the center of the hotspot area (yellow square in Figure 4.24) and the corresponding antenna diagrams, respectively, as described below.

The multilevel beam structure presented in Section 4.4.2 is illustrated in Figure 4.23. Here, the condition (4.50) is met by definition of the coverage areas. However, condition (4.51) is relaxed in order to allow narrower beams (e.g. level 3 in Figure 4.23d) to cover blank spaces left by wider beams (e.g. level 2 in Figure 4.23c).

Figure 4.24 presents the beam selection algorithm performed according to Equation (4.52) for a given user. The SINR of the selected user gradually increases from 7.75dB to 17.38dB at three iterations, i.e. by a factor of 9. It is noted that the SINR gains are expected to be even higher for cell edge users.

The antenna diagrams corresponding to the selected beams in each level in Figure 4.24 are presented in Figure 4.25. E and H planes respectively designate the vertical and horizontal planes. These diagrams were designed according to the optimization problem (4.49), with the side-lobe level constraint (4.49c) of $Th = 30\text{dB}$. One can see that the beam width of the main lobe gets narrower in elevation or azimuth plane and the maximum gain increases (from 23.76dB to 30.2dB) with the beam level. The antenna diagram for level 0 which correspond to the full cell coverage is omitted.

Results: We next evaluate the performance of multilevel beamforming for the mass event urban scenario. We use the Nakagami-m distribution which models fast-fading in environments with strong LoS component and many weaker reflection components. The *shape parameter* m dictates the contribution of the LoS component in the overall signal. For $m = 1$, there is no LoS component and the fast fading reduces to a Rayleigh distribution. As m grows to infinity, the LoS component dominates. We consider various Nakagami-m fading scenarios with $m = 2, 5$ and 10 , and the no-fading case (corresponding to $m = +\infty$). We do not consider the $m = 1$ case due to

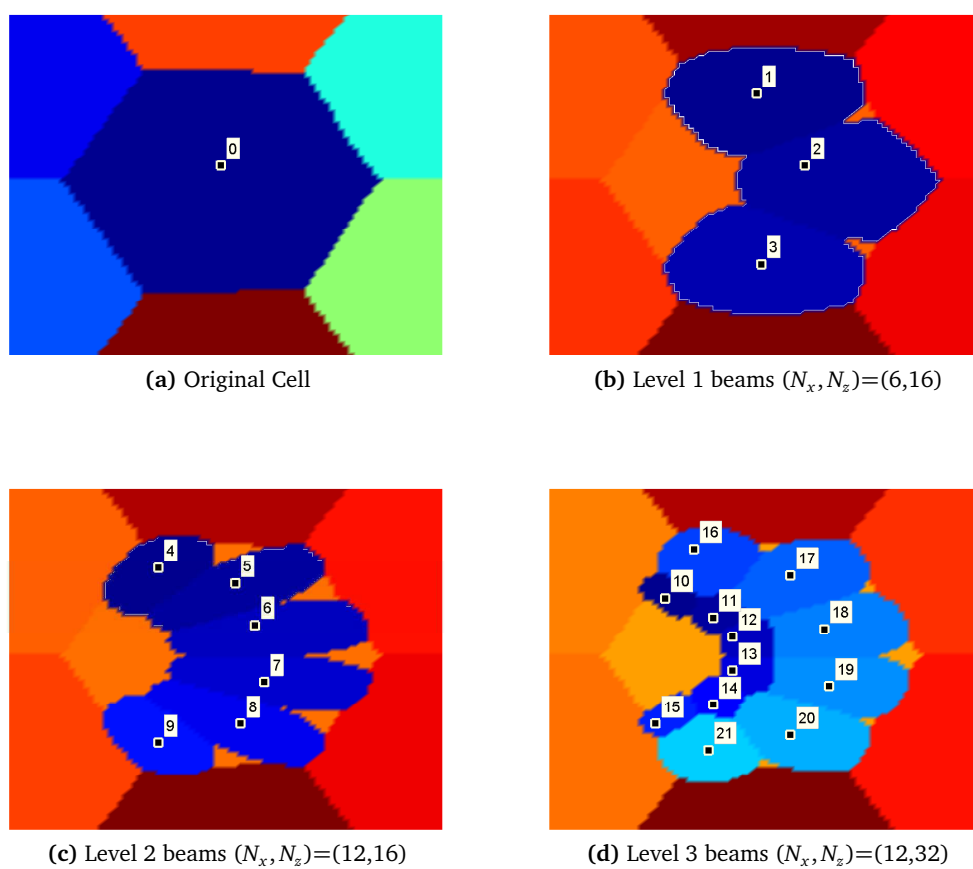
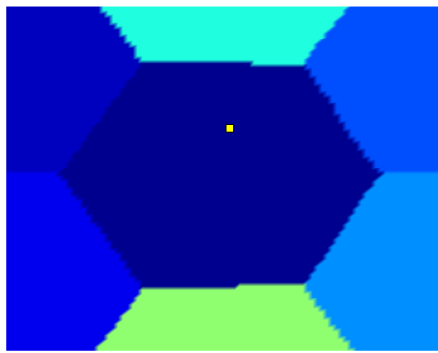
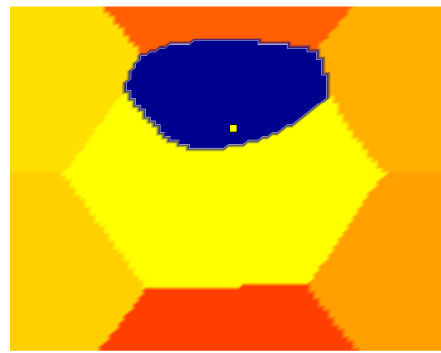


Figure 4.23: Multilevel beam coverage maps for the mass event scenario



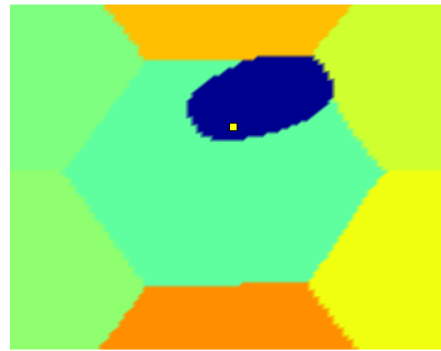
(a) Level 0 beam ($\text{SINR} = 7.75\text{dB}$)



(b) Level 1 beam ($\text{SINR} = 12.52\text{dB}$)



(c) Level 2 beam ($\text{SINR} = 14.24\text{dB}$)



(d) Level 3 beam ($\text{SINR} = 17.38\text{dB}$)

Figure 4.24: Successive narrowing of the beam for a given user

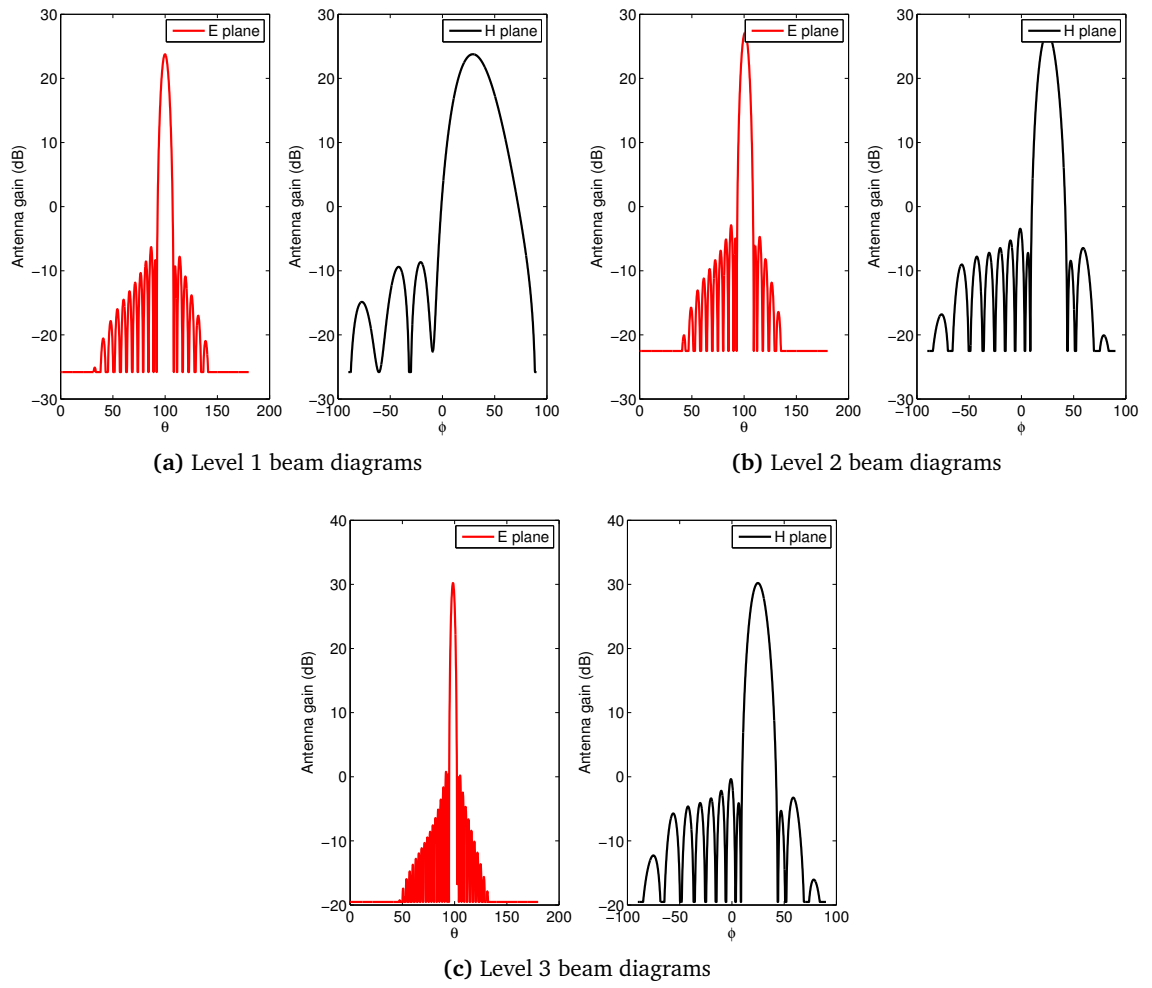


Figure 4.25: Antenna diagrams for a given user's best beam in each level.

Table 4.5: Network and traffic characteristics for the mass event scenario

Intersite distance	500 m
Nakagami-m shape parameter	2, 5 or 10
Traffic spatial distribution	Gaussian Hotspot + Uniform (see Figure 4.22)

the open environment considered in the scenario. The simulation parameters for the scenario are summarized in Table 4.5.

Figure 4.26 presents the frequencies of selected beams throughout the simulation. As expected, beam 17 which covers most of the hotspot region (see Figures 4.22 and 4.23d) is the most frequently selected. So the beam selection algorithm successfully locates the traffic in the direction of the hotspot and adjusts the beam width without any prior knowledge of the hotspot location and size.

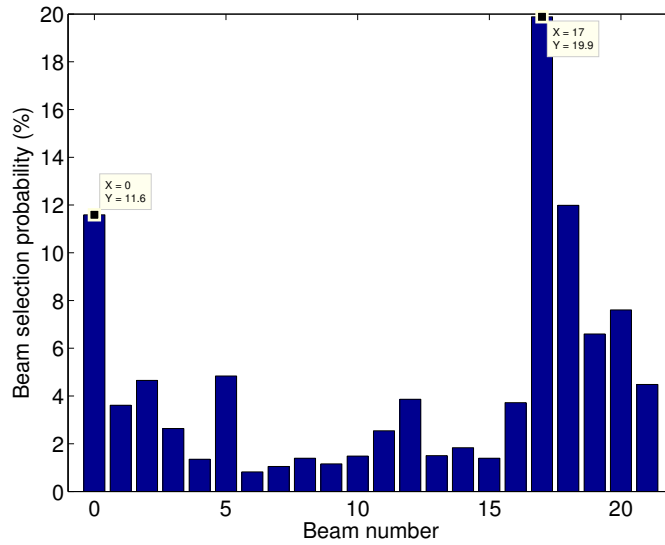


Figure 4.26: Histogram of selected beams throughout the simulation: X is the beam number (see Figure 4.23) and Y - its selection probability.

Table 4.6 presents the MUT, the CET and the Power Consumption (PC) obtained for the various shape parameters of the Nakagami-m fading distribution, with and without multilevel beamforming (denoted respectively as 'w.' and 'wo.' in Table 4.6). For example, in Table 4.6, '2 wo.' means $m=2$ without multilevel beamforming. The PC is evaluated using the approximate linear PC model given in [37, Eq. (4-3)]

$$P_c = P_0 + \alpha P \quad (4.53)$$

where $P_0 = 260W$ is the PC for zero-load, $\alpha = 2 \times 4.7$ is the scaling factor term for an antenna with two-transmission chains and P is the total transmit power when serving a user with the entire

Table 4.6: Performance gain using multilevel beamforming for the mass event scenario

m	MUT (Mbps)	CET (Mbps)	PC (W)
2 wo.	7.64	1.78	397
2 w.	21.49 (181%)	5.52 (210%)	334 (-15.92%)
5 wo.	7.21	1.35	400
5 w.	22.33 (210%)	5.59 (312%)	331 (-17.24%)
10 wo.	6.97	1.18	402
10 w.	22.43 (222%)	5.31 (349%)	331 (-17.51%)
$+\infty$ wo.	4.99	0.51	417
$+\infty$ w.	21.85 (337%)	4.75 (822%)	334 (-19.98%)

bandwidth.

The performance results show particularly high gain brought about by multilevel beamforming. The MUT is improved by a factor varying from 2.81 to 4.38, the CET - from 3.1 to 9.22, and the PC is reduced by 15.9 to 20 percent for m varying from 2 to $+\infty$ respectively. The difference in performance gain between MUT and CET is due to the fact that cell edge users have initially low SINR and their SINR gain with beam focusing is larger. The PC is reduced due to the significant reduction in the sojourn time of the users so the BS transmits less often.

The performance gains increase with the value of m , namely with the importance of the LoS component relative to the multipaths' components. It is recalled that in an environment rich with many scatterers, the initial level of beams (i.e. level 0 in Figure 4.21) will benefit from higher diversity gain by using an opportunistic scheduler (e.g. PF) and therefore the gain obtained by the multilevel beamforming is smaller. This observation further supports the claim that the multilayer beamforming is of particular interest for open type of environment having a significant LoS propagation.

Optimality gap of hierarchical beamforming: We now compare the performance of three cases: no beamforming, hierarchical beamforming and optimal beamforming. The theoretically optimal beamforming consists in evaluating exactly the position of the user and directing the most focused beam (with all antenna elements activated) towards him. Basically, if the user has the position (x, y, z) in a coordinate system with origin the BS antenna, then the antenna is calibrated so that

$$\theta_e = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \quad (4.54)$$

$$\phi_e = \arctan\left(\frac{y}{x}\right) \quad (4.55)$$

The same parameters as in Tables 4.4 and 4.5 are used in particular for the traffic demand, but we considered no fast fading. Figure 4.27 present the user throughput CDFs for the three cases. It

is shown that the proposed multilevel beamforming largely improves over the no-beamforming but still remains inferior to a theoretically optimal beamforming. However, optimal beamforming is very costly in terms of implementation complexity because extremely precise user location is needed.

It is noted that discontinuities in the CDFs of Figure 4.27 are due to the rate model (3.1) since when two or more users are present in the cell and all of them can reach the maximum peak rate (4.4bps/Hz in our case), then their mean data rate becomes a fraction of that maximum peak rate. With beamforming (multilevel or optimal), many users achieve the maximum peak rate which explains the discontinuity in the CDFs at fractions of the maximum peak rate.

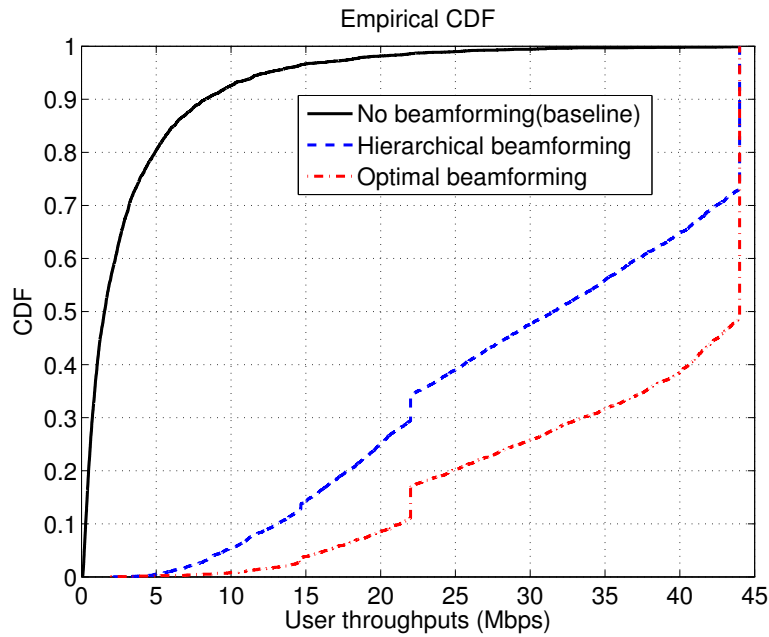


Figure 4.27: User throughput CDFs comparison between optimal, hierarchical and no beamforming.

4.4.3.2 Rural scenario

The simulation parameters for the rural scenario are summarized in Table 4.7. Unlike the mass event urban scenario, the large cell size make vertical beam separation complex. A modification of the beam direction in elevation by a fraction of a degree results in significant difference in its coverage. For this reason, we consider multilevel beamforming in the horizontal (azimuth) plane, as shown in Figure 4.28.

For the sake of simplicity, fast-fading is not considered here. However, similar results as those presented for the mass event urban scenario (see Section 4.4.3.1) are expected: the performance gains increase with the shape parameter m of the Nakagami- m fading.

Table 4.8 compares performance results for MUT, CET, and PC using different numbers of beamforming levels. The performance of level k corresponds to the case where equation (4.52) is applied to a highest beam level set to k . In the rural scenario, the performance gains are also very

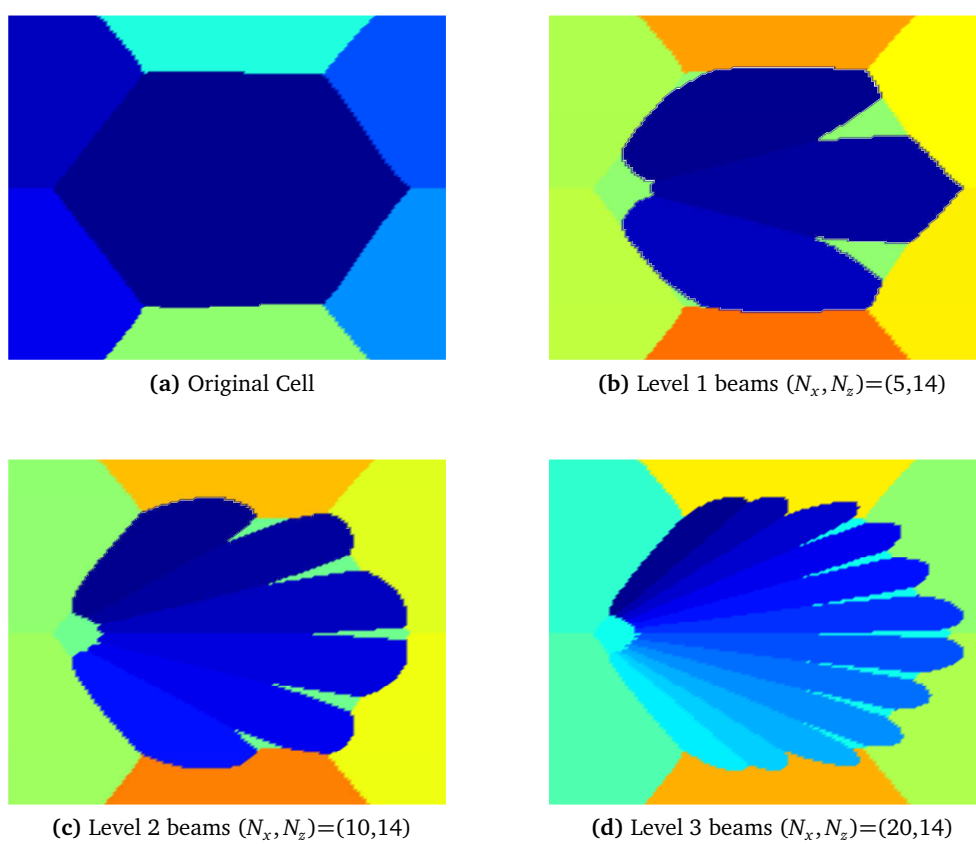


Figure 4.28: Coverage maps for different beamforming levels for the rural scenario

Table 4.7: Network and traffic characteristics for the rural scenario

Intersite distance	1732 m
Fast-fading	None
Traffic spatial distribution	uniform
Arrival rate	2.5 users/s/km ²

Table 4.8: Performance gain using multilevel beamforming for the rural scenario for different beam levels

	MUT (Mbps)	CET (Mbps)	PC (W)
Level 0	4.66	0.43	421
Level 1	9.9 (112%)	1.15 (168%)	388 (-7.93%)
Level 2	13.23 (184%)	1.96 (360%)	369 (-12.3%)
Level 3	15.78 (238%)	2.57 (501%)	356 (-15.42%)

high. For example, for three levels of beams, MUT and CET are increased by 238 and 501 percent. It is noted that the gain achieved is lower than that obtained in the mass event urban scenario. The reason for this is the smaller number of antenna elements used in the rural scenario which results in lower antenna gains. For example, in the third (highest) level, the number of antenna elements are $(N_{x,max}, N_{z,max}) = (20, 14)$ and $(N_{x,max}, N_{z,max}) = (12, 32)$ in the rural and the mass event scenarios, respectively.

4.5 Conclusion

In this chapter, we have presented three applications of the AASs technology namely VeSn, ViSn and hierarchical beamforming. We proposed self-optimizing algorithms for their optimal activation or operation. For VeSn, we proposed an optimal activation controller which decides when to enable the VeSn feature with full frequency reuse depending on the traffic distribution and intensity. We also proposed optimal bandwidth sharing algorithms in case the VeSn is deployed with no frequency reuse. Since the full reuse implementation is still needed at high traffic demand, we proposed a controller that switches between full reuse and bandwidth sharing implementations of the VeSn feature. For ViSn, we proposed bandwidth sharing algorithms and compared this implementation with a full bandwidth reuse. Finally we presented an antenna array design methodology and a beam codebook construction strategy for a multilevel beamforming. We also proposed an algorithm to efficiently navigate through the beam codebook (designed hierarchically) to select the best beam for each user. Extensive numerical results have been provided to support the attractiveness of the proposed methodologies and algorithms. The results obtained show that AAS especially with antenna arrays are definitely one of the enablers of 5G mobile communications.

Chapter 5

Coordination of SON algorithms

“If you want to go quickly, go alone. If you want to go far, go together.”

– African proverb

5.1 Introduction

SON functions are often designed as stand alone functionalities, and when triggered simultaneously, their interactions can lead to unpredictable behavior. The deployment of multiple control loops raises the questions of conflicts, stability and performance. The topics of conflict resolution, coordination, and a framework for managing multiple SON functionalities are receiving a growing interest (see for example [30, 38, 65]). Most contributions that have addressed the coordination problem between specific SON functionalities, provide a solution implemented in a centralized [45, 50] or distributed [25] fashion. In a centralized solution, the SON coordination problem can be treated as a multi-objective optimization [81], [36]. From the standardization point of view, the coordination problem has been addressed as a centralized, management-plane problem [5].

Little material has been reported on distributed, control plane solutions for the coordination problem in spite of its higher reactivity, attractiveness from an architecture point of view and potential performance gain. The aim of this chapter is to provide a generic coordination mechanism which is practically implementable by extending the solution reported in [26] and its domain of application.

In Section 5.2 we state the proposed model for interaction of SON mechanisms running in parallel and the coordination problem to be solved. In Section 5.3, the coordination problem is formulated as a convex optimization problem which can be solved quickly with modern computers and the existence of fully distributed coordination is discussed along with the validity of the coordination mechanism proposed in a stochastic environment. In Section 5.4 we illustrate the application of our model to a use case of the coordination method in a LTE network with three different SON functionalities deployed in each BS. Section 5.5 concludes the chapter.

5.2 Problem Description

5.2.1 General Problem

Consider N autonomous mechanisms operating simultaneously, each represented by a control loop in which a parameter is updated in a direction that improves a certain objective e.g. a KPI. Denote by $g_i(\cdot)$ the objective function that the i th control loop is trying to minimize, and by $S_i, i \in \{1, \dots, N\}$ some intervals in \mathbb{R} . This control loop can be written as

$$\dot{\theta}_i = -\frac{\partial g_i(\theta)}{\partial \theta_i} \quad (5.1)$$

where $\theta_i \in S_i$ is the parameter tuned by this particular control loop, and $\theta \in \prod_{j=1}^N S_j$ is the vector of all parameters tuned by the control loops. The sets $S_i, i \in \{1, \dots, N\}$ represent the sets of acceptable values of the different parameters.

This system of control loops is equivalent to a N -person game in which each agent tries to reach its own goal. This goal could be to maximize a payoff function (i.e. $-g_i$). The goal could

also be to force back a KPI to a certain target whenever it deviates from it, then g_i is the square distance between the KPI and its target.

It is assumed that the equilibrium point say $\theta_i^* \in S_i$ of the ODE (5.1) is asymptotically stable. By denoting $f_i(\theta) = -\frac{\partial g_i(\theta)}{\partial \theta_i}$, the assumption is verified when f_i is locally Lipschitz and strictly decreasing in θ_i . In this case, $V(\theta_i) = (f_i(\theta_i) - f_i(\theta_i^*))^2$ can act as a Lyapunov function for Equation (5.1) in the neighborhood of θ_i^* (see [48] for more details on Lyapunov Stability).

The simultaneous operation of all the control loops yields a system of autonomous ODEs written as

$$\dot{\theta} = F(\theta) \quad (5.2)$$

where $\theta = [\theta_1, \dots, \theta_N]^T$ and $F(\theta) = [f_1(\theta), \dots, f_N(\theta)]^T$. Due to the fact that the objective of each agent may be influenced by other parameters, the question of parallel stability arises even if the mechanisms are standalone stable. Denote the Jacobian of F as follows

$$J_F : \theta \longmapsto J_F(\theta) \quad (5.3)$$

where $\theta \in \prod_{j=1}^N S_j$ and $J_F(\theta) \in \mathbb{R}^{N \times N}$ is defined as $J_F(\theta)_{(i,j)} = \frac{\partial f_i(\theta)}{\partial \theta_j}$. A sufficient condition for parallel stability of system (5.2) is recalled in the following theorem.

Theorem 9. *If the matrix $J_F(\theta) + J_F(\theta)^T$ is negative definite for every θ in $\prod_{j=1}^N S_j$, then (5.2) has a unique equilibrium point and it is asymptotically stable in $\prod_{j=1}^N S_j$.*

Proof. If $J_F(\theta) + J_F(\theta)^T$ is negative definite, then the system of ODEs is diagonally strictly concave (see Section 2.6 for a definition). The result is then obtained using [63, Theorem 9]. \square

5.2.2 Linear Case

In order to further characterize the stability of the system of autonomous mechanisms, we restrict ourselves to the linear case. Assume that each function f_i can be written as a linear function of the parameters. So $F(\cdot)$ is expressed as

$$F(\theta) = A\theta \quad (5.4)$$

where $A \in \mathbb{R}^{N \times N}$. Note that this also includes the affine functions $A\theta + b$ which are reduced to linear functions by a change of variables applied to the parameters. This requires that the equilibrium point be feasible which means that A is invertible and $\theta^* = A^{-1}b$ is in $\prod_{j=1}^N S_j$. Indeed in this case one can choose a new variable $\phi = \theta - \theta^*$ and obtain the equivalent system

$$\dot{\phi} = \dot{\theta} = A(\theta - \theta^*) \quad (5.5)$$

where t is the time variable. Linearization is an acceptable approximation for control problems where the domain is a small neighbourhood of the equilibrium point (see Hartman–Grobman theorem [22]).

The Jacobian defined in Equation (5.3) now reduces to a constant $N \times N$ real matrix A , and the Rosen stability condition used in Theorem 9 reduces to $A + A^T$ being a negative-definite matrix. In fact, in the linear case the system is stable if and only if all eigenvalues of A lie in the Left Half Plane (LHP). This is equivalent to the existence of a symmetric definite positive matrix X such that $A^T X + XA < 0$. The assumptions made for standalone mechanisms clearly do not ensure that the negative definite condition be verified, thus a coordination mechanism may be needed.

In the following, it will be assumed that the control loops involved in the coordination problem are linear or linearizable.

5.3 Coordination Mechanism

5.3.1 ODE stabilization

A solution to the coordination problem exposed in the previous section has been proposed in [26]. The main idea of the method is to apply a coordination matrix denoted as C and use $CF(\theta)$ instead of $F(\theta)$ in the Right Hand Side (RHS) of Equation(5.2). Then we get the new system

$$\dot{\theta} = CF(\theta) = CA\theta \quad (5.6)$$

where C is chosen appropriately (i.e. to ensure convergence and maximize convergence speed) taking system constraints into account.

We now consider the problem of deriving a coordination matrix C such that Equation (5.6) is stable while Equation (5.5) is not. Stability is achieved if and only if there exists a symmetric matrix X such that:

$$(CA)^T X + XCA < 0, \quad X > 0, \quad (5.7)$$

where $X > 0$ denotes that X is positive definite.

A sufficient condition for stability is given in the following theorem.

Theorem 10. *Suppose there exists a $N \times N$ matrix C verifying*

$$(CA)^T + CA < 0, \quad (5.8)$$

then (5.6) is globally asymptotically stable.

Proof. See Theorem 9. □

In addition to the constraint (5.8), we need to consider an additional constraint which is related to the capability of the different SON entities to exchange information. For example, if two SON functions i and j are located in different BSs of a LTE network without a X2 interface between them, then the element $C_{i,j}$ in matrix C must be equal to 0. On the other hand, if $C_{i,j} \neq 0$, then updating the parameter θ_i requires the value of $F_j(\theta)$, so we have to be sure that this

information can be made available. Typically in a network for example, this relates to interfaces that exist between network entities (e.g. BSs), so the system constraints will be mapped from the network architecture. We denote by $\mathcal{C} \subseteq \mathbb{R}^{N \times N}$ the set of feasible matrices which satisfy the system constraints.

Denote the two constraints mentioned above as *stability* and *connectivity* constraints. These two constraints may be verified by a large number of matrices, and the one with the best convergence properties is sought. From standard convex optimization results, we know that iterative algorithms converge faster when their condition number is lower [60]. Indeed, the solution of the system of ODEs $\dot{x} = CAx$ can be written as $x(t) = e^{tCA}x_0$. The exponential of a matrix is defined using the power series, so

$$x(t) = \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} (CA)^k \right) x_0.$$

If we choose x_0 as an eigenvector of CA with the eigenvalue λ_0 , we can see that

$$x(t) = \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} \lambda_0^k \right) x_0 = e^{\lambda_0 t} x_0.$$

The same argument is valid for all the eigenvectors of the matrix CA so that for a random starting point x_0 , a lower condition number (ratio between the absolute values of the largest and the smallest eigenvalues) will ensure a better convergence as the speed of convergence will be homogeneous across all the eigenspaces.

Without constraints, the best coordination matrix would be $-A^{-1}$, leading to a diagonal matrix $CA = -I$ with the lowest condition number i.e. 1. When taking the constraints into account, we formulate the convex optimization problem as the minimization of the distance, defined in terms of the Frobenius norm, between the coordination matrix C and $-A^{-1}$:

$$\begin{aligned} & \text{minimize } \|C + A^{-1}\|_F \\ & \text{s.t. } (CA)^T + CA \prec 0; C \in \mathcal{C} \end{aligned} \tag{5.9}$$

where $\|\cdot\|_F$ is the Frobenius norm defined for a $\mathbb{R}^{m \times n}$ matrix M as

$$\|M\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |M_{i,j}|^2} = \sqrt{\text{Tr}(M^T M)} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2} \tag{5.10}$$

with σ_i being the singular values of M . It is noted that the Frobenius norm is often used in the literature for finding a preconditioner that improves the convergence behavior of iterative inversion algorithms [34].

The stability constraints are expressed in the form of Linear Matrix Inequalities (LMIs). LMIs are a common tool used in control theory for assessing stability. Solving convex optimization problems with LMI constraints is a tractable problem for which fast solvers are available [16].

From the implementation point of view, the coordination process can be performed in two steps as follows. In the first step, a centralized coordination block located in the management plane gathers and processes data to derive the matrix A , and performs the optimization problem (5.9) to obtain the coordination matrix C . Once C is available, each of its lines is sent to the corresponding SON function. This step is performed off-line and can be viewed as a feed-forward control since the coordination matrix is generally evaluated once or updated at very large intervals of time.

The second step is the on-line control process where each SON function performs the coordinated control, while satisfying the connectivity constraints, by using the appropriate line of matrix C . The control for SON function i changes from $\dot{\theta}_i = F_i(\theta)$ to $\dot{\theta}_i = \sum_{k=1}^N C_{i,k} F_k(\theta)$. The SON block remains a feedback loop by updating parameters according to measured KPIs, but uses the coordination matrix in these updates as described in Figure 5.1.

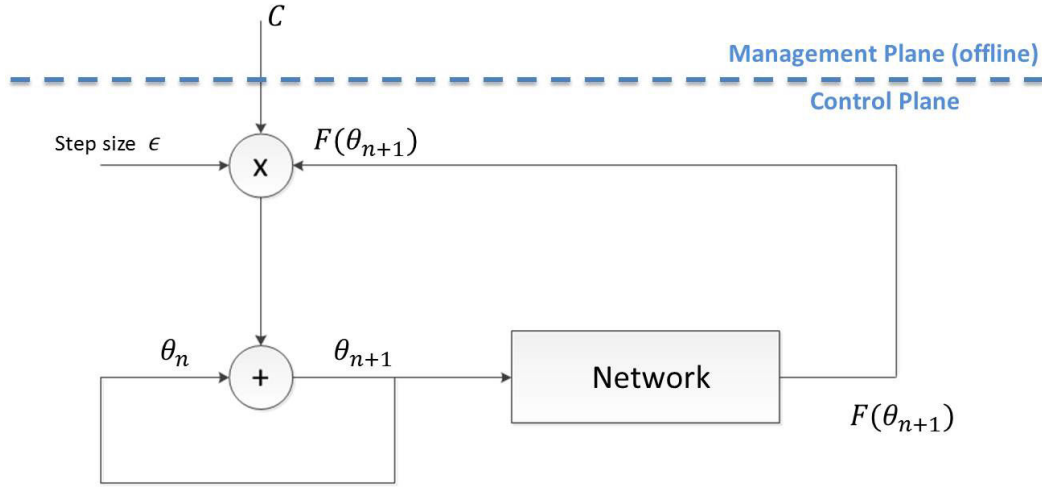


Figure 5.1: Coordination system block diagram

5.3.2 Fully distributed coordination

In this section we discuss the possibility of fully distributed coordination, where the coordination matrix C is *diagonal*. As said previously, if $C_{i,j} \neq 0$, $i \neq j$ then SON functions i and j need to exchange information. In fully distributed coordination, no information is exchanged. Sufficient conditions for the existence of a diagonal matrix can be found in the literature. In particular Fisher and Fuller (1958) [29] have proven that if there exists a permutation matrix P such that all leading principal sub-matrices of $\hat{A} = PAP^{-1}$ are of full rank, then A can be stabilized by scaling.

A more restrictive version of this condition which gives a simple way to construct the coordination matrix is given in the following theorem.

Theorem 11. *If all leading principal sub-matrices of A are of full rank, then there exists a diagonal matrix $C = \text{diag}(c_1, c_2, \dots, c_N) \in \mathbb{R}^{N \times N}$ that stabilizes A (i.e. CA is a Hurwitz matrix).*

Proof. Indeed, it then suffices to choose c_1, c_2, \dots, c_N sequentially such that

$$(-1)^i c_1 \dots c_i \det([A]_{i,i}) > 0 \quad (5.11)$$

for $i = 1, \dots, I$ where $[A]_{i,i}$ is the submatrix of A comprised of lines 1 through i and columns 1 through i . This means that $\forall k = 1..I$, $(-1)^k \det([CA]_{k,k}) > 0$ which implies by a known result [67, Section 16.7] on negative definite matrices that all eigenvalues of $CA + (CA)^T$ are strictly negative. \square

Later works have extended the Fisher and Fuller condition to more general cases [64].

5.3.3 Stochastic Control Stabilization

In practical systems, ODEs are replaced by SA algorithms. Indeed, the variables are updated at discrete times proportionally to functions values which are noisy.

The noise in the function values is due to the fact that time is slotted and functions are estimated by averaging certain counters during a time slot. For example, the load of a BS in a mobile network is often estimated by evaluating the proportion of time during which the scheduler is busy, and the file transfer time is estimated by averaging the file transfer times of all flows occurring in a certain period of time. The noise is also due to intrinsic stochastic nature of real systems, for example in wireless networks the propagation environment is inherently non-deterministic (because of fading, mobility, etc.) so the SINR will be noisy.

When the noise in the measurements of the function values is of Martingale difference type (see Section 2.2.4 for basic definition of Martingales), the mean behavior of those SA algorithms matches with the system of ODEs. Note that we consider Martingale difference type of noise but the SA results hold for much more general noise processes (stationary, ergodic). In [25] for example, SA results are used without the Martingale difference property.

The initial system of control loops is actually a system of SA algorithms, with one of them written as

$$\theta_i[k+1] = [\theta_i[k] + \epsilon_k(f_i(\theta[k]) + N_k^i)]_{S_i}^+ \quad (5.12)$$

where $[\cdot]_{S_i}^+$ is the projection on the interval $S_i = [a_i, b_i]$; $a_i < b_i \in \mathbb{R}$, $\theta[k] = (\theta_1[k], \dots, \theta_I[k])$ is the vector of parameters after the $(k-1)$ th update, ϵ_k - the step size of the k th update and N_k^i represents the noise in the measurement.

The projection in (5.12) aims at ensuring that the iterates are bounded which is also a condition for convergence of the SA algorithm towards the invariant sets of the equivalent ODE.

Many SON algorithms are or can be reduced to the form of (5.12). For example in [23], a load balancing SON function is presented in this very same form. In [25] relays are self-optimized using also a SA algorithm. In [85], SA algorithms are used for self-optimizing interference management for femtocells. A handover optimization SON function which can be rewritten as an SA algorithm is also presented in [80].

We suppose that N_k is a Martingale difference sequence that meets the conditions for stand alone convergence (see [15, 48]). Namely the SA algorithms have the same behavior as their equivalent ODE. Now we want to check if the conditions for the SA equivalence with the limiting ODE are still verified after the coordination mechanism is applied. The coordinated SA for the i -th mechanism is

$$\theta_i[k+1] = \left[\theta_i[k] + \epsilon_k \left(\sum_{j=1}^I C_{i,j} (f_j(\theta[k]) + N_k^j) \right) \right]_{S_i}^+ \quad (5.13)$$

The projection ensures that the iterates are bounded. The next step is to show that $\sum_{j=1}^I C_{i,j} N_k^j$ is a Martingale difference sequence in order to meet the convergence conditions. Denoting $\mathcal{F}_k = \left\{ \sum_{j=1}^I C_{i,j} N_l^j, l < k \right\}$, we have

$$E \left[\sum_{j=1}^I C_{i,j} N_k^j | \mathcal{F}_k \right] = \sum_{j=1}^I C_{i,j} E \left[N_k^j | \mathcal{F}_k \right] = 0$$

since $E[N_k^j | \mathcal{F}_k] = 0, j = 1 \dots I$. So the Martingale difference noise condition is satisfied ensuring the validity of the coordination method in a stochastic environment.

5.4 SON Coordination use case: Application to wireless networks

In this section we illustrate instability and coordination in the context of RANs for a use case involving 3 SON functions deployed in several BSs of a LTE network.

5.4.1 System Model

Consider three SON mechanisms deployed in the BSs of a LTE network: blocking rate minimization, outage probability minimization and load balancing. We focus on downlink File Transfer Protocol (FTP) type traffic model in which each user enters the network, downloads a file from its serving cell and then leaves the network.

Load balancing The SON function adjusts the pilot powers of I BSs in order to balance the loads between neighboring cells. The corresponding ODE is given by

$$\dot{P}_s = P_s(\rho_1(\mathbf{P}) - \rho_s(\mathbf{P})), \forall s = 1 \dots I \quad (5.14)$$

where $\mathbf{P} \in \mathbb{R}^I$ is the vector of BSs' pilot powers, and ρ - their corresponding loads. ρ_1 is the load of the reference BS which can be chosen as the most loaded one. This SON converges to a set on which all loads are equal as shown in [23, Theorem 4]. We use an equivalent formulation in order

to update the pilots directly in decibels (dB):

$$\dot{P}_{\text{dB}_s} = \rho_1(\mathbf{P}) - \rho_s(\mathbf{P}) \quad \forall s = 1 \dots I. \quad (5.15)$$

where P_{dB_s} is the pilot power of BS s in dB.

Blocking rate minimization This SON function adjusts the admission threshold in order to reach a given blocking rate target $\bar{B} > 0$. Consider $x_s \in \mathbb{R}^+$ such that the admission threshold of BS s is $\lfloor x_s \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. A new user finding the cell with n users is blocked with probability $P(n)$, where $P(n) \rightarrow 1$ when $n \rightarrow x_s$ and $P(n) \rightarrow 0$ when $n \rightarrow 0$. The update equation for the blocking rate minimization SON is

$$x_{s,t+1} = [x_{s,t} + \epsilon_t (B_s(x_t) - \bar{B}_s + N_t)]_{[0, x_{\max}]}^+ \quad (5.16)$$

where x_t is the vector of the admission thresholds of all the BSs considered at time t , x_{\max} - a sufficiently large value and N_t - a Martingale difference noise introduced by measuring $B(x_t)$. The equivalent ODE is

$$\dot{x}_s = B_s(x) - \bar{B}_s. \quad (5.17)$$

$x_s \rightarrow B_s(x)$ is a decreasing function of x_s and $\lim_{x_s \rightarrow \infty} B_s(x) = 0$. So for any blocking rate target $0 < \bar{B}_s < 1$, we have $B_s(0) \geq \bar{B}_s$ and there exists a finite $x_0 \in \mathbb{N}$ such that $\forall x \geq x_0; B_s(x) \leq \bar{B}$ and $\forall x \leq x_0; B(x) \geq \bar{B}$. By projecting the RHS of Equation (5.16) on any interval containing $[0, x_0]$, we ensure that $\sup_t \|x_t\| < \infty$.

Now considering the function $V(x) = \max(0, |x - x_0| - \delta)$ for δ sufficiently small, we can see that $V(\cdot)$ is a Lyapunov function for (5.17). Indeed, we have

- $\forall x \in [0, +\infty), V(x) \geq 0$.
- $H = \{x \in [0, +\infty), V(x) = 0\} \neq \emptyset$ because it contains x_0 .
- $\dot{V}(x) = \frac{\partial V}{\partial x} \dot{x}$

$$= \begin{cases} -(B(x) - \bar{B}) & \text{if } x < x_0 - \delta \\ B(x) - \bar{B} & \text{if } x > x_0 + \delta \\ 0 & \text{if } x \in [x_0 - \delta, x_0 + \delta] \end{cases}$$

$$\leq 0.$$
- $V(x) \rightarrow +\infty$ when $x \rightarrow +\infty$.

This implies that H is globally asymptotically stable for (5.17).

Outage Probability Minimization The aim of this SON mechanism is to adjust the transmit data power in order to reach a target outage probability. The outage probability considered is

expressed as

$$K_s = \frac{1}{|Z_s|} \int_{Z_s} \mathbb{1}_{\{R_s(r) \geq R_{min}\}}(r) dr \quad (5.18)$$

where Z_s is the area covered by BS s , R_{min} - a minimum data rate and $R_s(r)$ - the peak data rate obtained at position r when served by BS s . The SA algorithm modeling the actual control loop is

$$D_s[k+1] = D_s[k] - \epsilon_k (K_s(D[k]) - \bar{K} + N_k^s) \quad (5.19)$$

where N_k^s is a Martingale difference noise and D_s is the transmit data power of BS s . The limiting ODE representing the mean behaviour of SA (5.19) is then

$$\dot{D}_s = -(K_s(D) - \bar{K}). \quad (5.20)$$

This ODE is stable if there exists an admissible data power D_s^* such that $K_s(D_s^*) = \bar{D}_s$. Indeed, $(K_s(.) - \bar{K})^2$ would then be a Lyapunov function for (5.20) since $\frac{\partial K_s}{\partial D_s} > 0$. As a consequence, the SA (5.19) converges to invariant sets of (5.20), which means that the mechanism is standalone-stable.

5.4.2 Numerical Results

Consider a hexagonal network with 19 cells with omni-directional antennas as shown in Figure 5.2. A wrap-around model is used to minimize truncation effects of the computational domain. It is achieved by surrounding the original network with 6 of its copies while performing the simulation within the original 19 cells.

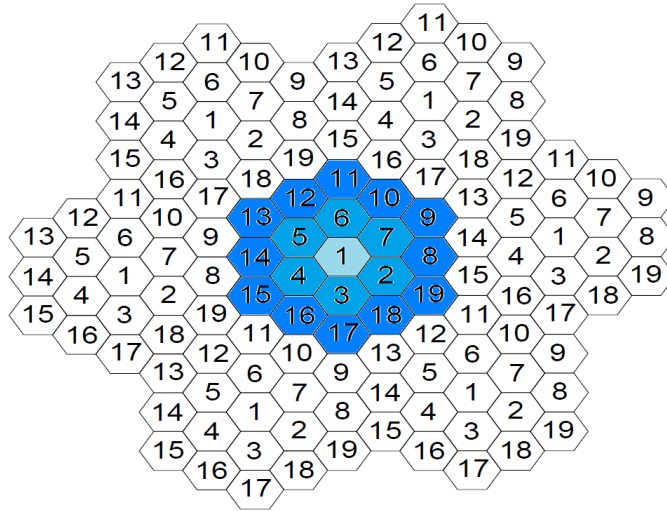


Figure 5.2: 19 cells hexagonal network with wrap-around

Table 5.1 lists the parameters used in the simulations including the environment, the network, the numerical simulation parameters and the KPIs' targets used by the SON mechanisms. The

Table 5.1: Network and Traffic characteristics

Network parameters	
Number of stations	19 (with wrap-around)
Cell layout	hexagonal omni
Intersite distance	500 m
Bandwidth	20MHz
Channel characteristics	
Thermal noise	-174 dBm/Hz
Path loss (d in km)	$128 + 36.4 \log_{10}(d)$ dB
Traffic characteristics	
Arrival rate	40 users/s
Service type	FTP
Average file size	10 Mbits
Hotspot additional arrival rate	2 users/s
Hotspot position	center of BS 1 cell
Hotspot diameter	330 m
Simulation parameters	
Spatial resolution	20 m x 20 m
Time per iteration	6 s
Minimum SINR for coverage	0 dB
Target outage probability	18%
Target blocking rate	2%

users arrive in the network according to a Poisson process with a certain arrival rate given in Table 5.1. A hotspot is placed at the center of the network with an additional arrival rate also given in Table 5.1. The hotspot provides initially unbalanced loads in the network, which is of interest for the load balancing SON function. The target outage probability is set to 18% in order to be feasible and according to the minimum SINR for coverage.

We activate the three SON functions in each of the 7 BSs located at the center of the map and observe the stability of the SON functions, with and without the coordination mechanism.

We first derive the matrix A using finite differences to compute the derivatives of the KPIs which are evaluated with closed form formulas. By choosing an adequate step size, this method yields very accurate results. However, closed-form expressions of the KPIs do not always exist, in which case estimations of the KPIs would be used instead, based on measurements from each user that arrive in the network. The stability matrix obtained through linearization already reveals instability since not all of its eigenvalues are negative, and hence the coordination step is inevitable.

We then derive the coordination matrix C by numerically solving Problem (5.9) using CVX, a package for specifying and solving convex programs [33]. Finally coordination is applied using $\dot{\theta} = CF(\theta)$.

We plot the KPIs evolutions of the coordinated (in blue) and non-coordinated (in red) systems (Figures 5.3, 5.4 and 5.5) for 3 representative BSs: the most loaded one (BS 1) and two of its neighbours (BSs 2 and 3). The coordinated system clearly performs better. The loads and the blocking rates are lower. The objectives related to each SON function are satisfied or close to being satisfied in the coordinated system. The loads are balanced in about 10 minutes. These results illustrate the usefulness of the distributed SON that benefits from much higher reactivity with respect to a centralized solution.

On the other hand, the outage probabilities in the non-coordinated system diverge. The outage probability of the initially most loaded BS (BS 1) is near zero while that of the other BSs is close to one. This is because the decrease in the cell size of the most loaded BS is not accompanied by a decrease of its transmit power. As a result, more interference is produced on its neighbors which have increased their cell size.

In Figure 5.4, we can see that the outage probability of BS 1 in the coordinated system is low but is off the target set to 18 percent. This is a consequence of the interaction of several SON functions. The operation point defined by the average KPIs can be modified using weights. Each F_i (corresponding to a distinct SON function) is multiplied by a given weight. We now investigate the impact of such weights on the stationary KPIs of the system.

Figures 5.6 and 5.7 compare the final values of the KPIs of the coordinated (in blue) and non-coordinated (in red) systems when they reach their steady state for different weight vectors. For equal weight across all SON functions, we can see in Figure 5.6 that the self-organized system balances the loads better than reaching the outage probability target. A closer look at BS1 shows that the power of its traffic channels increases to absorb more traffic while its cell size is reduced leading to a smaller outage probability.

Figure 5.7 considers the case where more importance is given to the outage probability, by increasing 20 times the corresponding weight. The outage probability for the BSs is practically the same as the target, while the loads are not balanced anymore. We can see that the coordination mechanism reaches a compromise between the different objectives, that can be adjusted by assigning weights to the SON functions to better reflect the network operator policies.

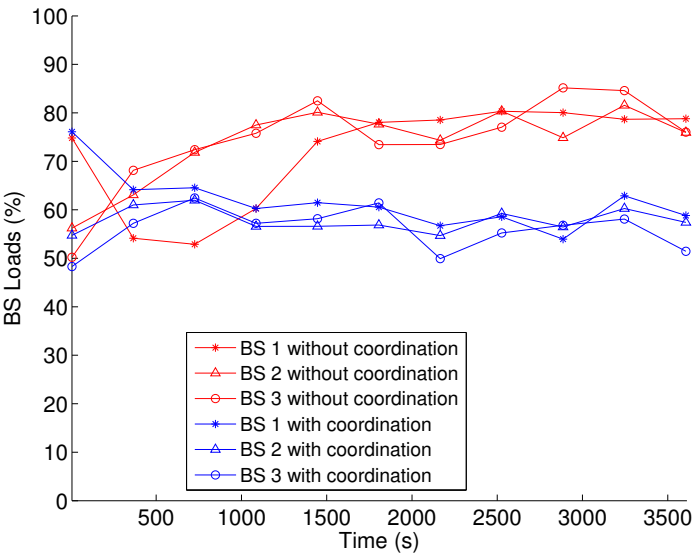


Figure 5.3: Impact of Coordination on Loads

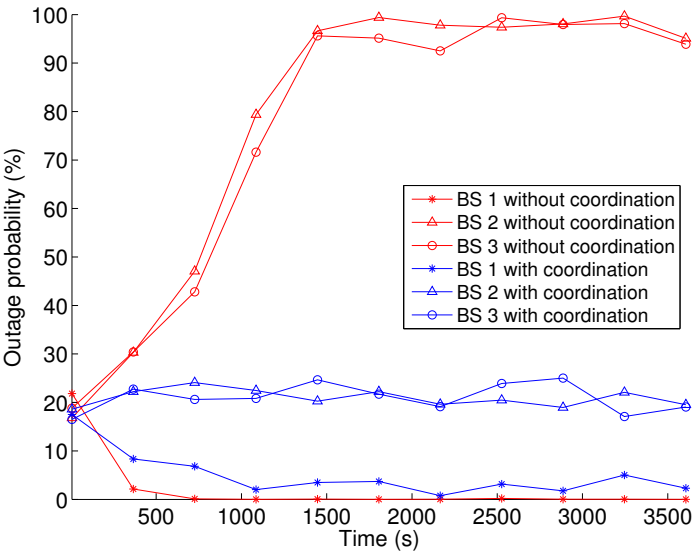


Figure 5.4: Impact of Coordination on Coverage Probability

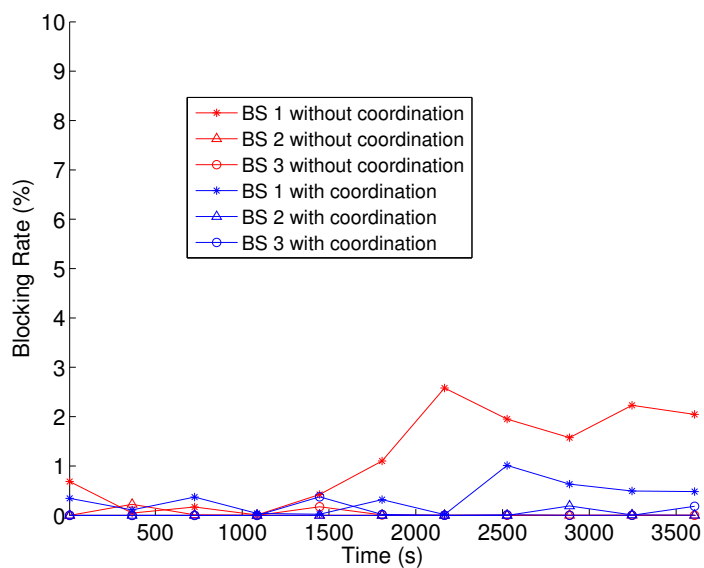


Figure 5.5: Impact of Coordination on Blocking Rate

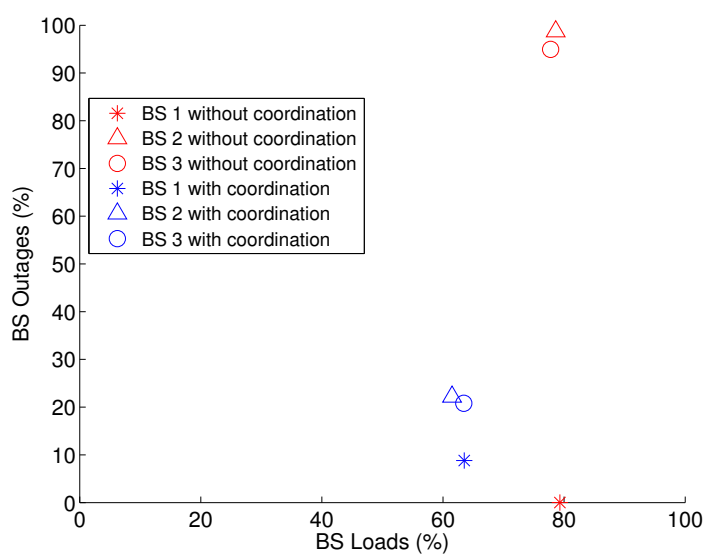


Figure 5.6: Stationary KPIs with all SON functions equally weighted

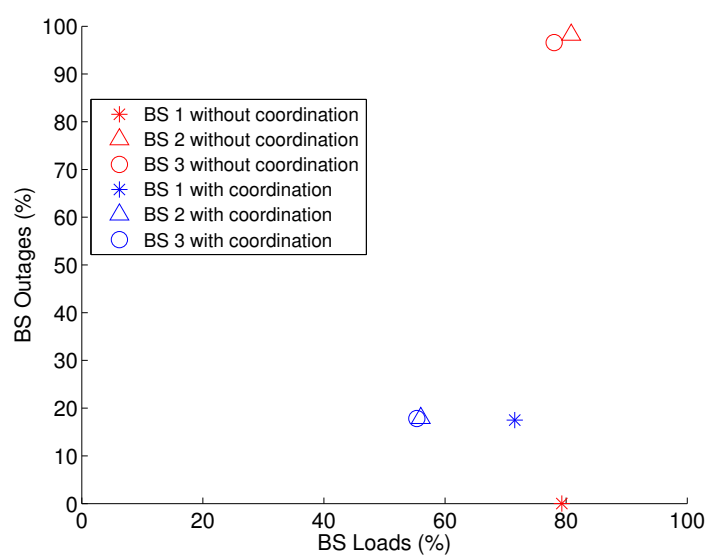


Figure 5.7: Stationary KPIs with outage probability prioritized

5.5 Conclusion

In this chapter, we have studied the problem of coordinating multiple SON entities operating in parallel. Using tools from control theory and Lyapunov stability, we have proposed a coordination mechanism that stabilizes the system. The problem of finding a coordination matrix has been formulated as a convex optimization problem with LMI constraints which ensures that the system of SON functions remain distributed. We have also shown that the coordination solution remains valid in the presence of measurement noise, using SA. A practical use case of the coordination method has been presented in a LTE network implementing three distributed SON functions deployed in several base stations. It has been shown that the coordination mechanism is necessary to stabilize the network. This use case has also shown that in spite of the linear control assumption, the method remains effective when applied to SON functionalities that are not linear in general.

It is noted that the coordination framework in this chapter has been applied to the HetNets use case where load balancing and interference coordination algorithms were run in parallel. The stability matrix computed for this use case was already negative definite, so there was no need for coordination.

Chapter 6

Conclusion

This chapter summarizes the contribution of the thesis and discusses avenues for future research.

6.1 Summary of contribution

In this thesis, we studied several scenarios in mobile networks and provided SON algorithms aiming at either improving user experience or making network management easier for the operator. We also proposed a generic methodology for the difficult problem of SON coordination. The scenarios studied were related to network densification.

The first densification scenario is the use of small cells in order to offload congested macro cells. We proposed SON algorithms for load balancing taking into account backhaul capacity and also for interference coordination in HetNets using ABS (time-domain mitigation) or frequency splitting (frequency-domain mitigation). Our work demonstrated the necessity of dynamic load balancing and interference mitigation schemes for a successful deployment of the small cells technology which is promising for solving the capacity issues. It has also been shown that backhaul must be taken into consideration in SON algorithms design. The implementation of some of the proposed algorithms in a AutoSDN demonstrator was also described showing a flexible way in which SON algorithms can be easily deployed in mobile networks.

The second densification scenario made use of AASs in order to create additional cells by focusing the beam towards a given area. In this context, we successively studied VeSn for a vertical separation of the beams, ViSn allowing to create the new cell anywhere in the given area and finally Massive MIMO beamforming which is able to focus the beam directly towards each user. SON algorithms for optimal feature activation, switching between different implementations (i.e. with full reuse or with frequency splitting), interference coordination and multilevel beam focusing were studied. The VeSn and ViSn can be viewed as cheap solutions for short-term network densification with no expensive backhaul deployment needed. Multilevel beamforming on the other hand represents one of the most interesting enablers for 5G technology helping to solve some of the most difficult 5G scenarios such as massive events or energy saving.

We next proposed a generic method that allows to detect instability in a system of concurrently operating SON functions and coordinate them. This methodology is based on concave games theory and convex optimization with LMIs. A LTE scenario with three different SON algorithms was simulated to numerically support the efficiency of the proposed methodology. This scenario was constructed to be unstable using non-standard SON algorithms. We have found that in general, the standardized SON algorithms are not unstable when operating simultaneously since either they do not have conflicting objectives or their design takes into consideration obvious conflicts that may appear. Nonetheless, our work provides a framework for the coordination of operator-made and non-standardized SON algorithms.

6.2 Directions for future work

Only some use cases have been developed during this thesis due to the limited time. We only consider elastic traffic in most of the use cases, so the algorithms could be extended to support other types of traffic such as GBR or services such as video streaming. The generalization to other types of traffic will not only impact the definition of some KPIs such as the load but entirely change the types of KPIs that should be considered. For example, the buffer starvation probability is a more relevant KPI for video streaming than the MUT.

The backhaul-constrained load balancing use case is only one example where the backhaul can impact the performance of a SON algorithm. Other SON algorithms such as CCO should be studied to possibly detect and correct problems related to backhaul. We also considered only the last segment of the backhaul just behind the BS. The full backhaul should be studied in order to find a unified global view of the performance of SON algorithms and improve their efficiency.

The numerical results provided in Chapter 3 showed that even using a lower bound to the exact utility function, the performance of the system can be improved. As mentioned before, this is due to the impact of the resource allocation algorithm on the trajectory of the system in its state space (number of users and their locations at any given time). An interesting way forward is to investigate this phenomenon further in order to better understand it and possibly find recommendations for the choice of the utility function which is equivalent to the choice of α in α -fair utilities. An interesting tool for this study could be the Discriminatory Processor Sharing (DPS) [6].

We provided an analytical development in order to design the activation algorithm for the VeSn feature. This choice limited us to consider only the MUT in the system in order to be able to derive a simple activation algorithm. Other KPIs can be considered if a more generic approach such as a learning algorithm is used. Multi-Armed-Bandit (MAB) is one possible way which meets the requirements for our SON algorithms, namely simplicity, robustness and fast convergence. Actually, MAB has been shown to provide the optimal convergence rate in some cases.

For the coordination problem, we focused in this thesis on linear or linearizable systems of SON algorithms. More general systems of SON algorithms can be considered where the relationship

between the KPIs and the parameters is highly non linear.

The general methodology and tools used throughout the thesis could be used to develop more SON algorithms for more scenarios. There is for example resource allocation in a Device-To-Device (D2D) scenario or the efficient implementation of sleep mode with SON in an energy saving scenario.

The algorithm proposed for multilevel beamforming is at the stage of an idea. More work is needed in order to make it fully usable among which the automatic generation of the multilevel codebook based on simulated coverage areas for example. Different optimization approaches could also be studied for the antenna optimization problem. The actual technological implementation of the approach could also present some challenges that need to be addressed such as the reconfigurability of antenna elements amplitudes and phases.

Appendix A

Theorem proofs for chapter 3

A.1 Proof of Theorem 2

Suppose that the ABSr θ is bounded away from 0 and 1 ($\theta \in (0, 1)$). Firstly, we can note that U_{PF1} is differentiable on $(0, 1)$ and let us evaluate its derivative. Using the properties of the log function ($\log(ab) = \log a + \log b$), we can rewrite $U_{PF1}(\theta)$ as

$$U_{PF1}(\theta) = \sum_{m=1}^M N_m \log(1 - \theta) + N_{p,CEN} \log(1 - \theta) + N_{p,CRE} \log(\theta) + C_1 \quad (\text{A.1})$$

where

$$C_1 = \sum_{m=1}^M \sum_{u \in m} \log(\bar{R}_{u,m}) + \sum_{u \in \text{center of } p} \log(\bar{R}_{u,p}^{\text{no ABS}}) + \sum_{u \in \text{CRE of } p} \log(\bar{R}_{u,p}^{\text{ABS}})$$

is independent of θ . From (A.1), we can easily derive

$$\frac{\partial U_{PF1}(\theta)}{\partial \theta} = \frac{N_{p,CRE}}{\theta} - \frac{N_{p,CEN} + \sum_{m=1}^M N_m}{1 - \theta} \quad (\text{A.2})$$

Using SA results (see Section 2.5), we can see that the equivalent ODE to (3.23) is

$$\dot{\theta} = \frac{\partial U_{PF1}(\theta)}{\partial \theta} \quad (\text{A.3})$$

Since $U_{PF1}(\theta)$ is the sum of the log of concave positive functions, it is concave. So the solution of (A.3) converges towards the maximum of $U_{PF1}(\theta)$.

A.2 Proof of Theorem 3

The proof is similar to that of Theorem 2. Except now $U_{\text{PF2_exact}}(\theta)$ is be rewritten as

$$\begin{aligned} U_{\text{PF2_exact}}(\theta) = & \sum_{u \in p} \log((1 - \theta)\bar{R}_{u,p}^{\text{no ABS}} + \theta\bar{R}_{u,p}^{\text{ABS}}) \\ & + \sum_{m=1}^M N_m \log(1 - \theta) + C_2 \end{aligned} \quad (\text{A.4})$$

where $C_2 = \sum_{m=1}^M \sum_{u \in m} \log(\bar{R}_{u,m})$ is independent of θ .

Function (A.4) is also a concave function of θ and its derivative reads

$$\frac{\partial U_{\text{PF2_exact}}(\theta)}{\partial \theta} = \sum_{u \in p} \frac{1}{\theta + \frac{\bar{R}_{u,p}^{\text{no ABS}}}{\bar{R}_{u,p}^{\text{ABS}} - \bar{R}_{u,p}^{\text{no ABS}}}} - \frac{\sum_{m=1}^M N_m}{1 - \theta} \quad (\text{A.5})$$

The rest of the proof follows from Appendix A.1.

A.3 Proof of Theorem 4

The proof is also similar to that of Theorem 2.

A.4 Proof of Theorem 5

We first treat the case of $\alpha = 0$. Since there is no singularity problem at $\theta = 0$ or $\theta = 1$, we can perform the maximization on $[0, 1]$. Since $\bar{R}_{u,m}$, $\bar{R}_{u,p}^{\text{no ABS}}$ and $\bar{R}_{u,p}^{\text{ABS}}$ do not depend on θ , the utility $U_\alpha(\theta)$ is linear in θ when $\alpha = 0$. So the maximum is attained at one of the ends of the interval $[0, 1]$: 1 if the function $U_\alpha(\theta)$ is increasing in θ and 0 otherwise.

For the cases $\alpha > 0$, we start by proving that $U_\alpha(\theta)$ is concave in $\theta \in [0, 1]$. To that end, we compute the second derivatives $U_\alpha(\theta)$ with respect to θ . The first derivatives are given in Equations (3.34) and (3.35). Deriving these expressions once again, we obtain

- For $\alpha = 1$

$$\frac{\partial^2 U_\alpha(\theta)}{\partial \theta^2} = - \sum_{p=1}^P \sum_{u \in p} \frac{1}{\left(\theta + \frac{\bar{R}_{u,p}^{\text{no ABS}}}{\bar{R}_{u,p}^{\text{ABS}} - \bar{R}_{u,p}^{\text{no ABS}}} \right)^2} - \frac{\sum_{m=1}^M N_m}{(1 - \theta)^2} \quad (\text{A.6})$$

- For $\alpha \neq 1$

$$\begin{aligned} \frac{\partial^2 U_\alpha(\theta)}{\partial \theta^2} = & - \sum_{m=1}^M \sum_{u \in m} \alpha \bar{R}_{u,m}^{1-\alpha} (1-\theta)^{-\alpha-1} - \sum_{p=1}^P \sum_{u \in p} \\ & \left[\alpha (\bar{R}_{u,p}^{\text{ABS}} - \bar{R}_{u,p}^{\text{no ABS}})^2 \left((1-\theta) \bar{R}_{u,p}^{\text{no ABS}} + \theta \bar{R}_{u,p}^{\text{ABS}} \right)^{-\alpha-1} \right] \end{aligned} \quad (\text{A.7})$$

So we have $\frac{\partial^2 U_\alpha(\theta)}{\partial \theta^2} < 0 \forall \theta \in [0, 1)$. By the second-order conditions for convexity [17, §3.1.4], we can conclude that $U_\alpha(\theta)$ is strictly concave.

$U_\alpha(\theta)$ is then continuous, differentiable and concave in $\theta \in [0, 1)$. It is then easy to show that for $\alpha > 0$, the maximum of this function is attained at the value of θ^* where $\frac{\partial U_\alpha(\theta^*)}{\partial \theta} = 0$ using KKT conditions (see Section 2.4).

Solving the equation $\frac{\partial U_\alpha(\theta)}{\partial \theta} = 0$ for $\alpha > 0$ can be involved. Also, the parameters used in the optimization (users throughputs) vary randomly with time. So instead of computing the optimal ABSr with noisy data, we can use a SA algorithm (see Algorithm 3) to optimize iteratively the ABSr. So we fall in the framework of SA as described in Section 2.5. The equivalent ODE for that update equation is then

$$\dot{\theta} = \frac{\partial U_\alpha(\theta)}{\partial \theta} \quad (\text{A.8})$$

$U_\alpha(\theta)$ is concave as shown earlier, so the SA update equation for (A.8) converges towards the maximum of the α -fair utility functions ($\alpha > 0$).

A.5 Proof of Theorem 6

The proof will consist in showing that (3.39) is equivalent to a sub-gradient optimization method. We first show that $\frac{\partial \rho_{\max}(\theta_k)}{\partial \theta}$ is a sub-gradient of ρ . Recall that at the points where there is only one maximum among all loads considered, the overall load is differentiable, i.e. the sub-gradient reduces to the classic gradient. At inflexion points, the sub-gradient becomes a set of values which we can show contains the individual derivatives of all the loads that are maximum at that point. Indeed, $g(\theta_0)$ is a sub-gradient of ρ at θ_0 if

$$\rho(\theta) \geq \rho(\theta_0) + g(\theta_0)(\theta - \theta_0) \forall \theta \quad (\text{A.9})$$

So if we consider a particular load with index \max which is a maximum of all loads in our cluster, and denote its gradient at θ_0 by $g_{\max}(\theta_0)$, we would have

$$\rho(\theta) \geq \rho_{\max}(\theta) \geq \rho_{\max}(\theta_0) + g_{\max}(\theta_0)(\theta - \theta_0) \forall \theta \quad (\text{A.10})$$

But since $\rho_{\max}(\theta_0) = \rho(\theta_0)$, we can conclude that $g_{\max}(\theta_0)$ is a sub-gradient of ρ at θ_0 . The equivalent ODE for (3.39) as shown in [15, Theorem 2, Page 15], can then be written as

$$\dot{\theta} = -\frac{\partial \rho(\theta)}{\partial \theta} \quad (\text{A.11})$$

where $\frac{\partial \rho(\theta)}{\partial \theta}$ is any sub-gradient of ρ at θ . Now since ρ is a convex function as show in Section 3.3.3.1, (A.11) is a sub-gradient optimization algorithm. Convergence results for the sub-gradient method are available in [18]. Note that since this method is not a descent method, in practice we keep track of the best parameter θ^{BEST} obtained so far.

A.6 Proof of Theorem 7

We begin by proving that (3.46) and (3.47) are concave. To this end, we will use the following propositions from [17].

Proposition 1. *A function $f : \mathbb{R} \mapsto \mathbb{R}$ which is twice differentiable is concave if and only if its domain is convex and its second derivative is negative : $\forall x \in \text{dom} f, \frac{\partial^2 f(x)}{\partial x^2} \leq 0$.*

Proposition 2. *Let us consider two functions $h : \mathbb{R} \mapsto \mathbb{R}$ and $g : \mathbb{R} \mapsto \mathbb{R}$. Then $f = h \circ g : \mathbb{R} \mapsto \mathbb{R}$ defined as $f(x) = h(g(x))$ is concave if h is concave non-decreasing and g is concave.*

Proposition 3. *The non-negative weighted sum of concave functions is concave.*

Let a, b, c be strictly positive real constants. We begin by proving that the function

$$f(x) = x \log\left(1 + \frac{a}{bx + c}\right) \quad (\text{A.12})$$

is concave for $x \in [\tau, 1 - \tau]$ with τ a small constant. Note that this function is continuous and twice differentiable on $[\tau, 1 - \tau]$. Let us evaluate its second derivative.

$$\frac{\partial f(x)}{\partial x} = \log\left(1 + \frac{a}{bx + c}\right) - \frac{ba}{(bx + c)(bx + c + a)}$$

So

$$\begin{aligned} \frac{\partial^2 f(x)}{\partial x^2} &= \frac{-ba}{(bx + c)(bx + c + a)} - \frac{ba(bx + c)(bx + c + a)}{(bx + c)^2(bx + c + a)^2} \\ &\quad + \frac{bax[b(bx + c + a) + b(bx + c)]}{(bx + c)^2(bx + c + a)^2} \\ &= \frac{-(ba + 2bc)x - 2c(c + a)}{(bx + c)^2(bx + c + a)^2} \end{aligned}$$

Since $a, b, c \in \mathbb{R}^{++}$, we can say that $\frac{\partial^2 f(x)}{\partial x^2} < 0, \forall x \in [\tau, 1 - \tau]$. So the function f defined in (A.12) is concave by Proposition 1.

From Proposition 2, we can say that the functions $g_1 : x \mapsto \log(wf(x))$ and $g_\alpha : x \mapsto \frac{(wf(x))^{1-\alpha}}{1-\alpha}$ are concave because the log function and the function $\frac{x^{1-\alpha}}{1-\alpha}$ are concave and non-decreasing on \mathbb{R}^+ with w being a real positive number.

Note that the same conclusions hold for the function $g'_1 : x \mapsto \log(wf(1-x))$ and $g'_\alpha : x \mapsto \frac{(wf(1-x))^{1-\alpha}}{1-\alpha}$ since $\frac{\partial^2 f \circ (1-x)}{\partial x^2}(x) = \frac{\partial^2 f}{\partial x^2}(1-x)$ and $1-x$ is strictly positive on $[\tau, 1-\tau]$.

The function $U_\alpha(\delta)$ can be written as

- For $\alpha = 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \log \left((1-\delta) \log \left(1 + \frac{\xi_{u,m}}{(1-\delta)N_0 + \sum_{k \neq m} \xi_{u,k}} \right) \right) + \sum_{p=1}^P \sum_{u \in p} \log \left(\delta \log \left(1 + \frac{\xi_{u,p}}{\delta N_0 + \sum_{j \neq p} \xi_{u,j}} \right) \right) + C_3 \quad (\text{A.13})$$

where C_3 is a constant independent of δ .

- For $\alpha \neq 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \frac{\left((1-\delta) c_{u,m} \log \left(1 + \frac{\xi_{u,m}}{(1-\delta)N_0 + \sum_{k \neq m} \xi_{u,k}} \right) \right)^{1-\alpha}}{1-\alpha} + \sum_{p=1}^P \sum_{u \in p} \frac{\left(\delta c_{u,p} \log \left(1 + \frac{\xi_{u,p}}{\delta N_0 + \sum_{j \neq p} \xi_{u,j}} \right) \right)^{1-\alpha}}{1-\alpha} \quad (\text{A.14})$$

where $c_{u,m}$ and $c_{u,p}$ are some positive constants independent of δ .

Denoting by $f_{u,c}$ the function defined in (A.12) but with a, b and c replaced respectively by $\xi_{u,c}, N_0$ and $\sum_{l \neq c} \xi_{u,l}$, where $\xi_{u,c}$ is the received signal strength of user u attached to BS c , we can rewrite (A.13) and (A.14) as

- For $\alpha = 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \log(f_{u,m}(1-\delta)) + \sum_{p=1}^P \sum_{u \in p} \log(f_{u,p}(\delta)) + C_3 \quad (\text{A.15})$$

- For $\alpha \neq 1$:

$$U_\alpha(\delta) = \sum_{m=1}^M \sum_{u \in m} \frac{(c_{u,m} f_{u,m}(1-\delta))^{1-\alpha}}{1-\alpha} + \sum_{p=1}^P \sum_{u \in p} \frac{(c_{u,p} f_{u,p}(\delta))^{1-\alpha}}{1-\alpha} \quad (\text{A.16})$$

So using the preceding results and Proposition 3, and also by recalling that the minimum of two concave functions is also concave, we can conclude that $U_\alpha(\delta)$ is continuous, differentiable and concave for $\delta \in [\tau, 1-\tau]$. The update equation in line 5 of Algorithm 4 is a SA algorithm. Following the arguments given in Section 2.5, the equivalent ODE to that update equation is

$$\dot{\delta} = \frac{\partial U_\alpha(\delta)}{\partial \delta} \quad (\text{A.17})$$

$U_\alpha(\delta)$ is concave as shown earlier, so (A.17) converges towards the maximum of the α -fair utility functions.

Appendix B

Event-based Matlab simulator

We describe in this chapter the general architecture of the elastic traffic simulator used throughout this thesis and depicted in Figure B.1. It is an LTE event-based simulator coded in **Matlab** in which users arrive at random times in the network according to a temporal Poisson process with random locations, download a file of exponential random size and leave the network as soon as the download is complete.

As shown in Figure B.1, the first step of the simulator consists in setting the various simulation parameters. These may include but are not limited to:

Network parameters

Network map resolution, Inter-site distance, Frequency bandwidth, etc.

Channel characteristics

Pathloss model for each simulated BS type (macro, micro, etc.), Fading model and parameters, Shadowing standard deviation, Thermal noise, etc.

Traffic characteristics

Arrival rate, Traffic spatial distribution (notably the presence of hotspots and their size and location), Minimum distance between a user and any BS, Mean file size, etc.

Simulation parameters

Simulation duration, Admission thresholds, QoS thresholds (e.g. minimum SINR for coverage), self-optimizing algorithms parameters, etc.

In the second step, all the BSs' configurations are defined either by reading a text file or programmatically. The configurations include the BSs' locations in the network, their heights, their type (macro, micro, relay) which will determine which pathloss model to use, their transmit power, their antenna's type (omni, trisectorial or array), azimuth and elevation, and all other relevant parameters such as CIOs.

Using the parameters obtained from steps 1 and 2, an attenuation map is created in step 3 for the whole network. The attenuation combines all the elements impacting the signal from the BS to the user. The network map is represented by a two-dimensional matrix of pixels whose size

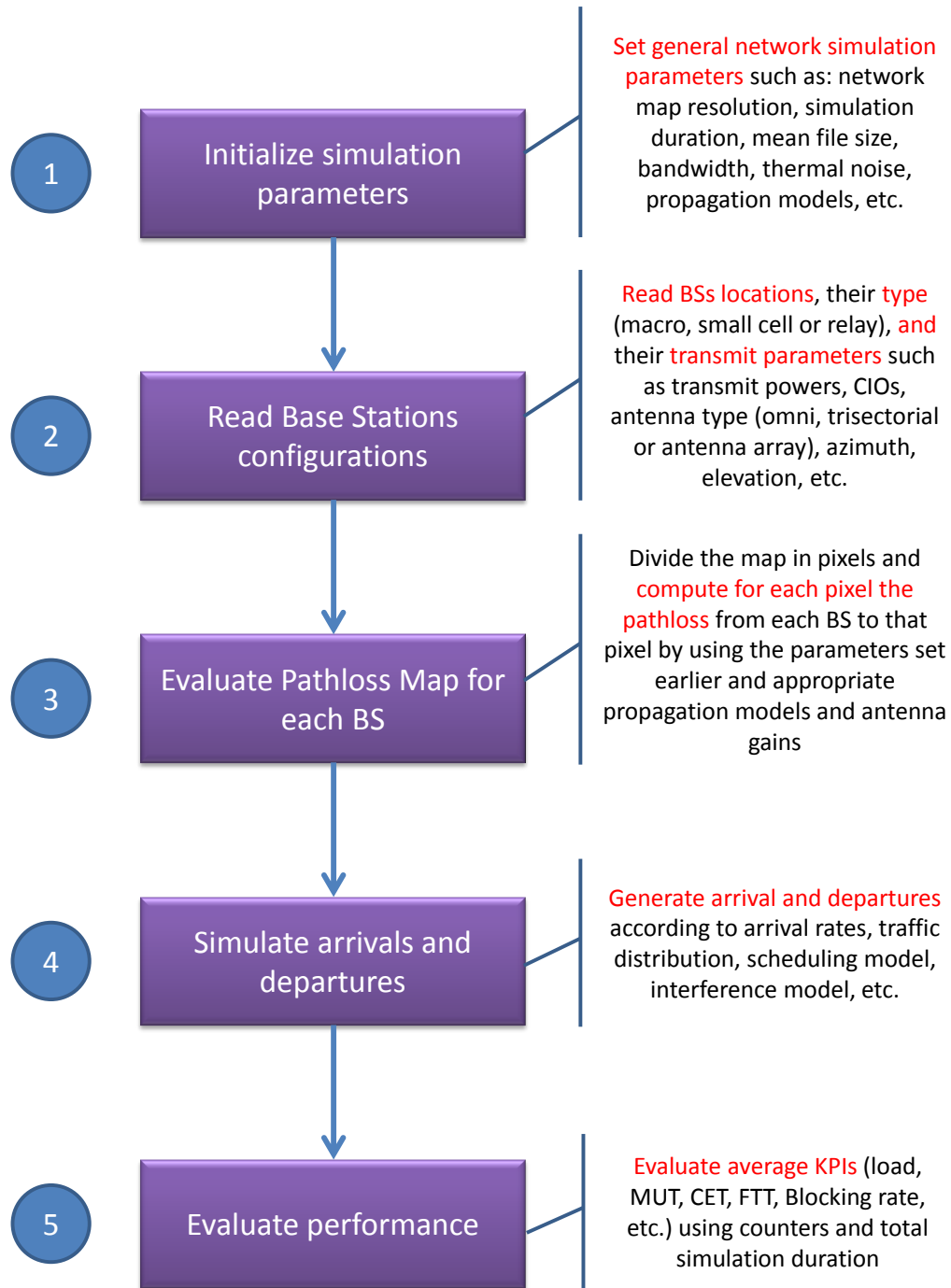


Figure B.1: General simulator architecture

depends on the resolution. Then for each BS and each pixel, knowing their position, the signal attenuation in decibels from the BS to the user can be evaluated as follows

$$attenuation = pathloss + shadowing + feederloss - antennagain \quad (B.1)$$

where *shadowing* is generated as a spatially correlated log-normal random variable, *feederloss* is the attenuation due to cable from the amplifier to the antenna, and *antennagain* is the gain of the BS's antenna in the specific direction of the pixel. *pathloss* follows the simple Hata model given by $A + B \log_{10}(r)$ where r is the distance in Km between the pixel and the BS, and A and B are parameters depending on the environment and the carrier frequency [1].

The antenna gain can be obtained directly from a real antenna diagram or using a 3GPP antenna gain model [1]. In some cases, we used a custom-made antenna model (courtesy of Dr. Zwi ALTMAN) for specific features such as vertical sectorization in order to get antenna diagrams corresponding to antenna arrays.

We evaluate the attenuation map at this point in order to alleviate calculations during the simulations. Once the attenuation map is obtained, the actual event-based simulation can start in step 4. The first user arrives in the network and its location is decided based on the traffic distribution. Its SINR is then calculated based on the network configuration and assumption on non-simulated interfering BSs. Using the SINR and a data rate model (e.g. [3]), the data rate of the user is derived since he is alone in the system. His data rate is then used together with the size of the file he is downloading to compute his departure time.

The next arrival time is also computed according to the arrival rate and we compare the departure time of the current user and the arrival time of the next user to determine which event will occur. If the arrival will occur first, the same operations as for the first user are performed. If the departure occurs first, the user is released and his mean throughput is computed and stored. The time counter is increased by the time spent till the current event which is the minimum between the arrival and departure times. Those operations are repeated until the simulation time attains the predefined simulation duration.

The SINRs are updated at each event because of presence or not of a user in one cell or another. Having the SINRs, the user average data rates between two events can be computed using a given scheduling model (round-robin, PF, etc.). Algorithm 7 summarizes the event-based simulation.

It is noted that all the parts of the simulator described in this appendix can be customized in order to implement specific features. For example, if an interference coordination mechanism is adopted, the SINR computation must be adapted accordingly. Likewise, if a SON function is implemented, some simulation parameters will be modified during the event-based simulation at specified time intervals.

Algorithm 7 Event-based simulation

```

1: Initialization using steps 1,2 and 3 in Figure B.1
2: Initialize simulation counters
3: loop:
4: while total_time < simulation_duration do
5:   Update KPI counters (e.g. load, total_time)
6:   Draw arrival_time  $\sim \exp(1/\text{arrival\_rate})$ 
7:   if at least one user is present then
8:     Update users sinrs
9:     Compute users' data_rates with scheduling
10:    departure_times = file_sizes/data_rates
11:    time_spent = min(arrival_time,departure_times)
12:    Update file_sizes by removing data_rates*time_spent
13:  else
14:    departure_times =  $+\infty$ 
15:  if arrival_time < min(departure_times) then
16:    Generate new user location
17:    Draw new user file_size  $\sim \exp(\text{mean\_file\_size})$ 
18:    Append file_size to file_sizes
19:    Update system state counters
20:  else
21:    departing_user = min_index(departure_times)
22:    Evaluate departing user mean throughput
23:    Update system state counters
24: Evaluate simulation KPIs

```

Bibliography

- [1] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects,” 3rd Generation Partnership Project (3GPP), TS 36.814 v9.0.0, Mar. 2010.
- [2] —, “Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions,” 3rd Generation Partnership Project (3GPP), TR 36.902, Mar. 2011.
- [3] —, “Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios,” 3rd Generation Partnership Project (3GPP), TR 36.942 v11.0.0, Sep. 2012.
- [4] —, “Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2,” 3rd Generation Partnership Project (3GPP), TS 36.300 v11.7.0, Sep. 2013.
- [5] —, “Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS),” 3rd Generation Partnership Project (3GPP), TS 28.628, Jun. 2013.
- [6] E. Altman, K. Avrachenkov, and U. Ayesta, “A Survey on Discriminatory Processor Sharing,” *Queueing Systems*, vol. 53, no. 1-2, pp. 53–63, Jun. 2006.
- [7] E. Altman, K. Avrachenkov, and A. Garnaev, “Generalized alpha-fair resource allocation in wireless networks,” in *Proc. of IEEE CDC*, 2008, pp. 2414–2419.
- [8] A. Bedekar and R. Agrawal, “Optimal muting and load balancing for eICIC,” in *Proc. of WiOpt*, 2013, pp. 280–287.
- [9] Y. Bejerano and S.-J. Han, “Cell breathing techniques for load balancing in wireless LANs,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 735–749, 2009.
- [10] Y. Bejerano, S.-J. Han, and L. E. Li, “Fairness and load balancing in wireless LANs using association control,” in *Proc. of ACM MobiCom*, 2004, pp. 315–329.
- [11] T. Bonald and M. Feuillet, *Performances des réseaux et des systèmes informatiques*, ser. Collection Télécom. Hermes Science Publications, 2011.

- [12] T. Bonald and A. Proutière, “Wireless Downlink Data Channels: User Performance and Cell Dimensioning,” in *Proc. of ACM Mobicom*, 2003.
- [13] T. Bonald and L. Massoulié, “Impact of fairness on Internet performance,” *SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 82–91, Jun. 2001.
- [14] T. Bonald, L. Massoulié, A. Proutière, and J. T. Virtamo, “A queueing analysis of max-min fairness, proportional fairness and balanced fairness,” *Queueing Systems*, vol. 53, no. 1-2, pp. 65–84, 2006.
- [15] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- [16] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. SIAM, Jun. 1994, vol. 15.
- [17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.
- [18] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient methods,” *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, 2003.
- [19] H. Braham, S. Ben Jemaa, B. Sayrac, G. Fort, and E. Moulines, “Coverage mapping using spatial interpolation with field measurements,” in *Proc. of IEEE PIMRC*, 2014, pp. 1743–1747.
- [20] O. Brickley, S. Rea, and D. Pesch, “Load balancing for qos enhancement in ieee802. 11e wlans using cell breathing techniques,” in *Proc. of IFIP MWCN*, 2005.
- [21] V. Cardellini, M. Colajanni, and S. Y. Philip, “Dynamic load balancing on web-server systems,” *IEEE Internet computing*, no. 3, pp. 28–39, 1999.
- [22] C. Chicone, *Ordinary differential equations with applications*. Springer, 2006, vol. 34.
- [23] R. Combes, Z. Altman, and E. Altman, “Self-organization in wireless networks: a flow-level perspective,” in *Proc. of IEEE INFOCOM*, 2012.
- [24] —, “Interference coordination in wireless networks: a flow-level perspective,” in *Proc. of IEEE INFOCOM*, 2013.
- [25] —, “Self-organizing relays: dimensioning, self-optimization, and learning,” *IEEE Transactions on Network and Service Management*, vol. 9, Dec. 2012.
- [26] —, “Coordination of autonomic functionalities in communications networks,” in *Proc. of WiOpt*, 2013.
- [27] G. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *Journal of parallel and distributed computing*, vol. 7, no. 2, pp. 279–301, 1989.

- [28] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for Enhanced Inter-Cell Interference Coordination (eICIC) in LTE HetNets," *IEEE/ACM Transactions on Networking*, 2013.
- [29] M. E. Fisher and A. T. Fuller, "On the stabilization of matrices and the convergence of linear iterative processes," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 04, pp. 417–425, 1958.
- [30] A. Galani, K. Tsagkaris, P. Demestichas, G. Nguengang, I. BenYahia, M. Stamatelatos, E. Kosmatos, A. Kaloxylos, and L. Ciavaglia, "Core functional and network empower mechanisms of an operator-driven, framework for unifying autonomic network and service management," in *Proc. of IEEE CAMAD*, 2012, pp. 191–195.
- [31] J. Ghimire and C. Rosenberg, "Resource Allocation, Transmission Coordination and User Association in Heterogeneous Networks: A Flow-Based Unified Approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1340–1351, Mar. 2013.
- [32] H. Gong and J. Kim, "Dynamic load balancing through association control of mobile users in wifi networks," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 342–348, 2008.
- [33] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.
- [34] T. Huckle and A. Kallischko, "Frobenius norm minimization and probing for preconditioning," *International Journal of Computer Mathematics*, vol. 84, no. 8, pp. 1225–1248, 2007.
- [35] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Multilevel millimeter wave beamforming for wireless backhaul," in *Proc. of IEEE GLOBECOM Workshops*, Dec. 2011, pp. 253–257.
- [36] O. C. Iacobaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "Son conflict resolution using reinforcement learning with state aggregation," in *Proc. of ACM SIGCOMM Workshops*, 2014, pp. 15–20.
- [37] M. Imran, E. Katranaras, G. Auer, O. Blume, V. Giannini, I. Godor, Y. Jading, M. Olsson, D. Sabella, P. Skillermarck, and others, "Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," Tech. Rep. ICT-EARTH deliverable, Tech. Rep., 2011.
- [38] T. Jansen, M. Amirijoo, U. Turke, L. Jorguseski, K. Zetterberg, R. Nascimento, L. Schmelz, J. Turk, and I. Balan, "Embedding Multiple Self-Organisation Functionalities in Future Radio Access Networks," in *Proc. of IEEE VTC Spring*, 2009, pp. 1–5.
- [39] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta Mathematica*, vol. 30, no. 1, pp. 175–193, 1906.

- [40] Y. Khan, B. Sayrac, and E. Moulines, "Centralized self-optimization of pilot powers for load balancing in lte," in *Proc. of IEEE PIMRC*, 2013, pp. 3039–3043.
- [41] —, "Surrogate Based Centralized SON: Application to Interference Mitigation in LTE-A HetNets," in *Proc. of IEEE VTC Spring*, 2013, pp. 1–5.
- [42] J. Kiefer, J. Wolfowitz *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.
- [43] H. Kim, G. de Veciana, X. Yang, and M. Venkatachalam, "alpha-Optimal User Association and Cell Load Balancing in Wireless Networks," in *Proc. of IEEE INFOCOM*, Mar. 2010, pp. 1–5.
- [44] L. Kleinrock, *Queuing systems*. Wiley, 1975.
- [45] H. Klessig, A. Fehske, G. Fettweis, and J. Voigt, "Improving coverage and load conditions through joint adaptation of antenna tilts and cell selection rules in mobile networks," in *Proc. of IEEE ISWCS*, 2012, pp. 21–25.
- [46] Z. Kostic, K. K. Leung, and H. Yin, "WLAN having load balancing based on access point loading," Jul. 2008, US Patent 7,400,901.
- [47] H. J. Kushner and P. A. Whiting, "Convergence of Proportional-Fair Sharing Algorithms Under General Conditions," *IEEE transactions on wireless communications*, vol. 3, pp. 1250–1259, Jul. 2004.
- [48] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications 2nd edition*. Springer Stochastic Modeling and Applied Probability, 2003.
- [49] K. Linehan and R. Chandrasekaran, "Active antennas: The next step in radio and antenna evolution," *Commonscope White Paper*, vol. 10, 2011.
- [50] Z. Liu, P. Hong, K. Xue, and M. Peng, "Conflict Avoidance between Mobility Robustness Optimization and Mobility Load Balancing," in *Proc. of IEEE GLOBECOM*, 2010, pp. 1–5.
- [51] D. Lopez-Perez, I. Guvenc, G. de la Roche, M. Kountouris, T. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 22–30, Jun. 2011.
- [52] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 2001.
- [53] R. R. Müller, M. A. Sedaghat, and G. Fischer, "Load modulated massive MIMO," in *Proc. of IEEE GlobalSIP*, Dec. 2015.
- [54] R. Nasri and Z. Altman, "Handover adaptation for dynamic load balancing in 3GPP Long Term Evolution systems," in *Proc. of MoMM*, Dec. 2007.

- [55] Y. Nesterov, A. Nemirovskii, and Y. Ye, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994, vol. 13.
- [56] A. Osseiran *et al.*, “Scenarios for 5G mobile and wireless communications: the vision of the METIS project,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.
- [57] J. Pang, J. Wang, D. Wang, G. Shen, Q. Jiang, and J. Liu, “Optimized time-domain resource partitioning for enhanced inter-cell interference coordination in heterogeneous networks,” in *Proc. of IEEE WCNC*, 2012, pp. 1613–1617.
- [58] K. I. Pedersen, Y. Wang, B. Soret, and F. Frederiksen, “eICIC Functionality and Performance for LTE HetNet Co-Channel Deployments,” in *Proc. of IEEE VTC Fall*, 2012, pp. 1–5.
- [59] G. Poullos, K. Tsagkaris, P. Demestichas, A. Tall, Z. Altman, and C. Destré, “Autonomics and SDN for Self-Organizing Networks,” in *Proc. of IWSON*, Aug. 2014.
- [60] A. Pyzara, B. Bylina, and J. Bylina, “The influence of a matrix condition number on iterative methods’ convergence,” in *Proc. of IEEE FedCSIS*, 2011, pp. 459–464.
- [61] H. Robbins and S. Monro, “A Stochastic Approximation Method,,” *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.
- [62] J. Rodriguez, I. de la Bandera, P. Munoz, and R. Barco, “Load balancing in a realistic urban scenario for LTE networks,” in *Proc. of IEEE VTC Spring*, 2011, pp. 1–5.
- [63] J. B. Rosen, “Existence and Uniqueness of Equilibrium Points for Concave N-Person Games,” *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.
- [64] S. Roy, J. Minter, and A. Saberi, “Some new results on stabilization by scaling,” in *Proc. of ACC*, 2006.
- [65] L.-C. Schmelz, M. Amirijoo, A. Eisenblaetter, R. Litjens, M. Neuland, and J. Turk, “A coordination framework for self-organisation in LTE networks,” in *Proc. of IFIP/IEEE IM*, 2011, pp. 193–200.
- [66] M. Shirakabe, A. Morimoto, and N. Miki, “Performance evaluation of inter-cell interference coordination and cell range expansion in heterogeneous networks for LTE-Advanced downlink,” in *Proc. of IEEE ISWCS*, 2011, pp. 844–848.
- [67] C. P. Simon and L. Blume, *Mathematics for economists*. Norton New York, 1994, vol. 7.
- [68] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.
- [69] Small Cell Forum 5.1, “Backhaul technologies for small cells,” *White Paper*, Feb. 2014.

- [70] J. Suga, Y. Kojima, and M. Okuda, "Centralized mobility load balancing scheme in LTE systems," in *Proc. of ISWCS*, 2011, pp. 306–310.
- [71] A. Tall, Z. Altman, and E. Altman, "Self organizing strategies for enhanced ICIC (eICIC)," in *Proc. of WiOpt*, May 2014, pp. 318–325.
- [72] —, "Self-optimizing Strategies for Dynamic Vertical Sectorization in LTE Networks," in *Proc. of IEEE WCNC*, 2015.
- [73] —, "Virtual sectorization: design and self-optimization," in *Proc. of IWSON*, May 2015.
- [74] A. Tall, R. Combes, Z. Altman, and E. Altman, "Distributed coordination of self-organizing mechanisms in communication networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 328–337, 2014.
- [75] K. Trichias, R. Litjens, A. Tall, Z. Altman, and P. Ramachandra, "Performance Evaluation & SON Aspects of Vertical Sectorisation in a Realistic LTE Network Environment," in *Proc. of IWSON*, Aug. 2014.
- [76] —, "Self-optimisation of vertical sectorisation in a realistic lte network," in *Proc. of EuCNC*, 2015, pp. 149–153.
- [77] K. Tsagkaris, G. Poullos, P. Demestichas, A. Tall, Z. Altman *et al.*, "An open framework for programmable, self-managed radio access networks," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 154–161, 2015.
- [78] Unverself, "Unified Management Framework (UMF) Specifications Release 3," 2013.
- [79] S. Vasudevan, R. Pupala, and K. Sivanesan, "Dynamic eICIC—A Proactive Strategy for Improving Spectral Efficiencies of Heterogeneous LTE Cellular Networks by Leveraging User Mobility and Traffic Dynamics," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 4956–4969, 2013.
- [80] I. Viering, M. Dottling, and A. Lobinger, "A mathematical perspective of self-optimizing wireless networks," in *Proc. of IEEE ICC*, 2009, pp. 1–6.
- [81] P. Vlacheas, E. Thomatos, K. Tsagkaris, and P. Demestichas, "Operator-governed SON coordination in downlink LTE networks," in *Proc. of FutureNetw*, 2012, pp. 1–9.
- [82] H. Wang, L. Ding, P. Wu, Z. Pan, N. Liu, and X. You, "Dynamic load balancing and throughput optimization in 3GPP LTE networks," in *Proc. of IWCMC*, 2010, pp. 939–943.
- [83] Y. Wang and K. Pedersen, "Performance Analysis of Enhanced Inter-Cell Interference Coordination in LTE-Advanced Heterogeneous Networks," in *Proc. of IEEE VTC Spring*, May 2012, pp. 1–5.

-
- [84] M. H. Willebeek-LeMair and A. P. Reeves, "Strategies for dynamic load balancing on highly parallel computers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 9, pp. 979–993, 1993.
 - [85] J.-H. Yun and K. G. Shin, "CTRL: a self-organizing femtocell management architecture for co-channel deployment," in *Proc. of ACM MobiCom*, 2010, pp. 61–72.

Summary

The mobile network of Orange in France comprises more than 100 000 2G, 3G and 4G antennas with several frequency bands, not to mention many femto-cells for deep-indoor coverage. These numbers will continue to increase in order to address the customers' exponentially increasing need for mobile data. This is an illustration of the challenge faced by the mobile operators for operating such a complex network with low Operational Expenditures (OPEX) in order to stay competitive. This thesis is about leveraging the Self-Organizing Network (SON) concept to reduce this complexity by automating repetitive or complex tasks. We specifically propose automatic optimization algorithms for scenarios related to network densification using either small cells or Active Antenna Systems (AASs) used for Vertical Sectorization (VeSn), Virtual Sectorization (ViSn) and multilevel beamforming. Problems such as load balancing with limited-capacity backhaul and interference coordination either in time-domain (eICIC) or in frequency-domain are tackled. We also propose optimal activation algorithms for VeSn and ViSn when their activation is not always beneficial. We make use of results from stochastic approximation and convex optimization for the mathematical formulation of the problems and their solutions. We also propose a generic methodology for the coordination of multiple SON algorithms running in parallel using results from concave game theory and Linear Matrix Inequality (LMI)-constrained optimization.

Résumé

Le réseau mobile d'Orange France comprend plus de 100 000 antennes 2G, 3G et 4G sur plusieurs bandes de fréquences sans compter les nombreuses femto-cells fournies aux clients pour résoudre les problèmes de couverture. Ces chiffres ne feront que s'accroître pour répondre à la demande sans cesse croissante des clients pour les données mobiles. Cela illustre le défi énorme que rencontrent les opérateurs de téléphonie mobile en général à savoir gérer un réseau aussi complexe tout en limitant les coûts d'opération pour rester compétitifs. Cette thèse s'attache à utiliser le concept SON (réseaux auto-organisés) pour réduire cette complexité en automatisant les tâches répétitives ou complexes. Plus spécifiquement, nous proposons des algorithmes d'optimisation automatique pour des scénarios liés à la densification par les small cells ou les antennes actives. Nous abordons les problèmes classiques d'équilibrage de charge mais avec un lien backhaul à capacité limitée et de coordination d'interférence que ce soit dans le domaine temporel (notamment avec le eICIC) ou le domaine fréquentiel. Nous proposons aussi des algorithmes d'activation optimale de certaines fonctionnalités lorsque cette activation n'est pas toujours bénéfique. Pour la formulation mathématique et la résolution de tous ces algorithmes, nous nous appuyons sur les résultats de l'approximation stochastique et de l'optimisation convexe. Nous proposons aussi une méthodologie systématique pour la coordination de multiples fonctionnalités SON qui seraient exécutées en parallèle. Cette méthodologie est basée sur les jeux concaves et l'optimisation convexe avec comme contraintes des inégalités matricielles linéaires.

Abdoulaye TALL was born in 1988 in Ivory Coast. He received the Engineering degree from TPS (Tunisia Polytechnic School, La Marsa, Tunisia) in 2012. He is currently pursuing a Ph.D. degree with Orange Labs under the direction of Zwi Altman (Orange Labs), Eitan Altman (INRIA). His current research interests include self-organizing mobile networks, queuing theory, control theory and stochastic approximation.

