

### THESE DE DOCTORAT DE L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité : Mathématiques

## École Doctorale Paris Centre

présentée par

## **Richard COMBES**

Pour obtenir le grade de

## DOCTEUR DE L'UNIVERSITE PIERRE ET MARIE CURIE

Sujet de la thèse:

# Mécanismes auto-organisants dans les réseaux sans fil

dirigée par Zwi Altman / Eitan Altman / Sylvain Sorin

Soutenue le 15/02/2013 devant le jury composé de :

Prof. Serge Fdida
Prof. Sylvain Sorin
Prof. Eitan Altman
Dr. Zwi Altman
Prof. Stéphane Gaubert
Prof. Thomas Bonald
Prof. Mérouane Debbah
Dr. Adam Ouorou

UPMC UPMC INRIA Orange Labs INRIA Télécom Paristech Supélec Orange Labs

président directeur directeur rapporteur examinateur examinateur examinateur

Laboratoire Combinatoire et Optimisation Institut de Mathématiques de Jussieu, CNRS, UMR 7586 Université Pierre et Marie Curie (Paris 6), Case 247 4 place Jussieu, 75252 Paris Cedex 05

> École doctorale Paris centre Case 188 4 place Jussieu 75 252 Paris cedex 05

## Acknowledgments

I would like to thank my wife, family and friends who supported and encouraged me throughout the completion of this thesis.

I am honored to have worked with my three advisers Dr. Zwi Altman, Prof. Eitan Altman and Prof. Sylvain Sorin, whose advice and guidance have enabled my to complete this thesis.

I would like to thank the jury members, and I am grateful to Prof. Vivek Borkar and Prof. Stéphane Gaubert who have spent their time and effort reviewing this manuscript. Their remarks have significantly improved the quality of this manuscript.

Finally I would like to thank my colleagues at Orange Labs, Paris VI and INRIA for helpful discussions and camaraderie throughout this thesis. Particular thanks to my coauthors: Dr. Salah Eddine Elayoubi, Dr. Louai Saker, Prof. Tijani Chahed, Dr. Stéphane Sénecal, Dr. Daniel Mustaki, Dr. Majed Haddad, Dr. Sara Akbarzadeh, Dr. Veeraruna Kavitha, Ilham El Bouloumi and Liu Sulan.

## Abstract

## Mécanismes auto-organisants dans les réseaux sans fil

## Résumé

Dans cette thèse on étudie la mise au point, la modélisation et la performance de mécanismes (dits auto-organisants) pour gérer les réseaux sans fils de façon autonome. Le contexte technologique est rappelé, et les outils mathématiques nécessaires sont introduits succinctement: théorie des files d'attente, processus ponctuels, théorie de l'information, approximation stochastique, processus de décision markoviens et apprentissage par renforcement.

Dans une première partie, on s'intéresse à l'évaluation de performance des ordonnanceurs opportunistes, et à leur utilisation pour l'optimisation capacité/couverture. Les phénomènes de la couche physique tels que l'évanouissement rapide du canal, les interférences, la structure du récepteur et les schémas de modulation et codage pratiques sont pris en compte.

Dans la deuxième partie, un mécanisme d'équilibrage de charge automatique prenant en compte les arrivées et départs des utilisateurs est présenté. Pour un trafic stationnaire, sa convergence vers l'optimum est prouvée par une technique d'approximation stochastique. Pour un trafic non stationnaire, des expériences numériques suggèrent que la méthode est capable de s'adapter aux variations de trafic journalières.

Dans une troisième partie, on s'intéresse aux réseaux avec relais. Une formule analytique simple basée sur la théorie des files d'attentes est proposée pour leur dimensionnement. La formule est valable pour le modèle de trafic le plus général (stationnaire ergodique). Le mécanisme d'équilibrage de charge est étendu pour prendre en compte les relais. Une méthode d'équilibrage de charge dynamique utilisant l'apprentissage par renforcement est étudiée.

#### Mots-clefs

réseaux sans fils, réseaux cellulaires, auto-organisation, auto-configuration, auto-optimisation, auto-réparation, théorie des files d'attente, processus ponctuels, théorie de l'information, approximation stochastique, processus de décision markoviens, apprentissage par renforcement

# Self-organizing mechanisms in wireless networks

### Abstract

In this thesis we study the design, modeling and performance evaluation of mechanisms which can manage wireless networks autonomously (self-organizing mechanisms). We recall the technological context, and the required mathematical tools are introduced concisely: queuing theory, point processes, information theory, stochastic approximation, Markov decisions processes and reinforcement learning.

In the first part, we study opportunistic scheduling. We are interested in their performance evaluation and their use to perform coverage-capacity optimization. Physical layer phenomena such as channel fading, interference, receiver structure and practical modulation and coding schemes are taken into account.

In the second part, an algorithm for automatic load balancing is presented. The dynamical arrivals and departures of users are taken into account. For stationary traffic, the convergence of the mechanism to the optimal configuration is shown using stochastic approximation theorems. For non-stationary traffic, numerical experiments suggest that the mechanism is able to adapt itself to daily traffic patterns.

In the third part, we study relay-enhanced networks. Based on a queuing analysis, a simple formula for network dimensioning is given. It is valid for the most general traffic model (stationary ergodic input). The load balancing mechanism is extended to relay-enhanced networks. A dynamical load balancing algorithm based on reinforcement is studied.

## Keywords

wireless networks, cellular networks, self-organization, self-configuration, self-optimization, self-healing, queuing theory, point processes, information theory, stochastic approximation, Markov decisions processes and reinforcement learning

# Résumé en francais

Dans cette thèse on étudie la mise au point, la modélisation et la performance de mécanismes (dits auto-organisants) pour gérer les réseaux sans fils de façon autonome.

Un réseau auto-organisant ("Self-Organizing Network" ou SON, en anglais) est un réseau capable de se gérer lui-même de façon autonome sans l'aide d'un opérateur humain. La gestion du réseau comporte toutes les tâches effectuées à l'heure actuelle par des ingénieurs réseaux. Ces tâches sont: le déploiement et la configuration de nouveaux noeuds du réseau, l'optimisation des paramètres, la maintenance et la réparation des pannes. Les ingénieurs utilisent des mesures rapportées par les stations de base et des sondes placées sur certaines interfaces du réseau. Des indicateurs de performance et des alarmes sont générés à partir de ces mesures. L'optimisation est généralement assistée par des outils d'aide à la décision. Un réseau SON doit être capable de s'auto-configurer, s'auto-optimiser et de s'auto-réparer. Les organismes de standardisation tels que 3GPP (3rd Generation Partnership Project) et de pré-standardisation tels que NGMN (Next Generation Mobile Network) ont identifié le SON comme un élément clé des réseaux futurs. Dans le cadre des réseaux mobiles, les cas d'étude les plus importants pour le SON ont été identifiés: la gestion des interférences inter-cellules, l'équilibrage de charge, la mobilité robuste, et la réduction de la consommation d'énergie par les équipements. Les algorithmes SON seront implémentés dans les équipements du réseau, ce qui impose plusieurs limitations. Les algorithmes doivent être distribués, nécessiter une faible puissance de calcul, être tolérants aux délais, nécessiter peu d'échanges d'information de signalisation, et être robustes au bruit.

Dans un chapitre préliminaire, les outils mathématiques nécessaires pour la mise au point, la modélisation, l'analyse et l'évaluation de performance des mécanismes auto-organisants dans les réseaux sont traités. Les outils utilisés sont: la théorie des files d'attente et des processus ponctuels, la théorie de l'information, l'approximation stochastique, les processus de décision markoviens et l'apprentissage par renforcement. La théorie des files d'attente nous permet de modéliser le comportement du réseau en prenant en compte les arrivées et départs dynamiques des utilisateurs. Prendre en compte le comportement dynamique du réseau est fondamental pour l'étude des mécanismes SON. En y ajoutant la théorie des processus ponctuels, nous pouvons prendre en compte des processus d'arrivées aussi généraux que possible grâce au théorème de Loynes. Nous pouvons ainsi développer des algorithmes SON valables avec peu d'hypothèses sur le modèle de trafic sousjacent. La théorie de l'information est introduite afin d'évaluer la performance du réseau au niveau lien, c'est à dire quand une seule station de base est présente et que les utilisateurs sont immobiles. Les phénomènes de la couche physique tels que l'évanouissement rapide du canal, les interférences, la structure du récepteur et les schémas de modulation et codage pratiques sont pris en compte. Cette base théorique nous servira pour la modélisation et l'évaluation de performance des ordonnanceurs opportunistes. On introduit ensuite la théorie de l'approximation stochastique, qui établit un lien entre certains algorithmes itératifs en présence de bruit, et une équation différentielle ordinaire. Un lien précis existe entre le comportement de l'algorithme et certains ensembles limites de l'équation différentielle. L'intérêt de cette approche est que l'équation différentielle est purement déterministe et plus simple à analyser que l'algorithme itératif qui est un processus stochastique. L'approximation stochastique est essentielle pour l'analyse de la convergence des algorithmes SON qui sont des algorithmes itératifs en présence de bruit. Ce bruit est lié au fait que les algorithmes SON se basent sur des mesures. On peut alors démontrer de façon élégante que ces algorithmes convergent, même en présence de bruit. Finalement, les processus de décision markoviens sont traités. La technique d'uniformisation qui permet de réduire un processus de décision semi-markovien (en temps continu) à un processus de décision markovien en temps discret est décrite. L'apprentissage par renforcement, qui consiste à trouver le contrôle optimal d'un processus de décision markoviens sans connaitre les probabilités de transition est introduit. On s'intéresse en particulier aux techniques d'approximation de politiques, qui permettent d'utiliser l'apprentissage par renforcement pour des problèmes de grande dimension. L'apprentissage par renforcement est appliqué pour allouer dynamiquement les ressources dans un réseau avec relais.

Dans une première partie, on s'intéresse à l'évaluation de performance des ordonnanceurs opportunistes, et à leur utilisation pour l'optimisation capacité/couverture. On étudie le modèle classique dans lequel un nombre fixe d'utilisateurs est servi par une station de base, et possède un nombre infini de paquets à transmettre. Ce modèle est dit "full-buffer". La station connait l'état du canal de chacun des utilisateurs et décide dynamiquement quel utilisateur peut transmettre. On considère une famille d'ordonnanceurs dite  $\alpha$ -équitable. Cette famille comprend en particulier l'équité proportionnelle ( $\alpha = 1$ ), l'équité min-max ( $\alpha \to +\infty$ ) et la maximisation de la somme des débits ( $\alpha = 0$ ). La convergence de l'algorithme est étudiée en utilisant l'approximation stochastique. La preuve est une généralisation de la preuve de convergence dans le cas particulier de l'équité proportionnelle. On calcule les débits moyens alloués par l'ordonnanceur aux utilisateurs à l'aide de formules analytiques. Les cas étudiés sont  $\alpha \in \{0, 1, +\infty\}$ . Pour la plupart des modèles de canal rencontrés dans les réseaux opérationnels, une formule analytique est donnée. Par exemple: réseaux OFDMA (par exemple LTE), réseaux CDMA (par exemple 3G et HSPA), réseaux MIMO-OFMDA (par exemple LTE-Advanced). Finalement, on étudie un mécanisme SON permettant de changer la stratégie d'ordonnancement dynamiquement (la valeur de  $\alpha$ ) pour optimiser la qualité de service d'un réseau. On considère un modèle dans lequel les utilisateurs arrivent et repartent dynamiquement. Durant les périodes de congestion, la valeur de  $\alpha$  augmente pour s'assurer que tous les utilisateurs atteignent un débit cible. Les utilisateurs qui n'atteignent pas ce débit cible sont coupés. Une étude numérique à l'aide d'un simulateur de réseau dynamique montre que le mécanisme proposé permet de réduire la probabilité de coupure au prix d'une faible perte de débit global du réseau.

Dans la deuxième partie, un mécanisme d'équilibrage de charge automatique prenant en compte les arrivées et départs des utilisateurs est présenté. On considère un trafic élastique pour le lien descendant, dans lequel les utilisateurs téléchargent une quantité de données finie et aléatoire et quittent le réseau ensuite. Ce modèle de trafic décrit les applications de données telles que le trafic web (HTTP) et transfert de fichiers (FTP). Les utilisateurs s'attachent à la station dont le signal pilote reçu est le plus fort. Les résultats de la théorie des files d'attente permettant le calcul de la performance stationnaire du système sont rappelés. Les stations de base n'ont pas de connaissances sur le modèle de trafic et la géométrie du réseau et utilisent des mesures remontées par les utilisateurs arrivant dans le réseau pour estimer leurs charges. Les propriétés statistiques des estimateurs de charge sont étudiées. Basé sur ces charges estimées, un mécanisme d'équilibrage de charge est proposé. Les stations les plus chargées réduisent leur puissance pilote transmise, ce qui réduit la zone qu'elles servent, diminue leur charge et permet à leurs voisines de les décharger. Une station, par exemple la plus chargée, est définie comme station de référence, et diffuse sa charge estimée aux autres stations. Les stations calculent la différence entre leur charge et la charge de la station de référence et augmentent leur puissance transmise de facon proportionnelle à cette différence. La convergence du mécanisme d'équilibrage est étudiée grâce à l'approximation stochastique. On montre que l'équation différentielle associée converge vers un état dans lequel toutes

les stations ont la même charge, et que cet état est stable au sens de Lyapunov. Une extension possible au trafic à débit constant (services voix et streaming) est décrite. Des expériences numériques suggèrent que le mécanisme d'équilibrage converge suffisamment rapidement pour s'adapter aux variations journalières du trafic. Les aspects pratiques du mécanisme sont également abordés: échelles de temps, fréquence de mise à jour, charge de signalisation, tolérance aux délais.

Dans une troisième partie, on s'intéresse aux réseaux avec relais. Comme dans la partie précédente on considère un réseau servant du trafic élastique, pour le lien descendant. Les relais sont des noeuds du réseau qui n'ont pas de lien filaire avec le coeur de réseau, et sont raccordés à une station de base par un lien sans fil. Quand un utilisateur est servi par un relai, les données qu'il reçoit passent par un lien station-relai, puis par le lien relai-utilisateur. Les ressources radio sont partagées entre les liens directs (stations-utilisateurs et relais-utilisateurs) et les liens dits "backhaul" (liens stations-relais). Une formule analytique simple basée sur la théorie des files d'attentes est proposée pour leur dimensionnement. La formule est valable pour le modèle de trafic le plus général (stationnaire ergodique). L'influence du placement des relais sur le gain de capacité est étudiée. Un résultat important est qu'un déploiement anarchique des relais peut diminuer de façon notable la capacité du réseau. Le mécanisme d'équilibrage de charge étudié dans la seconde partie est généralisé pour prendre en compte les relais. Deux paramètres sont ajustés simultanément: la puissance pilote transmise par les relais qui contrôle la zone servie par ceux-ci et la quantité de ressources allouées aux liens backhaul. La convergence est prouvée en utilisant une approche d'approximation stochastique avec échelles de temps multiples. Finalement, le problème d'allocation de ressources est modélisé comme un processus de décision Markovien. Cela permet d'adapter le réseau à la position instantanée des utilisateurs actifs plutôt qu'à la charge qui représente la performance moyenne du système. Une famille de politiques dont la performance est proche de l'optimum est introduite. Un mécanisme d'apprentissage par renforcement est étudié, pour trouver la meilleure politique. L'intérêt de l'apprentissage par renforcement est que il n'est pas nécessaire de connaitre la dynamique du système c'est à dire les probabilités de transition, qui dépendent de l'intensité du trafic et de sa répartition spatiale. De plus, le mécanisme d'apprentissage fonctionne pour une large classe de processus d'arrivée et n'est pas limité au cas où le trafic suit un processus de Poisson.

En conclusion, les perspectives futures ouvertes par ces travaux sont abordées, notamment le problème de la coordination de mécanismes SON multiples fonctionnant en parallèle.

# Contents

1	Intr	oducti	ion	19
	1.1	The S	ON concept	19
		1.1.1	Autonomic network management	19
		1.1.2	SON in standards	21
	1.2	Challe	enges for SON in wireless networks	23
		1.2.1	The key use cases	23
		1.2.2	Requirements for SON solutions	25
	1.3	Our C	Contribution	28
		1.3.1	Content and methodology	28
		1.3.2	Organization	28
		1.3.3	Publications	28
<b>2</b>	The	oretica	al foundations	<b>31</b>
	2.1	Queui	ng theory	31
		2.1.1	Point processes	31
		2.1.2	Classical queues	35
	2.2	Inform	nation theory	37
		2.2.1	Source coding theorem	38
		2.2.2	Noisy channel coding theorem	42
		2.2.3	Continuous channels	45
		2.2.4	Channel models	45
	2.3	Stocha	astic approximation	52
		2.3.1	Definitions	53
		2.3.2	The ODE approach	53
		2.3.3	Martingale difference noise: decreasing step sizes	56
		2.3.4	Martingale difference noise: constant step sizes	56
		2.3.5	Correlated noise: decreasing step sizes	57
		2.3.6	Correlated noise: constant step sizes	59
	2.4	Reinfo	preement learning	59
		2.4.1	Markov decision processes	60
		2.4.2	Q-learning	63

		049	Delien anna d
		2.4.5	Policy search approach
		2.4.4	Continuous time models
		2.4.5	Semi-Markov decision processes
3	Pac	ket sch	neduling 73
	3.1	Chanr	nel-aware scheduling
		3.1.1	The model $\ldots$ $74$
		3.1.2	$\alpha$ -fair scheduling
	3.2	Conve	rgence of $\alpha$ -fair schedulers
		3.2.1	The mean ODE
		3.2.2	Convergence to a unique limit
	3.3	Calcul	lation of scheduling gain
		3.3.1	Rayleigh-fading AWGN
		3.3.2	Multi-tap Rayleigh-fading AWGN
		3.3.3	MIMO Rayleigh-fading AWGN
	3.4	Nume	rical experiments
		3.4.1	Rayleigh-fading AWGN
		3.4.2	Multi-tap Rayleigh-fading AWGN
		3.4.3	MIMO Rayleigh-fading AWGN
	3.5	Covera	age-capacity optimization
		3.5.1	Algorithm
		3.5.2	Admission Control
		3.5.3	Simulation
		3.5.4	Simulation Results
4	Таа	d hala	noing 101
4	1 1	u bala	Inclug 101
	4.1	Г IOW-I	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
		4.1.1	
	4.0	4.1.2	Load estimation
	4.2	Load	Dalancing mechanism
		4.2.1	The mass ODE
		4.2.2	Comment ODE
		4.2.3	Convergence of the load balancing mechanism 100
	4.9	4.2.4	Extension to constant data rate trame
	4.3	Nume	rical experiments
	4.4	Proofs	$\mathbf{S} \dots \dots$
		4.4.1	Proof of Theorem 4.2
		4.4.2	Proof of Theorem 4.3
		4.4.3	Lemma 4.1
		4.4.4	Proof of Theorem $4.4$
		4.4.5	Proof of theorem $4.5$

#### CONTENTS

<b>5</b>	5 Relay networks			117
	5.1 Dimensioning $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$			
		5.1.1	System model	. 118
		5.1.2	System capacity	. 120
		5.1.3	Relay gain	. 122
		5.1.4	Numerical experiments	. 122
5.2 Self-Optimization				. 124
		5.2.1	Traffic estimation	. 125
		5.2.2	Traffic balancing for the backhaul	. 127
		5.2.3	Coordination between backhaul and cell sizes	. 128
		5.2.4	Numerical experiments	. 129
	5.3	Dynan	nic resource allocation	. 130
		5.3.1	Infinite buffer case: stabilizing policy	. 130
		5.3.2	Finite buffer case: MDP formulation	. 132
	5.4	Learni	ng	. 137
		5.4.1	Policy gradient approach	. 138
		5.4.2	Convergence to a local optimum	. 138
		5.4.3	Implementation issues: traffic and scalability	. 140
		5.4.4	Numerical experiments	. 140
	5.5	Proofs		. 141
		5.5.1	Proof of theorem $5.1$	. 141
		5.5.2	Traffic estimation	. 143
		5.5.3	Proof of theorem 5.2 $\ldots$ $\ldots$ $\ldots$ $\ldots$	. 145
		5.5.4	Proof of theorem 5.3 $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	. 147
6	3 Conclusion and future work 1			151
7 Appendices				153
	7.1	Simula	tion methodology	. 153
	7.2	Acrony	yms	. 155
In	$\mathbf{dex}$			159
Bi	bliog	graphy		163

CONTENTS

# Notations

The following notations are	used in this thesis:
$\mathbb{N}$	set of positive integers
$\mathbb{Z}$	set of integers
$\mathbb{R}$	set of real numbers
$\mathbf{P}\left[. ight]$	probability
$E\left[. ight]$	expectation
$\mathbf{var}(.)$	variance
	independence of random variables
	equality of random variables in distribution
$\xrightarrow{a.s}$	almost sure convergence
$\xrightarrow{\mathbf{P}}$	convergence in probability
$\stackrel{d}{\rightarrow}$	convergence in distribution
$\xrightarrow{\mathcal{L}^2}$	convergence in mean square
.	Euclidean norm
$\langle . , . \rangle$	(Euclidean) scalar product
H	conjugate transpose
$\ f\ _{\mathcal{L}^{2}} = \sqrt{\sum_{t \in \mathbb{N}}  f[t] ^{2}}$	$\mathcal{L}^2$ norm
$\langle f, g \rangle_{\mathcal{L}^2} = \sum_{t \in \mathbb{N}} f[t]^H g[t]$	$\mathcal{L}^2$ inner product
$ A  = \int_A dr$	Lebesgue measure (with $A$ Borel set)
$\mathcal{N}(\mu,\sigma^2)$	Gaussian distribution with mean $\mu$
	and variance $\sigma^2$

CONTENTS

# Chapter 1

# Introduction

In this chapter we define the Self-organizing networks (SON) concept and why it currently represents a key problem for both industrials and researchers in the field of networking. We highlight the key use cases as far as wireless networks are concerned and the requirements for the adoption at a large scale of the SON technology by industry and network operators.

## 1.1 The SON concept

#### 1.1.1 Autonomic network management

The concept of SON comes from network management. *Network management* is composed of all the tasks necessary to the deployment of networks, their daily optimization, and the detection and correction of faults. Network management is performed daily by network engineers, and is highly complex and costly.

- Node deployment and configuration tasks generally includes hardware installation, setup of transport interfaces between the new node and the existing network, setup of a secure tunnel to gateways and the Operation and Maintenance Center (OMC), download of software needed for the node operation, download of configuration files, authentication of the new node and tests before the node can be fully operational.
- Optimization is done when poor performance is observed, e.g coverage holes, congestion, dropped calls, and concerns parameters which have a strong impact on the network performance. Optimization is done by network engineers, based on measurements. Measurements are performed by network users, and are sent to the OMC by either the Base

Stations (BSs) or probes installed on specific network interfaces. Engineers typically rely on optimization software. In the case of wireless networks, the optimized parameters include antenna tilts, frequency planning, transmitted powers, handover parameters and resources allocated to random access channels.

Maintenance and troubleshooting are done similarly to optimization, based on node measurements, customer complaints and field measurements e.g drive tests. Troubleshooting consists of fault detection, fault diagnosis and fault repair and is done on a daily basis. It includes alarm processing, update of obsolete software and hardware at the nodes. A major task is to detect, compensate and/or mitigate the effect of outages at the nodes.

The goal of SON is to enable network management tasks to be performed *autonomically*. An *automatic* process involves a computing entity working under the supervision of a human operator whom inspects its output and takes decisions, while an *autonomous* process is able to work without any human intervention if it has been properly set up. SON is generally divided into three sub-fields: *self-configuration* (autonomic deployment and configuration), self-optimization (autonomic optimization), and self-healing (autonomic troubleshooting). A fully autonomic configuration is mandatory when deploying femto-cells, which are low power nodes deployed directly in subscribers' houses, since most subscribers are not skilled network engineers. Another motivation for the introduction of SON is the fact that currently deployed networks might not work to their full potential, which could be achieved by better optimization techniques, with a finer granularity. It represents an alternative to deploying more equipment in order to cope with the rising traffic demand. It might be much less costly if SON features can be implemented as software features, without the need to buy and deploy new equipment.

It must be made clear that SON features are *not* a set of sophisticated Radio Resource Management (RRM) algorithms. RRM algorithms take decisions such as: admission of new calls, scheduling and resource allocation, power control, handover of mobile users and so on. It should be noted that RRM algorithms work in a fully autonomic manner. SON features work on a slower time scale in order to control and optimize the parameters of RRM algorithms, with higher level objectives such as block call rate, drop call rate, BS load and cell edge throughput.

#### 1.1.2 SON in standards

Standardization bodies such as the 3rd Generation Partnership Project (3GPP) and pre-standardization bodies such as Next Generation Mobile Network (NGMN) have recognized the importance of SON in wireless networks. They have described the requirements and the key use cases from both a vendor and operator point of view. It is noted that standardization bodies do not discuss algorithmic aspects of SON, and focus on architecture, use cases and associated performance requirements. It is highly likely that algorithms will remain proprietary, since they are one of the main differentiation elements for vendors. It remains unclear whether SON performance can still be ensured in a multi-vendor setting where SONs from multiple vendors run in parallel and interact. NGMN is an alliance comprising major operators, and has reported a list of the key use cases that can be found in [49, 48]. One key requirement from their point of view is the seamless coexistence between SON-enabled networks and legacy networks, and the ability of SON to work in a multi-vendor, multi-technology environment.

3GPP gives the main requirements for SON in [6]. Requirements for self-configuration and self-healing can be found in [5] and [7] respectively. Requirements for self-optimization as well as use cases are described in [4].

Table 1.1 presents the time line of the introduction of the main SON features by the 3GPP in the Long Term Evolution (LTE) standard.

Release 8	Release 9	Release 10	Release 11
• Intra-LTE/frequency	• Inter-system load bal-	• Automatic Neighbor Re-	• Mobility Robustness-
Automatic Neighbor	ancing between LTE and	lation for 3G	Load balancing coordina-
Relation	3G		tion
• Automatic Physical Cell	• Mobility Robustness Op-	• Mobility Robustness Op-	• Inter network, inter-
Identifier selection	timization	timization (enhancement)	frequency Mobility Ro-
			bustness Optimization
• Dynamic configuration	• Random Access Channel	• Mobility Load Balancing	• Minimization of Drive
of the transport network	Optimization (beginning)	(enhancement)	Tests (enhancements)
• Dynamic configuration	• Energy Saving (begin-	• Cell outage compensa-	• Energy Savings (en-
of the X2 interface	ning)	tion	hancements)
• ICIC (beginning)	• Minimization of Drive	• Optimization of param-	×.
	Tests (beginning)	eters due to troubleshoot-	
		ing	
• Intra-system load bal-			
ancing for LTE			
Table	e 1.1. Timeline of the main S	ON features in the LTE stan	dard

### 1.2 Challenges for SON in wireless networks

In this thesis we focus mainly on self-optimization, which represents the most challenging aspect of SON. Self-configuration features are already present in deployed networks, while self-optimization is still problematic. Network operators are indeed reluctant to leaving the control of sensitive network parameters to algorithms without strong guarantees on their performance and stability. We highlight the key SON use cases in wireless networks, and describe what constitutes, in our opinion, the main requirements for adoption of the SON technology by network operators.

#### 1.2.1 The key use cases

#### ICIC

In dense wireless networks, *inter-cell interference* strongly limits capacity and quality of service, and mechanisms to control interference are needed. There are various methods for Inter-Cell Interference Coordination (ICIC). At the physical layer, approaches based on multi-user detection, beamforming and interference alignment and Multiple Input Multiple Output (MIMO) have been studied. At the Medium Access Control (MAC) layer, introducing coordination between BSs to enable scheduling decisions to take into account multiple cells are promising. On a slower time scale, approaches based on *frequency reuse* other than reuse 1 have been shown to provide appreciable capacity gains. SON mechanisms adjust the parameters of those ICIC schemes to maximize the network Quality of Service (QoS).

#### Load balancing

In operational networks, due to irregular cell planning and/or to inhomogeneous traffic (*hot-spots*), some cells tend to be heavily congested while others experience low to medium loads. In such scenarios, a good load balancing mechanism can improve the network performance appreciably, especially since the performance of a network is not evaluated as the average performance of its cells, but rather as the performance of the most congested cells. At the MAC layer, load balancing can be achieved by intelligent *user association*. Namely, when a user enters the network, it might be beneficial not to attach him to the best serving BS if this BS already has a large number of active users, but rather to attach him to a station which might be farther, but is almost empty. At a slower time scale, load balancing can be achieved by adjusting cell sizes. We reduce the size of congested cells so that they serve less traffic, and let their less loaded neighbors serve more traffic. If users attach themselves to the BS with the strongest received pilot power, this can be done by reducing the transmitted pilot powers of the most congested cells. Alternatively, the handover margins can be modified to produce a similar effect.

#### Mobility robustness

When mobile users leave the coverage area of their current serving BSs, they must be handed over to another cell or else they will lose connection and their call will be dropped. The user computes the difference of received pilot powers between his currently serving BS and the BS with the strongest pilot power, and if this difference is larger than a threshold called *handover margin*, then it attempts a handover. In order to avoid a large amount of successive handover between two neighboring BSs, a hysteresis period called *time to trigger* is introduced. It turns out that the values of handover margins and time to trigger have a critical impact on the system performance. In particular, a large amount of dropped calls are due to improper values of those two parameters. From the point of view of the operator, a dropped call is possibly the worst event as far as user perceived QoS is concerned (worse than a blocked call), which is why mobility robustness has been identified as a key use case.

#### Energy savings

Recent studies have shown that wireless access networks have become one of the main consumers of energy for operators, which aroused their interest in so-called *green networking* in which energy is considered a scarce resource and the objective is to minimize the required energy per successfully received bit, rather than maximizing the number of successfully received bits. One approach is to do so is BS *sleep mode*: a fraction of BSs can be switched off when the traffic demand is low and QoS can be ensured with a smaller number of BSs than the number of deployed BSs. SON comes into play because BSs must be switched off autonomically, based on traffic measurements. The parameters of the sleep mode algorithms must tuned carefully in order to avoid creating outage by switching off too many BSs. The sleep mode approach was shown to have especially good performance in Heterogeneous Networks (HetNets), where low power nodes (micro-cells, pico-cells, femto-cells) are deployed in a traditional macro-cell network. Low power nodes are switched on and off dynamically and enable significant reduction of the energy consumption. Standardization activity in 3GPP on this subject can be found in [2].

Use case	Parameters	Performance indicators
ICIC	Transmitted powers	Network capacity, cell-edge throughput
		blocking rate, outage rate
Load balancing	Pilot powers	Network capacity, cell-edge throughput
	Handover margins	blocking rate, outage rate
Mobility robustness	Handover margins,	Dropped call rate,
	time-to-trigger	radio link failures
Energy savings	Transmitted powers	Energy consumption
	BS deactivation	

Table 1.2: The key SON use cases

#### Drive test minimization

When a problem has been identified in a zone of the network, technicians may need to conduct a *drive test* to obtain field measurements with precise localization information, and find the root of the problem. Since this is costly and tedious, and more and more current mobile terminals are equipped with a Global Positioning System (GPS) to provide accurate localization information, it has been suggested to transmit and store mobile measurements and their localization in a database as an alternative to drive tests. One of the possible applications of this database is to build maps of the radio environment (Radio Environment Maps (REMs)) in which the signal attenuation between any BS and any location is available. An accurate REM could be used to perform efficient optimization. The drive test minimization is treated by 3GPP in [1].

Table 1.2 presents the key SON use cases in a synthetic form.

#### 1.2.2 Requirements for SON solutions

We highlight what we feel constitutes the main requirements for SON solutions to be adopted by network operators. We have tried to fulfill those requirements throughout this thesis work.

#### Control plane solutions

The most important requirement is that the SON algorithms should run in the *control plane*, i.e directly in the network equipment such as routers and BSs. Running SON algorithms in the control plane imposes the constraint that the SON solutions can be implemented in a distributed manner, in which each network node can control its own parameters and make decisions based on locally available information. An important advantage of this approach is that it enables to react to traffic variations on the time scale of minutes to tens of minutes. Human network operators cannot react on this time scale, which is why SON is expected to bring gains in this respect. Another important advantage of control plane solutions is to enable optimizing the network at a much finer granularity than what is currently feasible. Namely, in current networks, engineers can at best optimize a critical network parameter for a group of BSs. Some parameters might be technically impossible for a human operator in the OMC to adjust on a per BS basis. Even when the technical possibility exists, optimizing every parameter on a BS is so tedious and time consuming that it might not be feasible in practice. A network with SON mechanisms in the control plane could easily optimize a critical network parameter on a per BS basis. The main goal of SON is to enable optimizing a network at a much finer granularity than what can currently be done even by the most competent engineers, both in time and space. Running SON algorithms in the *management plane* i.e the OMC, would not enable to take decisions with such a fine granularity, and would sacrifice some of the gains to be expected from SON.

#### Stability

From the point of view of the network operator, giving SON algorithms complete control over critical network parameters with little or no means of monitoring and supervising the decisions they take might appear as a huge risk. On the other hand, many real-world systems where failure cannot be tolerated are controlled by algorithms, such as nuclear plants, planes and helicopters. The point is that network operators need very strong guarantees on the functioning of SON algorithms in order to accept taking the risk of deploying a SON-enabled network. One way to provide such guarantees is by providing mathematical proofs of convergence and stability of the proposed SON algorithms. We feel that merely providing simulation results highlighting the SON gains are not enough, and that SON will never be adopted without solid theoretical guarantees of convergence and stability.

#### Robustness to noise

Since we expect SON algorithms to run in real-time at a fast time scale using traffic measurements available at network nodes, the proposed algorithms must be able to work with highly noisy measurements. Robustness to noise is therefore one of the most important requirements for SON. Examples of sources of randomness in networks are: channel variations, user mobility, call arrival and termination. Convergence and stability of SON algorithms is an important concern, and stochastic approximation proves an important tool to study it as shown in this thesis.

#### Signaling and Delay

SON algorithms must be able to run in real time in a distributed fashion, using locally available information. Therefore an important requirement is that the incurred signaling load is small. Neighboring BSs can typically exchange information through an interface (X2 interface in the LTE standard), and a large delay on this interface could prevent SON algorithms from functioning correctly. Hence not only the signaling load must be small, but the frequency at which the algorithm updates the network parameters should be reasonably smaller than the maximal frequency allowed by the interface delay. Typical delay values for current networks are between 5ms and 50ms.

#### Coordination

Most SON algorithms have been designed in a standalone manner where we assume that no other SON algorithm is active at the same time. However, in the long term, a SON-enabled network should feature tenths to hundreds of SON entities active at the same time and interacting. There is no way that network operators will accept deploying such a complex and potentially unstable system, and it is fair to say that analyzing the interaction between multiple SON algorithms and providing efficient coordination mechanisms is currently the most important open problem in SON research. Current research can be split in roughly two approaches: the first approach would be to choose dynamically which SON algorithm to activate at a given time based on Key Performance Indicators (KPIs) and alarms, and the second approach consists in defining an aggregating mechanism to enable all SONs to run in parallel while solving conflicts between them. This thesis work focuses on the second approach. As said previously, the ability to coordinate SONs from multiple vendors is part of the open questions, and the issue of a standardized interface for communication between SONs of different vendors should be discussed.

## **1.3** Our Contribution

#### 1.3.1 Content and methodology

In this thesis, our goal is to study the design, modeling and performance evaluation of SON mechanisms in wireless networks. We propose SON algorithms to solve some of the important use cases listed above. Mathematical models based on queuing theory are proposed. We follow the requirements previously identified for SON solutions: implementability in the control plane, stability, robustness to noise, low signaling overhead and tolerance to delays. This work is *not* simulation-driven: the issues of stability and robustness are studied mathematically. A large part of this thesis is concerned with convergence proofs of the proposed SON algorithms. The topic of coordination is still an open problem which we are currently investigating.

#### 1.3.2 Organization

The remainder of this thesis is organized as follows: in chapter 2 we introduce concisely the required mathematical tools: queuing theory, point processes, information theory, stochastic approximation, Markov decisions processes and reinforcement learning. In chapter 3 we study opportunistic schedulers: their convergence, closed-form formulas for their performance evaluation, and their use to perform coverage-capacity optimization. In chapter 4, an algorithm for automatic load balancing with flow-level dynamics is presented and its convergence is studied. In chapter 5 we study relay-enhanced networks: their dimensioning, algorithms to perform load balancing and dynamical resource allocation using reinforcement learning.

#### 1.3.3 Publications

#### Journal papers

- [J1] R. Combes, Z. Altman, and E. Altman. Scheduling gain for frequencyselective Rayleigh-fading channels with application to self-organizing packet scheduling. *Performance Evaluation*, 68(8):690 – 709, 2011. Special Issue: Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks: selected papers from WiOpt 2010.
- [J2] R. Combes, Z. Altman, and E. Altman. Self-organizing relays: Dimensioning, self-optimization and learning. *IEEE Transactions on Network* and Service Management, 2012.

#### 1.3. OUR CONTRIBUTION

[J3] L. Saker, S.E. Elayoubi, R. Combes, and T. Chahed. Optimal control of wake up mechanisms of femtocells in heterogeneous networks. *IEEE Journal on Selected Areas in Communication (JSAC), special issue on Femtocell Networks*, 2012.

#### Conference papers

- [C1] S. Akbarzadeh, R. Combes, and Z. Altman. Network capacity enhancement of OFDMA system using self-organized femtocell off-load. In *IEEE Wireless Communications and Networking Conference (WCNC 2012)*, april 2012.
- [C2] E. Altman, R. Combes, Z. Altman, and S. Sorin. Routing games in the many players regime. In 4th International ICST Workshop on Game Theory in Communication Networks (GameComm), may 2011.
- [C3] R. Combes, Z. Altman, and E. Altman. On the use of packet scheduling in self-optimization processes: Application to coverage-capacity optimization. In 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010, pages 98-107, june 2010.
- [C4] R. Combes, Z. Altman, and E. Altman. A self-optimization method for coverage-capacity optimization in OFDMA networks with MIMO. In 5th ICST International Conference on Performance Evaluation Methodologies and Tools (ValueTools), 2011, may 2011.
- [C5] R. Combes, Z. Altman, and E. Altman. Self-organizing fractional power control for interference coordination in OFDMA networks. In *IEEE International Conference on Communications (ICC)*, 2011, june 2011.
- [C6] R. Combes, Z. Altman, and E. Altman. Self-organizing relays in LTE networks: Queuing analysis and algorithms. In 7th International Conference on Network and Service Management (CNSM), 2011, Best Paper Award, october 2011.
- [C7] R. Combes, Z. Altman, and E. Altman. Self-organization in wireless networks: a flow-level perspective. In *The 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM* 2012), april 2012.
- [C8] R. Combes, Z. Altman, and E. Altman. Interference coordination in wireless networks: a flow-level perspective. In *IEEE INFOCOM*, april 2013.

- [C9] R. Combes, Z. Altman, M. Haddad, and E. Altman. Self-optimizing strategies for interference coordination in OFDMA networks. In *IEEE International Conference on Communications Workshops (ICC)*, 2011, june 2011.
- [C10] R. Combes, S.E Elayoubi, and Z. Altman. Cross-layer analysis of scheduling gains: Application to lmmse receivers in frequency-selective rayleigh-fading channels. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011*, pages 133–139, may 2011.

#### Patents

- [P1] R. Combes, S. Akbarzadeh, and Z. Altman. Base station having a plurality of antennas for the adaptative offloading of an overloaded area. patent WO2013001218, jan 2013.
- [P2] R. Combes and Z. Altman. Method for controlling power in mobile networks. patent WO2012175869, dec 2012.

#### Under peer review

- [I1] S. Akbarzadeh, R. Combes, and Z. Altman. Enhancing network capacity in the downlink of a cellular network using self-organized femtocell offloading. submitted to International Journal of Network Management, 2012.
- [I2] R. Combes, Z. Altman, and E. Altman. Coordination of autonomic functionalities in communication networks. submitted to WiOpt, 2013.
- [I3] V. Kavitha and R. Combes. Continuous polling with rerouting: Performance and modeling of ferry assisted wireless lans. under revision in Performance Evaluation, 2012.

#### In progress

[W1] R. Combes, I. El Bouloumi, S. Sénécal, and Z. Altman. The association problem in wireless networks: a policy gradient reinforcement learning approach. in progress, 2012.

# Chapter 2

# Theoretical foundations

In this chapter we give a basic introduction to several mathematical tools used in the design and evaluation of SON mechanisms. We feel that although SON research has an important practical component, it is necessary to use solid theoretical foundations in order to justify the design and prove the convergence of the proposed algorithms. Our aim is that this thesis is selfcontained. Results are stated without proofs, and the reader can consult the references given for a complete exposition of the topics. This chapter can be used in two ways: the reader can either read it in at one go in order to have a panorama of the mathematical tools used, or skip this chapter in a first reading and come back to it after reading the results of the next chapters to understand their proofs.

## 2.1 Queuing theory

We present several queuing theory results used for the modeling of wireless networks. We present a short introduction to point processes since they are an important component of the modern approach to queuing.

#### 2.1.1 Point processes

#### Marked point processes

Point processes are random collections of points in a measurable space. They are used for modeling punctual phenomena such as random arrivals of users in a queuing system. A complete exposition can be found in [30], and a concise introduction is found in [31]. We consider M a complete separable metric space equipped with its Borel  $\sigma$ -algebra. We call configuration a countable collection of points of M,  $\mathcal{X} = \{t_n\}_{n \in \mathbb{Z}}$  and we denote by  $\mathcal{C}$  the space of

configurations. For configuration  $\mathcal{X}$  and  $B \subset M$  a Borel set, we define the number of points of  $\mathcal{X}$  that fall in B:

$$N_{\mathcal{X}}(B) = \sum_{n \in \mathbb{Z}} \mathbf{1}_B(t_n).$$
(2.1)

A configuration is *locally finite* if  $N_{\mathcal{X}}(B) < +\infty$  when *B* is bounded. A configuration is *simple* if its points are disjoint. A *point process* is a mapping from a probability space  $(\Omega, \mathcal{F}, \mathbf{P})$  with values in the space of locally finite configurations, equipped with the smallest  $\sigma$ -algebra which makes  $\mathcal{X} \to N_{\mathcal{X}}(B)$  measurable. A simple point process is a point process with values in the space of disjoint configurations.

A realization of a point process is  $\{T_n(\omega)\}_{n\in\mathbb{Z}}$ , a countable collection of points in M. We denote by:

$$N(B) = \sum_{n \in \mathbb{Z}} \mathbf{1}_B(T_n), \qquad (2.2)$$

the number of points that fall in B. It can be proven that the distribution of a point process is full determined by its finite-dimensional distributions  $N(B_1), \ldots, N(B_K)$  with  $(B_1, \ldots, B_K)$ , bounded Borel sets. A remarkable simplification exists for simple point processes: their distribution if fully determined by the *void probabilities*  $\mathbf{P}[N(B) = 0]$ , for all B Borel sets.

We define marks attached to points of the point process  $\{\sigma_n(\omega)\}_{n\in\mathbb{Z}}$  with values in a complete separable metric space Q.  $\{T_n(\omega), \sigma_n(\omega)\}_{n\in\mathbb{Z}}$  is called a marked point process. In the context of queuing theory,  $T_n$  denotes the instants of arrival of the *n*-th customer, and  $\sigma_n$  his service requirement.

#### Campbell formulas

The *first-order measure* of the point process is the average number of points falling in a Borel set, namely:

$$m(B) = E[N(B)].$$
(2.3)

We assume that m is finite on bounded Borel sets, so that it indeed defines a measure on M. A fundamental result is the *Campbell formula*:

$$E\left[\sum_{n\in\mathbb{Z}}f(T_n)\right] = \int_M f(t)m(dt).$$
(2.4)

with  $f: M \to \mathbb{R}$  non-negative and measurable. The formula is true for  $f = \mathbf{1}_B$  by definition of m. By linearity and monotonicity, the formula is also true for all non-negative measurable functions.

#### 2.1. QUEUING THEORY

The first-order measure alone does not define a point process completely, and large families of different point processes share the same first-order measure. We can generalize the definition of m to obtain the K-th moment measure:

$$m_K(B_1 \times \ldots \times B_K) = E\left[N(B_1) \times \ldots \times N(B_K)\right].$$
(2.5)

Using the same arguments of linearity and monotonicity, we obtain a K-th order version of the Campbell formula:

$$E\left[\sum_{(n_1,\dots,n_K)\in\mathbb{Z}^K} f(T_{n_1},\dots,T_{n_K})\right] = \int_{M^K} f(t_1,\dots,t_K) m_K(dt_1,\dots,dt_K).$$
(2.6)

with  $f: M^K \to \mathbb{R}$  non-negative and measurable.

#### Poisson point process

The Poisson process is the simplest point process. Given m a measure on M, the Poisson process is the unique point process such that N(B) is a Poisson random variable with parameter m(B) and  $(N(B_1), \ldots, N(B_K))$  are independent for  $(B_1, \ldots, B_K)$  disjoint Borel sets. The Poisson process is completely described by its first-order measure i.e two Poisson processes with the same first-order measure have the same distribution.

#### Stationary ergodic point processes

Assume that M is an Euclidean space. Given  $t \in M$ , we define the *shift* operator  $\theta_t : \Omega \to \Omega$ , such that

$$\{T_n(\theta_t \circ \omega), \sigma_n(\theta_t \circ \omega)\}_{n \in \mathbb{Z}} = \{T_n(\omega) - t, \sigma_n(\omega)\}_{n \in \mathbb{Z}}.$$
 (2.7)

Namely  $\theta_t$  shifts all the points of the point process of -t, and the marks "follow" the points to which they are attached.

A point process is *stationary* if its distribution is invariant by  $\theta_t$ ,  $t \in M$ . The measure m of a stationary point process is finite and translation invariant, so it must be proportional to the Lebesgue measure. Namely

$$m(dt) = m_0 dt, \tag{2.8}$$

and  $m_0$  is called the *intensity* of the point process.

A point process is *ergodic* if  $\theta_t$  is an ergodic transformation of  $(\Omega, \mathcal{F}, \mathbf{P})$ . We recall that  $\theta_t$  is an *ergodic transformation* if it is:

- measure preserving:  $\mathbf{P}[\theta_t(E)] = \mathbf{P}[E]$ , for all  $E \subset \Omega$
- admits no invariant sets except  $\emptyset$  and  $\Omega$ : if  $\theta_t(E) = E$  then  $E = \emptyset$  or  $E = \Omega$ .

#### Campbell measure and Palm probability

Consider a marked stationary point process, B and C Borel sets of M and Q respectively. A quantity of interest is the average number of marks  $\sigma_n$  falling in C, providing that the point to which they are attached  $T_n$  falls in B. This quantity is called the *Campbell measure*:

$$c_{\sigma}(B \times C) = E\left[\sum_{n \in \mathbb{Z}} \mathbf{1}_B(T_n) \mathbf{1}_C(\sigma_n)\right].$$
 (2.9)

In the context of queuing, we are interested in the *load*, which is the expected workload arriving at the server during a unit of time. The load is expressed in terms of the Campbell measure:

$$\rho = \int_Q \sigma c_\sigma([0,1] \times d\sigma). \tag{2.10}$$

From stationarity of the point process,  $B \mapsto c_{\sigma}(B \times C)$  is translation invariant so it must be proportional to the Lebesgue measure. Furthermore,

 $c_{\sigma}(B \times Q) = m(B)$ . This allows to define the *Palm distribution* of the marks:

$$\nu_{\sigma}(C) = \frac{c_{\sigma}(B \times C)}{m(B)}.$$
(2.11)

It is noted that the right hand side does not depend on B. In particular, take B as a ball of arbitrarily small radius centered on 0. The intuitive meaning of  $\nu_{\sigma}(C)$  is the probability of a mark  $\sigma_0$  falling in C, conditional to the fact that the point process has a point at 0. We denote by  $E_T^0$  the Palm expectation, which is the expectation with respect to the measure  $\nu_{\sigma}$ .

Going back to our queuing example, by definition of the Palm probability, the load is expressed simply in terms of the Palm expectation of the marks:

$$\rho = m([0,1]) \int_{Q} \sigma \nu_{\sigma}(d\sigma) = m_0 E_T^0(\sigma_0).$$
(2.12)

In general there is a dependency between the marks and the points of the process, so that the Palm expectation of the marks  $E_T^0(\sigma_0)$  is not equal to the expectation of the marks  $E[\sigma_0]$ . This fact is known as the *hitchhiker's paradox*.

#### Loynes theorem

The stability of a single queue with stationary ergodic input is given by *Loynes theorem*, first proven in [45]. A demonstration can be found (for

instance) in [9]. We call W(t) the *workload* at time t which is the sum of the remaining service times of the active users at time t. We assume the queue to be *work-conserving*: W(t) diminishes at speed 1 for all t such that W(t) > 0 and there is no arrival at time t.

Theorem 2.1. The queue is stable if:

$$\rho < 1, \tag{2.13}$$

in the sense that there exists a unique finite workload process almost surely (a.s). If  $\rho > 1$ , no finite workload process exists a.s and the queue is unstable.

#### 2.1.2 Classical queues

We expose two queuing models which are popular for the modeling of wireless networks: the M/G/1 Processor Sharing (PS) for elastic traffic (web and File Transfer Protocol (FTP) traffic), and the Multi-rate Erlang for constant data rate traffic (voice and streaming traffic). The reader can refer to [15] for a more complete description.

#### Multi-rate Erlang

The multi-rate Erlang model is an extension of the well-known Erlang model (the M/M/C/C queue) where the number of circuits required by a customer is arbitrary. There are N classes of customers and a number of circuits C. Customers of class *i* arrive at a server according to a Poisson process of rate  $\lambda_i$ , they require  $c_i$  circuits, and stay in the service an exponentially distributed amount of time with parameter  $\mu_i$ . The traffic intensity generated by users of class *i* is:

$$\alpha_i = \frac{\lambda_i}{\mu_i}.\tag{2.14}$$

The *load* of the server is the average number of required circuits per unit of time divided by the number of circuits:

$$\rho = \frac{1}{C} \sum_{i=1}^{N} \alpha_i c_i. \tag{2.15}$$

Write  $x \in \mathbb{N}^N$  the state of the server i.e the number of users of each class present in the server. The system is *lossy*: when a user arrives in the server and there are not enough circuits to serve him, he is rejected. We want to evaluate the *blocking rate* which is the proportion of rejected users. The constraint on the state is written:

$$\langle x , c \rangle = \sum_{i=1}^{N} c_i x_i \le C \tag{2.16}$$

First consider the case where C is infinite. The stationary distribution  $\pi$  is known up to a constant by reversibility:

$$\pi(x) = \pi(0) \prod_{i=1}^{N} \frac{\alpha_i^{x_i}}{x_i!}, x \in \mathbb{N}^N.$$
(2.17)

When C is finite, the stationary distribution is obtained by truncation and normalization:

$$\pi(x) = \pi(0) \prod_{i=1}^{N} \frac{\alpha_i^{x_i}}{x_i!} , \ \langle x \,, \, c \rangle \le C.$$
(2.18)

The probability  $B_i$  of a user of class *i* to be blocked is:

$$B_i = \pi(0) \sum_{C - c_i < \langle x, c \rangle \le C} \pi(x).$$
(2.19)

The number of terms in 2.19 grows exponentially in N. Define:

$$f(n) = \sum_{\langle x, c \rangle = n} \prod_{i=1}^{N} \frac{\alpha_i^{x_i}}{x_i!},$$
(2.20)

then the blocking probabilities can be expressed as:

$$B_{i} = \frac{\sum_{C-c_{i} < n \le C} f(n)}{\sum_{0 \le n \le C} f(n)}.$$
(2.21)

The value of f can be calculated recursively by:

$$f(n) = \frac{1}{n} \sum_{i=1}^{N} \alpha_i c_i f(n - c_i), \qquad (2.22)$$

with f(0) = 1 and f(n) = 0 if n < 1. The relation 2.22 allows to calculate the value of  $B_i$  with linear complexity in N. This algorithm for calculating the blocking probabilities is called the *Kaufman-Roberts algorithm*.
#### 2.2. INFORMATION THEORY

## M/G/1 PS

The M/G/1 PS queue was originally studied in the context of multi-thread processors ([37]). As previously, there are N classes of users. Users of class i arrive at a server according to a Poisson process with arrival rate  $\lambda_i$  and require an average amount of service  $\frac{1}{\mu_i}$ . Write  $x \in \mathbb{N}^N$  the state of the server. Users are served on a *best-effort* basis, namely in state x each user is served at speed  $\frac{1}{\sum_{i=1}^{N} x_i}$ . Define  $\rho_i$  the *load* generated by users of class i:

$$\rho_i = \frac{\lambda_i}{\mu_i},\tag{2.23}$$

and the load of the server:

$$\rho = \sum_{i=1}^{N} \rho_i. \tag{2.24}$$

The stability condition is:

$$\rho < 1. \tag{2.25}$$

First assume that the service requirements are exponentially distributed. From reversibility, the stationary distribution is known up to a constant. The constant is obtained by summation over the state space:

$$\pi(x) = (1-\rho) \frac{(\sum_{i=1}^{N} x_i)!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} \rho_i^{x_i}.$$
(2.26)

The average number of active users has the simple form:

$$E[x] = \frac{\rho}{1-\rho}.$$
(2.27)

It can actually be shown that when the assumption that the service requirements are exponentially distributed is removed the results above remain true, so that the stationary distribution only depends on the average service requirements but not on their distribution. This property is called *insensitivity*.

# 2.2 Information theory

In this section we expose the basic concepts of information theory, in particular the theorems proven by Shannon in [54]: the *source coding theorem* and the *noisy channel coding theorem*. The Shannon theorems give a rigorous definition of the amount of information contained in a signal, and give capacity of a communications channel which is the maximal rate at which information can be transmitted through so that an arbitrary low probability of error can be ensured. The *capacity* represents the physical limit of a channel. A complete exposition can be found for instance in [29].

For the purpose of this thesis, those results enable to evaluate the performance of the wireless link between a BS and a user at a given location and serve as inputs to various models of packet scheduling and evolution of the number of active users in a network. To keep the exposition simple we describe the theory for discrete memoryless channels. For continuous channels, we simply state the results used in the thesis work.

# 2.2.1 Source coding theorem

#### Entropy, conditional entropy and mutual information

We consider random variables with values in a finite set  $\mathcal{X}$ . A fundamental quantity in information theory is the *entropy* of a random variable. which is, from an intuitive point of view, a measure of randomness of the distribution of X.

**Definition 2.1.** For random variable X with distribution p, the entropy of X denoted H(X) is defined as:

$$H(X) = -E\left[\log_2 p(X)\right] = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$
(2.28)

H measures the randomness of the distribution of X, as shown by three basic properties: entropy is null if and only if (iif) X is constant, entropy is maximal if X is uniformly distributed, and entropy is additive for independent random variables:

**Property 2.1.**  $-H(X) \ge 0$  with equality iff X is constant,

- $-H(X) \leq \log_2(|\mathcal{X}|)$  with equality if X is uniformly distributed on  $\mathcal{X}$ ,
- $H(X_1,\ldots,X_N) = \sum_{n=1}^N H(X_n)$  if  $(X_1,\ldots,X_N)$  are independent.

Another quantity of interest is the *conditional entropy* H(X|Y) which measures the randomness of the distribution of X once the value of Y is known.

**Definition 2.2.** The entropy of X conditional to Y is denoted H(X|Y) and is defined as:

$$H(X|Y) = H(X,Y) - H(Y).$$
 (2.29)

#### 2.2. INFORMATION THEORY

From the positivity of entropy, we deduce that conditioning reduces entropy:

$$H(X|Y) \le H(X)$$
,  $H(X|Y) = H(X) \iff X \perp Y.$  (2.30)

Indeed, if Y is highly correlated to X, observing Y almost fully determines the value of X and H(X|Y) is much smaller than H(X). By successive conditioning, we deduce the entropy chain rule:

$$H(X_1, \dots, X_N) = \sum_{n=1}^N H(X_n | X_1, \dots, X_{n-1}).$$
 (2.31)

Assume that X represents a quantity we are trying to estimate, based on the observation of another quantity Y. We define  $\hat{X}$  the estimator which associates  $\hat{X}(Y)$  the estimated value of X to the observed value of Y. The estimation error is the probability of the event:

$$P_e = \mathbf{P} \left[ \hat{X}(Y) \neq X \right]. \tag{2.32}$$

If H(X|Y) is large, X is still "very random" once the value of Y is known, so that even the best estimator must have a large probability of error. This intuitive link between conditional entropy and estimation error is illustrated by *Fano's inequality*:

**Theorem 2.2.** For any estimator  $\hat{X}$ , we have that:

$$H_2(P_e) + P_e \log_2(|\mathcal{X}|) \ge H(X|Y),$$
 (2.33)

with  $H_2(p) = p \log_2(p) + (1-p) \log_2(1-p)$ .

A third quantity of interest is the *mutual information*, which is linked to conditional entropy and measures the amount of information about the value of a random variable which can be inferred from the observation of another random variable.

**Definition 2.3.** The mutual information between X and Y is denoted I(X, Y) and is defined as:

$$I(X;Y) = H(X) - H(Y|X).$$
(2.34)

The mutual information I(X; Y) has several properties which make it a reasonable measure of information: it is positive, symmetrical, and null iff Xand Y independent.

Property 2.2. 
$$-I(X;Y) = I(Y;X)$$
  
 $-I(X;Y) \ge 0$  with equality iff  $X \perp Y$ .

From the *entropy chain rule*, we deduce a similar chain rule for mutual informations:

$$I(X_1, \dots, X_N; Y) = \sum_{n=1}^{N} I(X_n; Y | X_1, \dots, X_{n-1}).$$
(2.35)

The chain rule for mutual information has the interpretation that the information contained the vector  $(X_1, X_2)$  is also the sum of the information contained in  $X_1$  when  $X_2$  is unknown, and the information contained in  $X_2$ when  $X_1$  is known. *Multi-user detection*, in which several interfering signals are decoded iteratively and subtracted to the total received signal, is based on this principle.

# AEP for i.i.d sources

We consider  $X^N = (X_1, \ldots, X_N)$  an independent and identically distributed (i.i.d) sample with distribution p of size N.  $X^N$  is an element of  $\mathcal{X}^N$ , which has cardinality  $2^{N \log_2(|\mathcal{X}||)}$ . However, when N goes to infinity, we can show that with high probability  $X^N$  falls in a subset of  $\mathcal{X}^N$  which has a cardinality of  $2^{NH(X)}$ . This property is a consequence of the law of large numbers, and is called Asymptotic Equi-repartition Property (AEP) in information theory. It forms the basis of the proofs for both Shannon theorems.

**Definition 2.4.** The typical set with error margin  $\epsilon$  is defined as:

$$A_{\epsilon}^{N} = \{ x \in \mathcal{X}^{N} : |\frac{1}{N} \sum_{n=1}^{N} \log_2 p(x_n) + H(X)| \le \epsilon \}$$
(2.36)

From the (weak) law of large numbers, we have that:

$$-\frac{1}{N}\sum_{n=1}^{N}\log_2 p(X_n) \xrightarrow[N \to +\infty]{\mathbf{P}} H(X), \qquad (2.37)$$

which proves that the distribution of  $\mathcal{X}^N$  is concentrated on the typical set  $A_{\epsilon}^N$  when  $N \to +\infty$ .

**Theorem 2.3.** For all  $\epsilon > 0$  we have that:

$$\mathbf{P}\left[X^{N} \in A_{\epsilon}^{N}\right] \underset{N \to +\infty}{\to} 1 \tag{2.38}$$

and:

$$|A_{\epsilon}^{N}| \le 2^{N(H(X)+\epsilon)}.$$
(2.39)

There exists  $N_0$  such that:

$$|A_{\epsilon}^{N}| \ge (1-\epsilon)2^{N(H(X)-\epsilon)}, \ N \ge N_{0}$$
 (2.40)

### 2.2. INFORMATION THEORY

The AEP in Theorem 2.3 states that for large N, the distribution of  $X^N$  concentrates itself on the typical set  $A_{\epsilon}^N$ . Furthermore, the cardinal of the typical set is equivalent to  $2^{NH(X)}$ . The AEP justifies the interpretation of entropy as a measure of randomness since it measures the volume of the support of a probability distribution.

### Source coding for finite alphabets

The AEP points out that, whenever  $H(X) < \log_2(|\mathcal{X}|)$ ,  $X^N$  lies with a high probability in a set whose cardinality is much smaller than  $2^{N \log_2|\mathcal{X}|}$ . Therefore,  $X^N$  which is represented by  $N \log_2|\mathcal{X}|$  bits can be compressed and represented by NH(X) bits, up to an arbitrarily small probability of error. This is the idea behind the source coding theorem.

**Definition 2.5.** A source code C of length N with rate R is a collection of  $2^{NR}$  distinct elements of  $\mathcal{X}^N$  called code words  $Y_k^N \in \mathcal{X}^N$ ,  $1 \le k \le 2^{NR}$ .

Source coding is performed by observing a word  $X^N$  of length N emitted by the source, and transforming it into a number between 1 and K noted  $C(X^N)$ .

$$C(X^{N}) = k, \text{ if } \exists k, X^{N} = Y_{k}^{N},$$
  

$$C(X^{N}) = 0, \text{ otherwise }.$$
(2.41)

If there exists  $1 \leq k \leq K$  such that  $Y_k^N = X^N$ , then  $C(X^N) = k$ , otherwise  $C(X^N) = 0$ . Since the code words are distinct, the value of  $X^N$ can always be deduced from  $C(X^N)$  as long as  $C(X^N) \neq 0$ . The probability of error is the probability of outputting a 0:

$$P_e = \mathbf{P} \left[ C(X^N) = 0 \right]. \tag{2.42}$$

**Definition 2.6.** A rate R is said to be achievable if there exists a sequence of codes whose rates are smaller or equal to R and whose probability of error vanishes.

It is noted that  $R = |\mathcal{X}|$  is always achievable, and that if R is achievable, so is  $R' \geq R$ . The source coding gives another interpretation of entropy as the number of bits per source symbol required to represent the words emitted by the source.

**Theorem 2.4.** All rates strictly above H(X) are achievable, and all rates strictly below H(X) are not achievable.

This is a surprising result: no matter how smart a coding scheme is, data cannot be compressed beyond its entropy H(X). Furthermore, the proof of this theorem is constructive. Namely, choose a sequence of codes so that the codewords of the *N*-th code are the elements of the typical set  $A_{\epsilon}^{N}$ . Recall that  $|A_{\epsilon}^{N}| \leq 2^{N(H(X)+\epsilon)}$  so the rate of the *N*-th code is smaller than  $H(X) + \epsilon$ . The probability of error of the *N*-th code is  $\mathbf{P}\left[X_{N} \notin A_{\epsilon}^{N}\right]$ which vanishes when  $N \to +\infty$  using Theorem 2.3. The converse is proven by showing that the typical set is the "smallest set" (up to a negligible error) to contain all the distribution of  $X_{N}$ .

# 2.2.2 Noisy channel coding theorem

#### Communication channel

A communication channel is defined as a relationship between an input sequence  $\{X_n\}_{n\in\mathbb{N}}$  with values in  $\mathcal{X}$  and an output sequence  $\{Y_n\}_{n\in\mathbb{N}}$  with values in  $\mathcal{Y}$ .  $\{X_n\}_{n\in\mathbb{N}}$  denotes a sequence of messages sent by a source to a receiver, whose job is to infer it from the channel output  $\{Y_n\}_{n\in\mathbb{N}}$ . We consider discrete and memoryless channels. A channel is *discrete* if  $\mathcal{X}$  and  $\mathcal{Y}$  are both finite. A channel is *memoryless* if the distribution of  $Y_n$  depends only on  $X_n$ ,  $n \in \mathbb{N}$ .

A discrete memoryless channel is completely specified by the distribution of the output symbol Y conditioned on the input symbol X:

$$p(y|x) = \mathbf{P}\left[Y = y|X = x\right], \ (x, y) \in \mathcal{X} \times \mathcal{Y}.$$
(2.43)

A transmission scheme is specified by a *channel code*.

**Definition 2.7.** A channel code C of length N with rate R is composed by a coding function  $X^N : \{1, \ldots, 2^{NR}\} \mapsto \mathcal{X}^N$ , and a decoding function  $g : \mathcal{X}^N \mapsto \{0\} \cup \{1, \ldots, 2^{NR}\}$ . The set of codewords is  $\{X^N(1), \ldots, X^N(2^{NR})\}$ .

Transmission over the channel is done the following way: the source chooses a message  $x \in \{1, \ldots, 2^{NR}\}$ , encodes it into  $X^N(x)$  and transmits it through the channel. The decoder receives the corresponding channel output  $X^N(x)$ , and estimates the message  $g(X^N(x))$ . The decoding is successful if  $x = g(X^N(x))$ . The conditional probability of error is:

$$P_e(x) = \mathbf{P} \left[ X^N(x) \neq x \right], \qquad (2.44)$$

and the average probability of error, given a distribution p on the messages:

$$\overline{P}_e = p(x)P_e(x). \tag{2.45}$$

#### 2.2. INFORMATION THEORY

**Definition 2.8.** A rate R is said to be achievable if there exists a sequence of codes whose rates are larger or equal to R and whose probability of error vanishes.

**Definition 2.9.** The channel information capacity is defined as the maximal mutual information:

$$C = \max_{n} I(X;Y), \tag{2.46}$$

where the maximum is taken over all the distributions of the input X.

We can prove that C indeed defines the capacity of a channel, namely C is the maximal achievable rate. It is noted that  $p \to I(X;Y)$  is a concave function, so that the maximization problem has a unique solution and can be solved efficiently using standard *convex optimization* techniques ([20]).

### AEP for discrete memoryless channels

We want to deduce the channel input from its output, and from Fano's inequality there is no way to retrieve the value of  $X_n$  from  $Y_n$  without error unless H(Y|X) is null. We proceed as in the source coding problem: we consider input words of length  $N X^N = (X_1, \ldots, X_N)$ , and the corresponding output  $Y^N = (Y_1, \ldots, Y_N)$ . The input-output vector  $(X^N, Y^N)$  is hence an i.i.d sample drawn from the joint distribution distribution p(x, y). We study the asymptotic properties of the distribution of  $(X^N, Y^N)$ .

**Definition 2.10.** We define the typical set:

$$A_{\epsilon}^{N} = \{(x, y) \in \mathcal{X}^{N} \times \mathcal{Y}^{N} : \left| \frac{1}{N} \sum_{n=1}^{N} \log_{2} p(x_{n}) + H(X) \right| \leq \epsilon ,$$
$$\left| \frac{1}{N} \sum_{n=1}^{N} \log_{2} p(y_{n}) + H(Y) \right| \leq \epsilon |$$
$$\left| \frac{1}{N} \sum_{n=1}^{N} \log_{2} p(x_{n}, y_{n}) + H(X, Y) \right| \leq \epsilon \} \quad (2.47)$$

As in the case of source coding, the typical set describes the distribution of  $(X^N, Y^N)$  for large N. Namely, if  $(X^N, Y^N)$  is an i.i.d sample with distribution p(x, y), then  $(X^N, Y^N) \in A^N_{\epsilon}$  with high probability. On the other hand, if  $(X^N, Y^N)$  is an i.i.d sample with distribution p(x)p(y), then  $(X^N, Y^N) \in A^N_{\epsilon}$  with small probability.

**Theorem 2.5.** If  $(X^N, Y^N)$  is i.i.d with distribution p(x, y), for all  $\epsilon > 0$  we have that:

$$\mathbf{P}\left[(X^N, Y^N) \in A^N_\epsilon\right] \underset{N \to +\infty}{\to} 1.$$
(2.48)

Furthermore:

$$|A_{\epsilon}^{N}| \le 2^{N(H(X,Y)+\epsilon)}.$$
(2.49)

If  $(\tilde{X}^N, \tilde{Y}^N)$  is i.i.d with distribution p(x)p(y), for all  $\epsilon > 0$  we have that:

$$\mathbf{P}\left[\left(\tilde{X}^{N}, \tilde{Y}^{N}\right) \in A_{\epsilon}^{N}\right] \le 2^{-N(I(X;Y)-3\epsilon)}.$$
(2.50)

# Noisy channel coding theorem

Theorem 2.5 points out a coding scheme based on typical sets and this coding scheme in fact achieves capacity.

#### **Definition 2.11.** The information capacity C is the maximal achievable rate.

As in the case of source coding, the proof is constructive. Given R < Cwe construct a sequence of channel codes of length N with rate R. For length N, the codewords are chosen by drawing  $2^{NR}$  i.i.d samples of size N with distribution p(x). Decoding is based on *joint typicality*:

$$g(Y^N) = x \text{ if } \exists x, (X^N(x), Y_n) \in A^N_{\epsilon},$$
  

$$g(Y^N) = 0 \text{ otherwise.}$$
(2.51)

We can assume that the message transmitted is equal to 1 without loss of generality by symmetry of the construction of the code. Furthermore  $P_e(1) = \overline{P}$  using the same argument. The conditional probability of error for message 1 is the probability that  $(X^N(1), Y_n) \notin A_{\epsilon}^N$ , or that there exists  $x \neq 1$  such that  $(X^N(x), Y_n) \in A_{\epsilon}^N$ .

Using (2.48), for large N:

$$\mathbf{P}\left[\left(X^{N}(1), Y_{n}\right) \notin A_{\epsilon}^{N}\right] \leq \epsilon.$$
(2.52)

Since the codewords are independent,  $(X^N(x), Y_n)$ ,  $x \neq 1$  has distribution p(x)p(y), and using (2.50):

$$\mathbf{P}\left[(X^{N}(x), Y_{n}) \in A_{\epsilon}^{N}\right] \le 2^{-N(I(X;Y)-3\epsilon)}, x \in \{2, \dots, 2^{NR}\}.$$
(2.53)

Hence the conditional error probability can be bounded by:

$$P_e(1) \le \epsilon + (2^{NR} - 1)2^{-N((I(X;Y) - R) - 3\epsilon)}.$$
(2.54)

Hence whenever R < I(X;Y),  $P_e(1)$  can be made arbitrarily small when N is large enough. Choosing the distribution of X to maximize the mutual information, it proves that all rates below capacity are achievable.

44

It is noted that although this coding scheme allows to achieve capacity, it is of very little use in practice since its complexity is exponential: decoding a codeword of length N requires at least  $2^{NR}$  operations. The question of designing codes that enable reaching the Shannon bound while keeping decoding complexity at a minimum is much more complex and is investigated by the field of *coding theory*.

# 2.2.3 Continuous channels

Continuous channels models allow to describe a channel in a much more concise and elegant manner than discrete models, mostly due to the continuous nature of the physical media used for transmission of information. Both  $\mathcal{X}$  are  $\mathcal{Y}$  lie in some Euclidean space. Entropy and mutual information can be defined in a similar way as in the discrete case, so that both the source coding and noisy channel coding theorem hold. In particular, the mutual information between the channel input and output represents the largest rate at which information can be transmitted over the channel with vanishing probability of error. The differential entropy is defined as:

# Definition 2.12.

$$H(X) = -\int_{\mathcal{X}} p(x) \log_2 p(x) dx \qquad (2.55)$$

The *mutual information* is defined as in the discrete case:

### Definition 2.13.

$$I(X;Y) = H(X) - H(Y|X).$$
(2.56)

Any partition  $\mathcal{X} \times \mathcal{Y}$  induces a corresponding discretized version of the continuous channel. The continuous definition of mutual information has the following intuitive interpretation: it can be seen as the maximum of the (discrete) mutual information over all partitions of  $\mathcal{X} \times \mathcal{Y}$ .

### Property 2.3.

$$I(X,Y) = \sup_{\tilde{X},\tilde{Y}} I(\tilde{X},\tilde{Y})$$
(2.57)

where (X, Y) denotes a discretized version of (X, Y) according to some partition of  $\mathcal{X} \times \mathcal{Y}$ .

# 2.2.4 Channel models

# AWGN

The simplest channel model is the Additive White Gaussian Noise (AWGN) channel.

**Definition 2.14.** The AWGN channel is a memoryless channel such that:

$$y[t] = x[t] + n[t], t \in \mathbb{N}$$
 (2.58)

where  $t \to n[t]$  is i.i.d Gaussian and  $n[t] \equiv \mathcal{N}(0, N)$ .

If we are allowed to transmit at arbitrary large power, it is intuitive that the capacity of the channel is infinite since we can distinguish an infinite number of inputs from the output. Therefore we introduce the constraint that  $E[x^2] \leq P$  with P the maximal transmit power. The channel capacity is given by maximizing the mutual information:

$$C = \max_{E[x^2] \le P} I(x; y) = \frac{1}{2} \log_2(1 + \frac{P}{N}).$$
(2.59)

N is the noise power of the channel.  $\frac{P}{N}$  is called Signal to Noise Ratio (SNR). To reach the maximum, the input of the channel must have a Gaussian distribution  $x \equiv \mathcal{N}(0, P)$ .

#### Parallel AWGN channels

A classical result is the capacity of parallel AWGN channels. Assume K parallel AWGN channels with  $N_k$  the noise power of the k-th channel and  $P_k$  the power allocated to the k-th channel. The power constraints are  $P_k \ge 0$  and  $\sum_{k=1}^{K} P_k \le P$ . The capacity is derived by solving a convex optimization problem, and writing the Karush-Kuhn-Tucker (KKT) conditions shows that there exists a constant  $\lambda > 0$  such that the power allocation maximizing the total capacity is:

$$P_k = \max(\frac{1}{\lambda} - N_k, 0).$$
 (2.60)

This result is known as *water-filling*: channels with noise power superior to  $\frac{1}{\lambda}$  are not allocated any power, and the power allocated to other channels decreases with their noise power such that  $\frac{1}{\lambda} - N_k$  is a constant.

#### Band-limited AWGN

We define a continuous-time version of the AWGN channel.

**Definition 2.15.** The band-limited continuous time AWGN channel is a channel with input-output relationship:

$$y(t) = x(t) + n(t), t \in \mathbb{R}$$

$$(2.61)$$

where both x and y are band-limited signals with bandwidth W and n(t) is a white Gaussian noise process with spectral power density  $N_0$ .

### 2.2. INFORMATION THEORY

From the Nyquist-Shannon sampling theorem, this channel is equivalent to 2W parallel (discrete-time) AWGN channels obtained by sampling x and y every  $\frac{1}{2W}$  seconds. The capacity of the band-limited AWGN channel is therefore:

$$C = W \log_2(1 + \frac{P}{WN_0}), \qquad (2.62)$$

which is perhaps the most famous information theory result. In the reminder we always work with discrete time channels, since a band-limited continuous time channel can be reduced to a discrete time equivalent channel.

### AWGN capacity in practical systems

From the demonstration of the noisy channel coding theorem (theorem 2.11), in order to reach capacity, a codebook whose elements are vectors with normally distributed components must be used. The probability of error vanishes when the length of the codewords goes to infinity. In practical systems several limitations arise:

- 1. Since the delay and decoding time grow with the length of codewords, there is a limit on the allowed length of codewords.
- 2. Due to physical limitations of the electronic circuits used for signal processing, elements of a codeword must be bounded in absolute value. This is incompatible with using a codebook made of vectors with normally distributed elements, since the normal distribution does not have a bounded support.
- 3. For each value of the SNR, a different codebook must be used. In practical settings, only a finite number of codebooks are available.
- 4. A fraction of the available bandwidth is required for exchanging signaling information.

From link-level simulations for various SNR values, a function  $\phi$  mapping SNR into data rate can be obtained. We call  $\phi$  a *link-level curve*. Results in [47] show a very good fit between simulations and the so-called *modified* Shannon formula:

$$\phi(S) = bW \log_2\left(1 + \frac{S}{a}\right) \tag{2.63}$$

with  $a \ge 1$ , and  $b \le 1$  two constants. For a = 1 and b = 1 (2.63) is simply the Shannon formula. a represents the loss of efficiency due to practical coding schemes, and b the proportion of effectively usable bandwidth. [47] suggested that a = 1.25 and b = 0.75 were the correct values for LTE systems.

#### Block fading AWGN

The wireless channel is characterized by a phenomenon known as *channel* fading: the average received power from a transmitter varies on the timescale of milliseconds, due to the combined effect of multi-path propagation and mobility of the receiver. The most common model for this phenomenon is called *Rayleigh fading*.

**Definition 2.16.** The Rayleigh-fading AWGN channel is a channel with input-output relationship:

$$y[t] = hx[t] + n[t], t \in \mathbb{N}$$
 (2.64)

where h is a circular symmetric complex normal random variable, i.e  $h \equiv C\mathcal{N}(0,1)$ .

Such a model can be justified from the central limit theorem: if the channel output is the sum of a large number of copies of the input with an independent phase shift uniformly distributed in  $[0, 2\pi]$ , then the output is indeed a normally distributed complex random variable. Hence Rayleigh fading models a propagation environment with a large number of scatterers.

The common assumption is that channel fading occurs on a much slower time scale than the Gaussian noise, so that channel fading remains constant during the duration of a codeword. Such an assumption is called *block fading*. Unless explicitly stated, we always work with the block fading assumption. If block-fading applies and the value of h is known to the transmitter, then for each time t, the channel can be seen as an AWGN channel with SNR:

$$SNR = \frac{|h|^2 P}{WN_0}.$$
(2.65)

If the channel fading process is ergodic, the average throughput over a large number of codewords is then:

$$C = WE\left[\log_2(1 + \frac{|h|^2 P}{N_0 W})\right] = W \int_{\mathbb{R}^+} \log_2\left(1 + \frac{Px}{WN_0}\right) e^{-x} dx.$$
(2.66)

The value of h can be known to the transmitter as long as the time-scale on which the fading varies is slow enough to allow the transmitter to transmit a *training sequence* (whose value is known to both the transmitter and the receiver) to the receiver, and the receiver to feed back the received value to the transmitter.

#### 2.2. INFORMATION THEORY

#### Multi-tap Rayleigh-fading AWGN

When considering communication over a wide-band channel, the channel becomes frequency selective due to multi-path propagation and several copies of the transmitted message are received with delays.

**Definition 2.17.** The multi-tap Rayleigh-fading AWGN channel is a channel with input-output relationship:

$$y[t] = (h * x)[t] + n[t], \qquad (2.67)$$

$$h[t] = \sum_{l=1}^{L} \delta[t - \tau_l] h_l; \qquad (2.68)$$

with \* denoting convolution,  $\delta$  - Kronecker's delta,  $\tau_l \geq 0$  - the delay of the *l*th tap and  $\{h_l\}_l$  - *L* independent circular symmetric complex normal random variables, i.e  $h_l \equiv C\mathcal{N}(0, \overline{h}_l^2)$  and  $\sum_{l=1}^L \overline{h}_l^2 = 1$ .

In tables 2.1, 2.2 and 2.3 we describe three widely used models known as Pedestrian A, Pedestrian B and Vehicular A respectively. Those models were defined in [3], and are based on International Telecommunication Union (ITU) models [33].

Pedestrian A 3km/h		
Relative delay (ns)	Relative mean power (dB)	
0	0	
110	-9.7	
190	-19.2	
410	-22.8	

Table 2.1: Pedestrian A 3km/h

Communication over wide-band channels can be achieved with two techniques:

– Orthogonal Frequency-Division Multiple Access (OFDMA): the bandwidth is divided in a large number of narrow-band channels called *subcarriers*. Using a *cyclic prefix* technique, sub-carriers form parallel channels and there is no interference between sub-carriers. Each subcarrier can be described by a single tap (narrow-band) Rayleigh-fading AWGN channel. The capacity of such a scheme is equal to the capacity of a single-tap Rayleigh fading AWGN channel multiplied by the number of sub-carriers.

Pedestrian B 3km/h	
Relative delay (ns)	Relative mean power (dB)
0	0
200	-0.9
800	-4.9
1200	-8
2300	-7.8
3700	-23.9

Table 2.2: Pedestrian B 3km/h

Vehicular A 30km/h	
Relative delay (ns)	Relative mean power (dB)
0	0
310	-1.0
710	-9.0
1090	-10.0
1730	-15.0
2510	-20.0

Table 2.3: Vehicular A 30km/h

- Code Division Multiple Access (CDMA): N orthogonal signals  $\{s_n[t]\}_{1 \le n \le N}$  called *spreading codes* are defined, and the channel becomes equivalent to N parallel channels, namely:

$$y[t] = \left(\sum_{n=1}^{N} (h * s_n)[t] x_n[t]\right) + n[t], \qquad (2.69)$$

with  $x_n$  the transmitted signal on the *n*-th spreading code.

In CDMA,  $x_n$  changes on a much slower time scale than  $s_n$ . The time scale of  $s_n$  is called the *chip time*, and the time scale of  $x_n$  is called the *symbol time*. This allows to ignore inter-symbol interference. The capacity in the CDMA case is more complex than the OFDMA case. Namely, the multi-tap channel destroys the orthogonality of the spreading codes and  $\{h * s_n\}_{1 \le n \le N}$  is not an orthogonal set anymore. This phenomenon is known as *inter-code interference*: the signals transmitted on different codes interfere with each other.

#### 2.2. INFORMATION THEORY

#### **RAKE** receiver

Consider a user receiving data on the first code, all the other codes being used by other users. The *RAKE receiver* is a maximal ratio combiner: it projects the received signal on the first code convoluted with the channel impulse response  $h * s_1$ , treating the signal transmitted on the other codes as noise. We assume that the codes are normalized,  $||s_n||_{\mathcal{L}^2} = 1$ . The RAKE output is:

$$o[t] = S[t] + I[t] + W[t], (2.70)$$

$$S[t] = x_1[t] \|h * s_1\|_{\mathcal{L}^2}^2,$$
(2.71)

$$I[t] = \sum_{n=2}^{N} x_n[t] \langle h * s_1 , h * s_n \rangle_{\mathcal{L}^2}, \qquad (2.72)$$

$$W[t] = \langle h * s_1, n \rangle_{\mathcal{L}^2}.$$
(2.73)

The terms S, I and W represent the useful signal, inter-code interference and Gaussian noise respectively. We treat I as Gaussian noise so that the channel becomes equivalent to an AWGN channel. Define the correlation of the codes:

$$R_{n,n'}(t) = \langle s_n[.], s_{n'}[.-t] \rangle_{\mathcal{L}^2}.$$
 (2.74)

In current CDMA systems, the codes used are the convolution of a code identifying the serving cell called *scrambling code*, and a *Walsh-Hadamard code*. Detailed calculation of R in this case is done in [18]. Given R, we can calculate  $E[I^2]$  and the corresponding Signal to Interference plus Noise Ratio (SINR).  $E[I^2]$  depends on h, and we use the approximation to replace  $E[I^2]$  by its average value on the distribution of h. If power is split equally among the codes, then the interference power becomes proportional to the received power times (N - 1). The SINR has the following form:

$$SINR = \frac{P \sum_{l=1}^{L} |h_l|^2}{\beta (N-1)P + W N_0}.$$
 (2.75)

 $\beta \in [0, 1]$  is known as the *orthogonality factor* and depends on the codes correlation R as well as the taps delays and power. For instance, when there is only one tap, the codes remain orthogonal,  $\beta = 0$  and equation (2.75) becomes (2.65).

### MIMO Rayleigh-fading AWGN

The last channel model considered in this thesis is MIMO Rayleigh-fading AWGN, and is appropriate to model a multiplicity of transmit and receive

antennas communicating over a narrow-band channel. There are  $n_t$  transmit antennas and  $n_r$  receive antennas, and the propagation between each pair of transmitters is affected by a large number of scatterers, so that Rayleigh fading applies.

**Definition 2.18.** The MIMO Rayleigh-fading AWGN channel is a channel with input-output relationship:

$$Y[t] = HX[t] + N[t], (2.76)$$

with N[t] a vector of size  $n_r$  whose elements are independent white Gaussian noise processes and H - a  $n_t \times n_r$  matrix whose entries are circular symmetric complex normal random variables.

We assume that all entries of H are independent, which is the optimal case as far as capacity is concerned ([58]). We consider the Vertical Bell Labs Space-Time (V-BLAST) architecture in which the transmitter does not know the instantaneous channel realization H, but knows its statistics.  $n_t$ independent streams are transmitted in directions of the canonical vectors, and each stream is allocated equal power  $\frac{P}{n_t}$ . We write  $I_{n_r}$  - the  $n_r \times n_r$ identity matrix. The channel capacity is obtained by calculating the mutual information between two Gaussian vectors, and is (see [59](p337)):

$$C = \log_2 \left[ \det \left( I_{n_r} + \frac{P}{WN_0 n_t} H H^H \right) \right].$$
(2.77)

If the value of H is known, further gains can be obtained by transmitting independent data streams in the directions of the eigenvectors of  $H^H H$ , and allocating power to those directions using water-filling.

The distribution of the capacity given the distribution of H is not available in closed form. However, [35, 32] show that when the number of antennas increase, the capacity converges in distribution to a Gaussian variable with known mean and variance available in closed form, as a function of the SNR. The intuitive justification is that the computation of det induces a significant amount of averaging, so that a form of the central limit theorem can be obtained. This is indeed correct, and can be proven using random matrix theory, which analyses the distribution of eigenvalues of large random matrices. The formulas for mean and variance are stated in 3.3.3.

# 2.3 Stochastic approximation

In this section we give a basic exposition to *stochastic approximation*, which allows to analyze the behavior of discrete time iterative algorithms in the presence of noise by studying the behavior of an associated Ordinary Differential Equation (ODE). The ODE represents the mean dynamics of the algorithm and is generally much simpler to analyze than the discrete time algorithms since it is fully deterministic. Stochastic approximation was introduced in [51] to find the root of a function whose value is known through noisy measurements. [36] introduced a class of algorithms called stochastic gradient algorithms which optimize a cost function which is known through noisy measurements. The gradient is either known, or is estimated using *finite differences*. Stochastic approximation forms the basis for most forms of learning, including reinforcement learning as shown later. The ODE approach was introduced in [42] and is a popular approach to stochastic approximation. A complete exposition of stochastic approximation methods is found in [38, 16]

# 2.3.1 Definitions

We define  $\Theta = \mathbb{R}^N$  the parameter space equipped with the Euclidian norm,  $f: \Theta \to \Theta$  - a Lipschitz continuous vector field,  $H = \prod_{n=1}^{N} [a_n, b_n]$  with  $-\infty < a_n < b_n < +\infty$  an hyper-rectangle,  $[.]_H^+$  - the projection on H,  $\{\theta_n\}_{n\in\mathbb{N}}$  - a sequence of parameters,  $\{\epsilon_n\}_{n\in\mathbb{N}}$  - a sequence of positive step sizes and  $\{Y_n\}_{n\in\mathbb{N}}$  - a sequence of update vectors. The update vectors can be decomposed in three terms:

$$Y_n = f(\theta_n) + M_n + \beta_n, \qquad (2.78)$$

where  $\{M_n\}_{n\in\mathbb{N}}$ ,  $\{\beta_n\}_{n\in\mathbb{N}}$  are two (random) sequences in  $\Theta$ , which we call noise and bias respectively. We define the *filtration*  $\mathcal{F}_n$  by:

$$\mathcal{F}_n = \sigma\left(\{\theta_0, M_{n'}, \beta_{n'}, n' < n\}\right). \tag{2.79}$$

The iterate  $\theta_n$  is defined recursively as:

$$\theta_{n+1} = \left[\theta_n + \epsilon_n Y_n\right]_H^+. \tag{2.80}$$

To simplify the exposition of we restrict ourselves to the case where the parameters remain constrained to a compact set. If it not the case, we need to prove *stability*, i.e that the sequence  $\{\theta_n\}_{n\in\mathbb{N}}$  is either bounded a.s, or that it is tight. The reader can refer to the references given for more details.

# 2.3.2 The ODE approach

# Heuristic justification

Let us give an intuitive justification of the ODE approach. The iterate  $\theta_n$  is updated by taking a step in the direction of  $Y_n$ , and projecting the result

on the constraint set H. The "average direction" of update is  $f(\theta_n)$ . If step sizes  $\epsilon_n$  and the bias terms  $\beta_n$  are small enough, it is reasonable to think that the effect of the noise  $\{M_n\}_{n\in\mathbb{N}}$  will disappear because of averaging, and the behavior of the sequence  $\{\theta_n\}_{n\in\mathbb{N}}$  will be close to:

$$\theta_{n+1} = [\theta_n + \epsilon_n f(\theta_n)]_H^+ \,. \tag{2.81}$$

We recognize (2.81) as a Euler scheme for the ODE:

$$\dot{\theta} = f(\theta) + G(\theta), \qquad (2.82)$$

where G is the "minimal force" so that the solutions of the ODE remain inside H. If  $\theta$  is in the interior of H, we define  $G(\theta) = 0$ , and otherwise we define G so that  $f(\theta) + G(\theta)$  is the projection of  $f(\theta)$  on the tangent cone to H at point  $\theta$ .

Our reasoning is only heuristic, and stochastic approximation theorems provide us with a rigorous analysis of the link between the discrete algorithm and the ODE. In particular, it can be shown that under reasonable assumptions on the noise, the sequence  $\{\theta_n\}_{n\in\mathbb{N}}$  converges to asymptotically stable sets of the ODE. Depending on the assumptions on the step size sequence, almost sure convergence or convergence in distribution occurs.

#### Asymptotic behavior of ODEs

We define the mean ODE as:

$$\dot{\theta} = f(\theta) + G(\theta). \tag{2.83}$$

Since we are mainly concerned with the asymptotic behavior of (2.83), we briefly recall the concept of *Lyapunov stability*. This enables to define asymptotic behavior of the solutions of ODEs.

**Definition 2.19.** A stationary point of (2.83) is a point  $\theta^* \in \Theta$  such that:

$$f(\theta^*) + G(\theta^*) = 0.$$
 (2.84)

We define  $B(\theta, \delta)$  the open ball of radius  $\delta \ge 0$  centered at  $\theta \in \Theta$ . We denote by  $d(\theta, M) = \inf_{\theta_m \in M} \|\theta - \theta_m\|$  the distance to a closed set M.

Definition 2.20. A set M is said to be invariant if

$$\theta(0) \in M \implies \theta(t) \in M, t \ge 0.$$
(2.85)

#### 2.3. STOCHASTIC APPROXIMATION

**Definition 2.21.** A compact invariant set  $M \subset \Theta$  is an attractor if it has an open and invariant neighborhood O such that:

$$\theta(0) \in O \implies d(\theta(t), M) \xrightarrow[t \to +\infty]{} 0,$$
 (2.86)

O is called the basin of attraction of M.

**Definition 2.22.** A compact set M is said to be Lyapunov stable if for all  $\delta > 0$  there exists  $\delta' > 0$  such that if

$$d(\theta(0), M) < \delta', \tag{2.87}$$

then:

$$d(\theta(t), M) < \delta, t \ge 0. \tag{2.88}$$

Namely, a set is Lyapunov stable if all solutions remain close to it forever if they started close enough.

**Definition 2.23.** A compact set M is asymptotically stable if it is Lyapunov stable and an attractor. If the basin of attraction of M is  $\Theta$ , then M is globally asymptotically stable.

It is noted that there exists attractors which are not Lyapunov stable, so requiring Lyapunov stability in the definition of asymptotic stability is not redundant. A method for proving asymptotic stability is to define a *Lyapunov function*. A Lyapunov function extends the concept of potential energy of a physical system to a general dynamical system.

**Definition 2.24.** A continuously differentiable function  $V : \Theta \to R$  is a Lyapunov function if it is positive:

$$V(\theta) \ge 0, \ \theta \in \Theta, \tag{2.89}$$

radially unbounded:

$$V(\theta) \xrightarrow[\|\theta\| \to +\infty]{} +\infty \tag{2.90}$$

and strictly decreasing along trajectories:

$$V(\theta) > 0 \implies \dot{V}(\theta) < 0. \tag{2.91}$$

The existence of a Lyapunov function is enough to prove asymptotic stability.

**Theorem 2.6.** If V is a Lyapunov function, then the set:

$$V^{-1}(\{0\}) = \{\theta \in \Theta : V(\theta) = 0\}$$
(2.92)

is globally asymptotically stable.

# 2.3.3 Martingale difference noise: decreasing step sizes

We can state the theorems that link the asymptotic behavior of the discrete algorithms to the asymptotically stable sets of the ODE. We first examine the case in which the step sizes vanish at sufficient speed so that a.s convergence occurs.

Assumption 2.1.

$$\sum_{n \ge 0} \epsilon_n = +\infty \ , \ \sum_{n \ge 0} \epsilon_n^2 < +\infty$$
(2.93)

$$\sup_{n\geq 0} E\left[\left\|Y_n\right\|^2\right] < +\infty,\tag{2.94}$$

$$\sum_{n\geq 0} \epsilon_n \|\beta_n\| < +\infty \ a.s. \tag{2.95}$$

$$E\left[M_n|\mathcal{F}_n\right] = 0, \tag{2.96}$$

**Theorem 2.7.** If there exists a Lyapunov function V for the ODE (2.83), then:

$$d(\theta_n, V^{-1}(\{0\})) \xrightarrow[n \to +\infty]{a.s} 0, \qquad (2.97)$$

Application of Theorem 2.7 is straightforward in the case where  $f(\theta) = -\nabla_{\theta} g(\theta)$ , i.e the ODE is describes a *gradient descent* projected on *H*. In this case the sequence converges to the set of local minima of *g* in *H* a.s.

One must be careful when linking the behavior of the ODE and the discrete algorithm. In particular, it could be tempting to say that if all solutions of the ODE converge to a given point, then the sequence  $\{\theta_n\}$  converges to this point a.s. This turns out to be false, and we require global asymptotic stability for the result of Theorem 2.7 to hold.

# 2.3.4 Martingale difference noise: constant step sizes

While choosing decreasing step sizes as above guarantees strong (almost sure) convergence, it can lead to poor numerical behavior when the algorithm "gets stuck" in a given region for a long time, and choosing a small constant step size  $\epsilon_n = \epsilon > 0$  allows to overcome this problem. Furthermore, in many practical applications, the system on which the algorithm is applied is not stationary and varies slowly, so that a constant step size allows to "track" those slow variations while a decreasing step size would not allow adaptation.

The drawback is that if the noise does not vanish asymptotically, a.s convergence cannot occur, and only a weaker, distributional form of convergence can be expected. Namely, when  $\epsilon$  is small, it is reasonable to think that the

distribution of  $\theta_n$ ,  $n\epsilon >> 1$  will be concentrated around a globally asymptotically stable set (if such a set exists) of the ODE, and the amount of time spent next to this set can be rendered arbitrarily large by choosing  $\epsilon$  small enough.

In order to avoid ambiguity, we use the superscript  $\epsilon$  to explicitly highlight the dependence on  $\epsilon$ .

#### Assumption 2.2.

$$\epsilon_n = \epsilon > 0, n \ge 0 \tag{2.98}$$

 $\{Y_n^{\epsilon}\}$  is uniformly integrable.

$$\sup_{n\geq 0} E\left[\left\|Y_n^{\epsilon}\right\|^2\right] < +\infty,\tag{2.99}$$

$$E\left[\left\|\beta_{n}^{\epsilon}\right\|\right] \underset{n \to +\infty}{\to} 0.$$
(2.100)

$$E\left[M_{n}^{\epsilon}|\mathcal{F}_{n}\right] = 0, \qquad (2.101)$$

**Theorem 2.8.** If there exists a Lyapunov function V for the ODE 2.83, then, for all  $\mu > 0$ :

$$\limsup_{n} \mathbf{P}\left[d(\theta_n^{\epsilon}, V^{-1}(\{0\})) > \mu\right] = o(\epsilon).$$
(2.102)

Once again it is noticed that global asymptotic stability is required for the convergence to hold.

# 2.3.5 Correlated noise: decreasing step sizes

The theorems above assumed that the noise  $M_n$  is a martingale difference noise, which enables applying the Burkholder-Davis-Gundy inequality to bound the error due to the noise. It is noted that noise is uncorrelated in this case:  $E[M_nM_{n'}] = 0$ ,  $n \neq n'$ . However  $M_n$  and  $M_{n'}$  need not independent.

In many practical situations, the noise is not a martingale difference noise, and it exhibits correlation. For instance consider the case where we estimate the load of a queue by measuring the workload arriving per unit of time during successive time intervals. If the arrivals do not follow a Poisson process, instants of arrivals of jobs are correlated, and the load estimates are correlated as well. Another example of interest is packet scheduling where the *coherence time* of the fading process is large, so that the instantaneous throughput of users is correlated across several scheduling instants.

When the noise is correlated, we need to add a condition on the mixing time of the noise. If the noise mixes sufficiently fast, then the noise effect shall still be averaged out, and the system can be analyzed using the ODE as in the martingale difference noise case.

We write  $\xi_n \in \Xi$  the variables representing the effective memory of the noise process, with  $\Xi$  a metric space, such that the iterate can be decomposed as:

$$Y_n = f(\theta_n, \xi_n) + M_n + \beta_n.$$
(2.103)

The following assumptions are used:

### Assumption 2.3.

$$\sup_{n} E\left[\left\|Y_{n}\right\|^{2}\right] < +\infty, \tag{2.104}$$

$$\theta \to f(\theta, \xi) \text{ is continuous }, \ \xi \in \Xi,$$
 (2.105)

$$\epsilon_n = \frac{1}{n^{\gamma}} , \ \frac{1}{2} < \gamma \le 1.$$
(2.106)

There exists a function  $\overline{f}$  such that for all  $\theta$ :

$$\epsilon_N \sum_{n=1}^N (f(\theta, \xi_n) - \overline{f}(\theta)) \xrightarrow[N \to +\infty]{a.s} 0, \qquad (2.107)$$

$$\epsilon_N \sum_{n=1}^N M_n \xrightarrow[N \to +\infty]{a.s} 0, \qquad (2.108)$$

$$\epsilon_N \sum_{n=1}^N \beta_n \xrightarrow[N \to +\infty]{a.s} 0, \qquad (2.109)$$

$$(\theta,\xi) \to f(\theta,\xi)$$
 is bounded, (2.110)

$$\theta \to f(\theta, \xi)$$
 is continuous uniformly in  $\xi$ , (2.111)

Those assumptions are slightly less general than the conditions given in [38][Chapter 8], and are used to facilitate the exposition. Theorem 2.9 is a consequence of [38][Theorem 1.1, Chapter 6, page 166].

**Theorem 2.9.** If there exists a Lyapunov function V for the ODE

$$\dot{\theta} = \overline{f}(\theta) + G(\theta), \qquad (2.112)$$

then:

$$d(\theta_n, V^{-1}(\{0\})) \xrightarrow[n \to +\infty]{a.s} 0, \qquad (2.113)$$

# 2.3.6 Correlated noise: constant step sizes

As in the martingale difference noise case, assumptions are weaker for the constant step size. For the assumptions, A denotes an arbitrary compact subset of  $\Xi$ .

#### Assumption 2.4.

$$\epsilon_n = \epsilon > 0, n \ge 0 \tag{2.114}$$

$$\{Y_n^\epsilon\}_{n,\epsilon}$$
 is uniformly integrable, (2.115)

$$E\left[M_{n}^{\epsilon}|\mathcal{F}_{n}\right] = 0, \qquad (2.116)$$

$$\theta \to f(\theta, \xi) \text{ is continuous }, \xi \in A,$$
 (2.117)

$$\{\xi_n^\epsilon\}_{n,\epsilon}$$
 is tight, (2.118)

$$\{f(\theta,\xi_n^{\epsilon})\}_{n,\epsilon}, \{f(\theta_n^{\epsilon},\xi_n^{\epsilon})\}_{n,\epsilon}$$
 are uniformly integrable. (2.119)

$$\lim_{N,M,\epsilon} \frac{1}{M} \sum_{n=N}^{N+M-1} \beta_n^{\epsilon} = 0 \text{ in mean}, \qquad (2.120)$$

There exists a function  $\overline{f}$  such that for all  $\theta$ :

$$\lim_{N,M,\epsilon} \frac{1}{M} \sum_{n=N}^{N+M-1} (f(\theta,\xi_n^{\epsilon}) - \overline{f}(\theta)) \mathbf{1}_A(\xi_n^{\epsilon}) = 0 \text{ in probability.}$$
(2.121)

Theorem 2.10 is a consequence of [38] [Theorem 2.2, Chapter 8, page 255].

**Theorem 2.10.** If there exists a Lyapunov function V for the ODE (2.112), then, for all  $\mu > 0$ :

$$\limsup_{n} \mathbf{P}\left[d(\theta_n^{\epsilon}, V^{-1}(\{0\})) > \mu\right] = o(\epsilon).$$
(2.122)

# 2.4 Reinforcement learning

The principle of *reinforcement learning* is to find the optimal controller for a system which is both dynamical and random, without knowledge of its dynamics, based on trial-and-error. Its origins are in robotics and artificial intelligence and it has been applied to various areas such as computer backgammon, derivatives pricing in mathematical finance, and control of communication networks. Since reinforcement learning has mainly been developed considering that the system to control is a discrete time Markov Decision Process (MDP), we first give a short introduction to MDPs.

# 2.4.1 Markov decision processes

# Definition

MDPs model situations in which an agent controls a system whose evolution has the *Markov property*. The agent takes decisions sequentially based on the current state of the system: time is discrete, and at time  $t \in \mathbb{N}$  the agent observes the current state of the system s(t), chooses an action a(t), and receives a reward r(t). The system then moves to the next state s(t+1), and the chosen actions have an effect on the dynamics of the system. The goal of the agent is to maximize the cumulated rewards collected through time, for instance a discounted sum of rewards.

MDPs form the basis of reinforcement learning and have applications in various fields such as optimal stopping problems in mathematical finance, control of queuing networks in telecommunications, or shortest path problems. For a full exposition of MDPs and their optimal control, the reader can refer to [50].

We first define the probability space of a MDP. We consider a finite state space S, a finite action space A and a maximal reward value  $r_{max} < +\infty$ . The sample space  $\Omega$  is defined as:

$$\Omega = \{ \mathcal{S} \times \mathcal{A} \times [0, r_{max}] \}^{\infty}.$$
(2.123)

For S and A we use the discrete  $\sigma$ -algebra, and for  $[0, r_{max}]$  we use the Borel  $\sigma$ -algebra. We define  $\mathcal{F}$  the  $\sigma$ -algebra on  $\Omega$  as the product  $\sigma$ -algebra. A sample path of the MDP is:

$$\omega = (s(t), a(t), r(t))_{t \in \mathbb{N}}, \qquad (2.124)$$

with  $s(t) \in S$  the state at time  $t, a(t) \in A$  - the action chosen at time t and  $r(t) \in [0, r_{max}]$  - the reward at time t. We need to specify how actions are chosen given the observed sequence of states.

**Definition 2.25.** A Markov policy  $\pi$  is a mapping

$$\pi: \mathcal{S} \to \mathcal{D}(\mathcal{A}), \tag{2.125}$$

with  $\mathcal{D}(\mathcal{A})$  the set of probability distributions on  $\mathcal{A}$ . We write  $\pi(s, a)$  the probability of choosing action a in state s for policy  $\pi$ .

A Markov policy is a decision rule based on the current state, at time t, it specifies the distribution of a(t) as a function of s(t). We can consider more general *history dependent policies* for which the decision is a function of the complete history of the process up to time t, but it turns out (see for instance

#### 2.4. REINFORCEMENT LEARNING

[12]) that for every MDP there exists Markov policies which are optimal. We can restrict ourselves to Markov policies without loss of generality. We use the term "policy" in place of "Markov policy" in the rest of this chapter.

When applying policy  $\pi$ , the probability space of the MDP is  $(\Omega, \mathcal{F}, \mathbf{P}^{\pi})$ .

**Definition 2.26.** The process  $t \to (s(t), a(t), r(t))$  is a MDP with policy  $\pi$  if it verifies the Markov property. Namely, for all  $(s_t, a_t, r_t)_{t \in \mathbb{N}} \in \Omega$  and  $T \in \mathbb{N}$ :

$$\mathbf{P}^{\pi}[s(t) = s_t, a(t) = a_t, r(t) \in [0, r_t], 0 \le t \le T]$$
  
=  $\mathbf{P}[s(0) = s_0] \prod_{t=0}^{T} \mathbf{P}[s(t+1) = s_{t+1} | s(t) = s_t, a(t) = a_t]$   
 $\pi(s_t, a_t) \mathbf{P}[r(t) \in [0, r_t] | s(t) = s_t, a(t) = a_t].$  (2.126)

Equation (2.126) states that:

- the distribution of the action a(t) depends only on the current state s(t),
- the distribution of the reward r(t) depends only on the current action a(t) and state s(t),
- the transition from s(t) to s(t+1) depends only on the current action a(t).

**Definition 2.27.** A MDP is time-invariant if the transition probabilities and conditional rewards do not depend on time:

$$t \mapsto \mathbf{P} [r(t) = r_t | s(t) = s_t, a(t) = a_t] \text{ and} t \mapsto \mathbf{P} [s(t+1) = s_{t+1} | s(t) = s_t, a(t) = a_t],$$
(2.127)

are both constant.

We work with time-invariant MDPs in the rest of this chapter. To ease notation, we define the *transition probabilities*:

$$p(s', s, a) = \mathbf{P}\left[s(t+1) = s' | s(t) = s, a(t) = a\right], \qquad (2.128)$$

and the average rewards:

$$\overline{r}(s,a) = E[r(t)|s(t) = s, a(t) = a].$$
 (2.129)

The goal of the agent is to maximize the expectation of a function of the rewards. We call this quantity the *total reward* (as opposed to the reward

obtained at a given time t). We denote by  $R(\omega)$  the total reward obtained for sample path  $\omega$ , and several definitions are possible for  $R(\omega)$ :

Average reward, horizon T:  $R(\omega) = \frac{1}{T} \sum_{t=0}^{T} r(t)$ 

Average reward, infinite horizon :

$$R(\omega) = \liminf_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T} r(t) \quad (2.131)$$

(2.130)

Discounted reward, discount factor  $\lambda$ :  $R(\omega) = (1 - \lambda) \sum_{t>0} \lambda^t r(t)$ . (2.132)

When considering the average reward with infinite horizon, using a lim inf is necessary since the limit does not exist in general. The expected total reward of the MDP when using policy  $\pi$  is  $E^{\pi}[R(\omega)]$ .

### **Optimal control**

The optimal control of a MDP consists in finding the policy which maximizes the expected total reward, and can be done by defining the *value function*  $V: S \to \mathbb{R}^+$ :

$$V(s) = \max E^{\pi} [R(\omega)|s(0) = s], \qquad (2.133)$$

which is the best expected total reward that can be obtained starting at state s. The maximum is taken on all policies.

We consider the case where the total reward is the *discounted reward* with discount factor  $\lambda$  as in (2.132). The value function obeys a dynamic programming principle called the *Bellman equation*. The Bellman equation is:

$$V(s) = \max_{a \in \mathcal{A}} \left( \overline{r}(s, a) + \lambda \sum_{s' \in \mathcal{S}} p(s', s, a) V(s') \right), \qquad (2.134)$$

which we write in short form:

$$V = B(V). \tag{2.135}$$

The Bellman equation (2.134) is obtained by writing the discounted reward as the sum of the reward obtained at time 0 and the discounted reward obtained after time 0.

Given two functions  $V : \mathcal{S} \to \mathbb{R}^+, V' : \mathcal{S} \to \mathbb{R}^+, (2.134)$  gives that:

$$\max_{s \in \mathcal{S}} |B(V)(s) - B(V')(s)| \le \lambda \max_{s \in \mathcal{S}} |V(s) - V'(s)|.$$

$$(2.136)$$

Hence B is a contraction mapping with Lipschitz constant  $\lambda < 1$ , and the Bellman equation has a unique solution by the contraction mapping theorem.

#### 2.4. REINFORCEMENT LEARNING

Furthermore, V can be derived iteratively by applying B repeatedly. This procedure is called *value iteration*:

$$V^{(0)} = 0,$$
  
 $V^{(n+1)} = B(V^{(n)}), n \ge 0$  (2.137)

and  $V^{(n)} \xrightarrow[n \to +\infty]{} V$  geometrically at rate  $\lambda$ , using the contraction mapping theorem.

We define the Q function:

$$Q(s,a) = \overline{r}(s,a) + \lambda \sum_{s' \in \mathcal{S}} p(s',s,a) V(s'), \qquad (2.138)$$

which is the expected total reward if the agent starts at state s, uses action a at time 0, and then chooses the optimal policy afterwards. The Bellman equation (2.134) is the optimality equation for the MDP, namely a policy  $\pi^*$  is optimal if and only if:

$$V(s) = \sum_{a \in \mathcal{A}} \pi^*(s, a) Q(s, a).$$
(2.139)

Hence once the value function has been determined, an optimal policy can be derived by applying (2.139).

# 2.4.2 Q-learning

#### **Reinforcement learning**

In principle, an optimal policy can always be found. However, in order to use value iteration, we need to know the expectation of the rewards  $\overline{r}(s, a)$  and the transition probabilities p(s', s, a). Namely we need to know the *model* of the system we are controlling. In many situations of practical interest, the system model is either unknown, or too complicated to obtain with sufficient accuracy. The principle of reinforcement learning is to derive the optimal policy without knowledge about the system model, through repeated interaction with the system. Since the system can only be observed during a finite time, we call sample path of length T,  $(s(t), a(t), r(t))_{0 \le t \le T}$ , from which the value function and the optimal policy can be estimated. The definition of the probability space poses no difficulty when we restrict the observation to  $0 \le t \le T$ . Reinforcement learning is a *model-free* method. A complete exposition of reinforcement learning is found in [56].

### **Q**-learning

The simplest reinforcement learning algorithm is *Q*-learning introduced in [61]. The principle of Q-learning is to construct a sequence  $\{\tilde{Q}^{\epsilon}(t,.,.)\}_{t\in\mathbb{N}}$  of estimates of the Q function. The algorithm can be decomposed into three steps. The initialization is:

$$\tilde{Q}^{\epsilon}(0,s,a) = 0 \quad , \quad (s,a) \in \mathcal{S} \times \mathcal{A}.$$
(2.140)

The action selection rule is:

$$a(t) = \begin{cases} \arg\max_{a \in \mathcal{A}} \tilde{Q}^{\epsilon}(t, s(t), a) & \text{with probability } (1 - p_{exp}) \\ Uniform(\mathcal{A}) & \text{with probability } p_{exp} \end{cases}$$
(2.141)

with  $p_{exp} \in (0, 1)$  an exploration probability. The estimate of the Q function is updated by:

$$\tilde{Q}^{\epsilon}(t+1,s,a) = \tilde{Q}^{\epsilon}(t,s,a) , \quad (s,a) \neq (s(t),a(t)), \quad (2.142)$$
$$\tilde{Q}^{\epsilon}(t+1,s(t),a(t)) = (1-\epsilon)\tilde{Q}^{\epsilon}(t,s(t),a(t))$$

$$+ \epsilon \left( r(t) + \lambda \max_{a \in \mathcal{A}} \tilde{Q}^{\epsilon}(t, s(t+1), a) \right).$$
(2.143)

with  $\epsilon > 0$  a constant step size. The action selection rule is to choose the action which has yielded the best performance so far with probability  $(1 - p_{exp})$ (exploitation), and to choose a random action with probability  $p_{exp}$  (exploration). Exploration is necessary since all state-action pairs must be visited infinitely often for convergence to occur. The update step is a stochastic approximation scheme. It can be shown, using stochastic approximation, that the sequence of estimates  $\tilde{Q}^{\epsilon}(t_{\epsilon},.,.) \stackrel{d}{\underset{\epsilon \to 0}{\leftarrow} 0} Q$  with  $\epsilon t_{\epsilon} \stackrel{\to}{\underset{\epsilon \to 0}{\leftarrow} 0} +\infty$ .

# 2.4.3 Policy search approach

# Policy search

The advantages of Q-learning are its simplicity in terms of implementation, and its convergence to the true Q function, from which the optimal policy is easily obtained. However, one serious drawback is that it is only suitable for a small number of states, otherwise it converges very slowly. This is because Q-learning is not able to obtain information about states-action pairs that have not been visited before. In other words it is not able to *generalize* its knowledge about a state-action pair to other state-action pairs which are close. This is problematic because the number of states grows exponentially

### 2.4. REINFORCEMENT LEARNING

with the dimension of the state space, and problems of practical interest such as backgammon can easily have millions of states. A solution to maintain *scalability* is to employ a policy search technique. The space of all policies is much too large to be explored entirely, so the search is restricted to a "wellchosen" subset  $\{\pi(\theta) : \theta \in \Theta\}$  where  $\Theta$  is a convex subset of some Euclidean space.

 $\pi(\theta)$  is the policy associated to parameter  $\theta$ , and we define  $J(\theta)$  the expected total reward obtained when policy  $\pi(\theta)$  is applied. The optimal control problem is reduced to optimizing J with respect to  $\theta$ . If the parameterized policies are well chosen,  $J(\theta)$  is differentiable with respect to the policy parameter  $\theta$ , and a local optimum can be found using a local search such as gradient ascent. We mainly consider local search because the gradient of J with respect to  $\theta$  can be estimated from observing sample paths, which enables to optimize  $\theta$  in an *on-line* fashion, using measurements from a real system and optimizing  $\theta$  while the system is running.

### Finite horizon policy gradient

We assume that the total reward depends only on sample paths of length T, for instance choosing the total reward as the average reward with horizon T as in (2.130). The simplest approach to obtain the gradient  $\nabla_{\theta} J(\theta)$  would be to observe several sample paths of length T, and use *finite differences* to estimate  $\nabla_{\theta} J(\theta)$ . Although simple to implement, this approach has two drawbacks. The first is that obtaining several sample paths of the same system can only be done if the system can be simulated, and does not apply when we are using measurements from an up and running system. The second is that the number of sample paths required is equal to the number of components of  $\theta$ , so the method only applies for a small number of components.

A better alternative would be to estimate all components of  $\nabla_{\theta} J(\theta)$  from the same sample path. This is done using a technique known as the *likelihood ratio* technique. Alternatively, it is denoted *REINFORCE* ([62]) in the reinforcement learning literature.

We write the definition of J:

$$J(\theta) = E^{\pi(\theta)} \left[ R(\omega) | s(0) = s_0 \right] = \int_{\Omega} R(\omega) \mathbf{P}^{\pi(\theta)}(\omega) d\omega$$
 (2.144)

where  $\mathbf{P}^{\pi(\theta)}(\omega)$  is the probability of sample path  $\omega$  when starting at s(0) and applying policy  $\pi(\theta)$ . We use the notation  $\pi(\theta)(s, a)$  to denote the probability of selecting action a in state s according to policy  $\pi(\theta)$ . The probability of a sample path can be decomposed using the Markov property as in (2.126):

$$\mathbf{P}^{\pi(\theta)}(s(t) = s_t, a(t) = a_t, r(t) \in [0, r_t])$$
  
=  $\mathbf{P}[s(0) = s_0] \left(\prod_{t=0}^T p(s_{t+1}, s_t, a_t) \mathbf{P}[r(t) \in [0, r_t] | s(t) = s_t, a(t) = a_t]\right)$   
 $\left(\prod_{t=0}^T \pi(\theta)(s_t, a_t)\right).$  (2.145)

We have decomposed the sample path probability as two terms: a term containing transitions and rewards which *do not* depend on  $\theta$ , and a term containing selected actions which does depend on  $\theta$ . We differentiate the average cost (2.144) as:

$$\nabla_{\theta} J(\theta) = \int_{\Omega} R(\omega) \nabla_{\theta} \mathbf{P}^{\pi(\theta)}(\omega) d\omega$$
  
= 
$$\int_{\Omega} R(\omega) \frac{\nabla_{\theta} \mathbf{P}^{\pi(\theta)}(\omega)}{\mathbf{P}^{\pi(\theta)}(\omega)} \mathbf{P}^{\pi(\theta)}(\omega) d\omega$$
  
= 
$$E^{\pi(\theta)} \left[ R(\omega) \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)) \right]. \qquad (2.146)$$

Equation (2.146) indeed enables to estimate all components of  $\nabla_{\theta} J(\theta)$ from the same sample paths, as long as the term  $\nabla_{\theta} \log(\mathbf{P}_{\theta}(\omega))$  can be computed for any sample path. This is indeed the case, using equation (2.145):

$$\nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)) = \sum_{t=0}^{T} \nabla_{\theta} \log(\pi(\theta)(s_t, a_t)), \qquad (2.147)$$

where we have used the fact that only the terms linked to action selection depend on  $\theta$ .

For the reasoning above to make sense, we shall assume that:

#### Assumption 2.5.

$$0 < \pi(\theta)(s, a), s \in \mathcal{S}, a \in \mathcal{A}$$
(2.148)

and

#### Assumption 2.6.

$$\theta \to \pi(\theta)(s, a) \text{ is differentiable }, s \in \mathcal{S}, a \in \mathcal{A}.$$
 (2.149)

Finally, assume that we simulate N independent sample paths  $(\omega_n)_{1 \le n \le N}$ , then an unbiased estimate for  $\nabla_{\theta} J(\theta)$  is:

$$\frac{1}{N}\sum_{n=1}^{N} R(\omega_n) \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega_n)) \xrightarrow[N \to +\infty]{a.s} \nabla_{\theta} J(\theta), \qquad (2.150)$$

by the law of large numbers.

In summary, we are able to estimate all the components of  $\nabla_{\theta} J(\theta)$  from the same sample paths, and without any knowledge about the transition probabilities and rewards distribution, since  $\nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega))$  does not involve transition probabilities or rewards distribution.

Going back to assumptions 2.5 and 2.6, we can conclude that: in a given state, policy  $\pi(\theta)$  must assign a strictly positive probability to all actions and therefore cannot be deterministic, furthermore, the probability assigned to a given action in a given state must be differentiable. The learning can only be done with *stochastic policies*, although we know that in general the optimal policy is deterministic. This is a fundamental aspect of reinforcement learning: all actions in all states must be tried with strictly positive probability to find the optimal policy. This was already the case for Q-learning: the action selection policy (2.141) chooses a random action with probability  $p_{exp} > 0$  so that all actions in all states are selected with strictly positive probability.

# Variance reduction techniques

To enhance the efficiency of policy gradient methods, several *variance reduction* techniques have been developed. Since the gradient estimates are to be used in a *stochastic gradient* algorithm to optimize the total reward, reducing the variance of the gradient estimates enables faster convergence.

A first technique is to introduce a constant denoted *baseline* in the reinforcement learning literature. Replacing R by a constant function in (2.146), we have that:

$$E^{\pi(\theta)} \left[ \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)) \right] = \nabla_{\theta} \mathbf{1} = 0.$$
 (2.151)

Hence, for  $b \in \mathbb{R}$  we have that:

$$\nabla_{\theta} J(\theta) = E^{\pi(\theta)} \left[ R(\omega) \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)) \right]$$
  
=  $E^{\pi(\theta)} \left[ (R(\omega) - b) \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)) \right].$  (2.152)

The optimal baseline  $b^*$  minimizes the variance of the gradient estimator.  $b^*$  is found by differentiating the variance of the gradient estimator with respect to b:

$$b^* = \frac{E^{\pi(\theta)} \left[ R(\omega) (\nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)))^2 \right]}{E^{\pi(\theta)} \left[ (\nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega)))^2 \right]}.$$
 (2.153)

We simulate N independent sample paths  $(\omega_n)_{1 \leq n \leq N}$ , then the estimate with baseline for  $\nabla_{\theta} J(\theta)$  is:

$$b_N = \frac{\sum_{n=1}^N R(\omega_n) (\nabla_\theta \log(\mathbf{P}^{\pi(\theta)}(\omega_n)))^2}{\sum_{n=1}^N (\nabla_\theta \log(\mathbf{P}^{\pi(\theta)}(\omega_n)))^2} \xrightarrow[N \to +\infty]{a.s} b^*, \qquad (2.154)$$

$$\frac{1}{N}\sum_{n=1}^{N} (R(\omega_n) - b_N) \nabla_{\theta} \log(\mathbf{P}^{\pi(\theta)}(\omega_n)) \xrightarrow[N \to +\infty]{a.s} \nabla_{\theta} J(\theta).$$
(2.155)

In the case where the total reward is written as a sum of rewards, another variance reduction technique can be used. We expose the method for the average reward with horizon T. Given a sample path  $\omega$  and  $0 \leq t \leq T$ , intuition is that the value of r(t) should not depend on the actions taken after t. Combining (2.146) and (2.147) this suggests that:

$$\nabla_{\theta} J(\theta) = E^{\pi(\theta)} \left[ \frac{1}{T} \sum_{t=0}^{T} r(t) \sum_{u=0}^{t} \nabla_{\theta} \log(\pi(\theta)(s_u, a_u)) \right].$$
(2.156)

This intuition turns out to be correct, and a proof can be found in [62].

### Infinite horizon policy gradient

When the horizon is infinite, the previous approach cannot be applied directly. We work with the *infinite horizon* average cost. A method to estimate the gradient for infinite horizons is given in [10]. In this case we need a supplementary assumption on the ergodicity of the underlying Markov chain, for all values of  $\theta$ .

**Assumption 2.7.** For all  $\theta$ ,  $\{s(t)\}_{t\in\mathbb{N}}$  is an ergodic Markov chain.

This in particular ensures that:

$$J(\theta) = \lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T} r(t), \qquad (2.157)$$

where the limit exists because we have assumed ergodicity. As expected, if the initial state s(0) is not drawn according to the stationary distribution associated to policy  $P(\theta)$ , then the gradient  $\nabla_{\theta} J(\theta)$  cannot be estimated without bias. The bias term is linked to the mixing time of the Markov chain, so that for rapidly mixing systems, the bias is negligible.

To estimate the gradient we define the eligibility trace z(t):

$$z(0) = 0 z(t+1) = \beta z(t) + \nabla_{\theta} \log(\pi(\theta)(s(t), a(t))),$$
(2.158)

68

where  $\beta \in (0, 1)$  is a parameter controlling the time window used by the algorithm for averaging. The link between  $\beta$  and the mixing time is clarified later. The gradient is estimated recursively by  $\Delta(t)$ :

$$\Delta(0) = 0,$$
  

$$\Delta(t+1) = \frac{t\Delta(t) + z(t)r(t)}{t+1}.$$
(2.159)

When t is large, the dot product between  $E[\Delta(t)]$  and  $\nabla_{\theta} J(\theta)$  becomes strictly positive so that  $E[\Delta(t)]$  is a valid ascent direction.

**Theorem 2.11** ([10]). There exists  $\beta_0 \in [0, 1)$ , such that if  $\beta \geq \beta_0$ , then:

$$\liminf_{t \to +\infty} \langle E\left[\Delta(t)\right], \nabla_{\theta} J(\theta) \rangle > 0, \qquad (2.160)$$

and:

$$\lim_{t \to +\infty} \mathbf{P}\left[\left\langle \Delta(t) , \nabla_{\theta} J(\theta) \right\rangle \le 0\right] = 0.$$
(2.161)

Results in [10] show that when the mixing time is large,  $\beta_0$  must be close to 1, and that the variance of the gradient estimates grows as  $\frac{1}{(1-\beta)^2}$ . Namely there is a bias-variance trade-off: for the gradient estimates to be accurate we need to choose  $\beta$  close to 1, but  $\beta$  too close to 1 makes the variance of the gradient estimates too large, so that the resulting stochastic gradient ascent converges very slowly. Similarly to the finite horizon case, a baseline can be added to reduce the variance of the gradient estimate  $\Delta(t)$ .

# 2.4.4 Continuous time models

The reinforcement learning techniques exposed above are valid in problems in which time is discrete. However, a lot of problems of interest are naturally expressed in continuous time, and we would like to be able to work in continuous time, without approximating the continuous system using a discretization. For instance, problems of control of queuing networks are naturally described in continuous time. It turns out that by sampling the continuous time system at well chosen *random times*, we can use the existing reinforcement learning algorithms for discrete time systems directly. We give a brief summary of this technique known as *uniformization*. Complete exposition is found in [50]. The continuous time equivalent is called a Semi-Markov Decision Process (SMDP).

# 2.4.5 Semi-Markov decision processes

Informally, a SMDP is the same as a MDP, except for the fact that the system stays in each state a random amount of time. The sample space becomes:

$$\Omega = \{ \mathcal{S} \times \mathcal{A} \times [0, r_{max}] \times \mathbb{R}^+ \}^{\infty}.$$
(2.162)

We use the Borel  $\sigma$ -algebra on  $\mathbb{R}^+$ , and the  $\sigma$ -algebra for the sample space is the product  $\sigma$ -algebra. A sample path of the SMDP is:

$$\omega = (s(t), a(t), r(t), T(t))_{t \in \mathbb{N}}, \qquad (2.163)$$

where T(t) denotes the amount of time that the system stays in state s(t). Namely, at the *t*-th decision period, the system arrives in state s(t), the agent chooses an action a(t), then the system stays in state s(t) during a random duration T(t), and the agent receives a reward r(t)T(t). We define the time of arrival in state s(t) by:

$$U(t) = \sum_{u=0}^{t-1} T(u).$$
 (2.164)

It is noted that the SMDP model allows the agent to take decisions only *upon arrival in a given state*. The same definition of policies is used as in the MDP case.

When applying policy  $\pi$ , the probability space of the SMDP is  $(\Omega, \mathcal{F}, \mathbf{P}^{\pi})$ .

**Definition 2.28.** The process  $t \to (s(t), a(t), r(t), T(t))$  is a SMDP with policy  $\pi$  if it verifies the Markov property. Consider a sample path

 $(s_t, a_t, r_t, T_t)_{t \in \mathbb{N}} \in \Omega$  and write  $(s_t, a_t, r_t, T_t)_{0 \leq t \in \mathbb{N}}$  its restriction to  $0 \leq t \leq T' < +\infty$ . Namely we only consider the T' first state transitions. The Markov property holds if, for all  $T' \in \mathbb{N}$ :

$$\mathbf{P}^{\pi}[s(t) = s_t, a(t) = a_t, r(t) \in [0, r_t], T(t) \in [0, T_t], 0 \le t \le T']$$
  
=  $\mathbf{P}[s(0) = s_0] \prod_{t=0}^{T'} p(s_{t+1}, s_t, a_t) \pi(s_t, a_t) \mathbf{P}[r(t) \in [0, r_t] | s(t) = s_t, a(t) = a_t]$   
=  $\mathbf{P}[T(t) \in [0, T_t] | s(t) = s_t, a(t) = a_t].$  (2.165)

The Markov property (2.165) states that the distribution of amount of time spent in a given state is only a function of the action upon arrival in this state.

An interesting particular case is when the time spent in each state is exponentially distributed, which makes the system a Continuous Time Markov Decision Process (CTMDP). This model is useful for instance in the the control of queuing networks when users arrive according to a Poisson process with exponentially distributed service times. **Definition 2.29.** A SMDP is a CTMDP if the time spent in each state is exponentially distributed:

$$\mathbf{P}[T(t) \in [0, T_t]] = 1 - \exp(-T_t\beta(s_t, a_t)), \qquad (2.166)$$

where  $\beta(s, a) > 0$  is the inverse of the average time spent in state s when action a was chosen.

The total reward for a sample path are defined in the same way as for a MDP.

Average reward, horizon 
$$T$$
:  
 $R(\omega) = \frac{\sum_{t=0}^{T} T(t)r(t)}{U(T+1)}$  (2.167)  
Average reward, infinite horizon :  
 $R(\omega) = \liminf_{T \to +\infty} \frac{\sum_{t=0}^{T} T(t)r(t)}{U(T+1)}$   
(2.168)  
Discounted reward, discount factor  $\lambda$ :  
 $R(\omega) = (1-\lambda) \sum_{t\geq 0} r(t) \int_{U(t)}^{U(t+1)} \lambda^{u} du$   
(2.169)

It is noted that a Discrete Time Markov Decision Process (DTMDP) is a particular case of a SMDP, where T(t) = 1 a.s.,  $t \in \mathbb{N}$ .

# Uniformization

We show how an equivalent DTMDP can be associated to a CTMDP, and we call this DTMDP the *uniformization* of the CTMDP. In order to avoid any ambiguity, we use the superscript <sup>c</sup> to denote the CTMDP, and <sup>d</sup> to denote the uniformization. For the CTMDP, let  $p^c(s', s, a)$ ,  $\overline{r}^c(s, a)$  and  $\beta^c(s, a)$  denote the transition probabilities, average reward and inverse of time spent in a state respectively. For the CTMDP to be equivalent to a DTMDP, we need to assume that the inverse of the average time spent in any state is bounded:

### Assumption 2.8.

$$\beta_{\infty} = \max_{s \in \mathcal{S}, a \in \mathcal{A}} \beta(s, a) < +\infty.$$
(2.170)

Physically,  $\beta_{\infty}$  represents the *fastest speed* at which a state can be exited. Uniformization is based on sampling the CTMDP at exponentially distributed times with parameter  $\beta_{\infty}$ . We define the uniformized DTMDP with transition probabilities:

$$p^{d}(s', s, a) = \frac{p^{c}(s', s, a)\beta(s, a)}{\beta_{\infty}},$$
(2.171)

$$p^{d}(s, s, a) = 1 - \frac{\beta(s, a)}{\beta_{\infty}},$$
 (2.172)

and rewards:

l rewards: Average reward : Discounted reward, discount factor  $\lambda$   $r^{d}(s, a) = \frac{r^{c}(s, a)}{\beta_{\infty}},$   $r^{d}(s, a) = \frac{r^{c}(s, a)(\lambda + \beta(s, a))}{\lambda + \beta_{\infty}}.$ (2.174)

Optimal control of the CTMDP can be achieved by finding the optimal control for the uniformized DTMDP.

**Theorem 2.12** ([50]). An optimal policy for the uniformized DTMDP is also optimal for the CTMDP.
## Chapter 3

# Packet scheduling

In modern radio access networks, channel quality information of active users is available at the BS on the time scale of milliseconds, so that the users can be selected dynamically for transmitting based on their channel quality. *Channel-aware scheduling* provides appreciable gains in terms of user throughputs since users only transmit when their channel quality is good. In this chapter we investigate how packet scheduling can be used as a SON functionality. We are concerned with three main questions:

- the convergence of scheduling algorithms to steady-state throughputs which maximize a utility function ( $\alpha$ -fair utility in our case)
- the analytical evaluation of scheduling gains for different channel models
- dynamic adaptation of the scheduling strategy to perform coveragecapacity optimization.

This chapter is based on our contributions [22, 23, 24, 28]. Channel-aware scheduling has received a large amount of attention in the literature. Analytical performance evaluation for Proportional Fair (PF) scheduling over Rayleigh-fading channels were given in [11] and [13]. The convergence of PF scheduling was proven in [39]. [14] and [17] showed the impact on the flow-level performance of the network serving elastic traffic through a queuing analysis. [57] considered channel-aware scheduling without the full buffer assumption: the backlog of active users evolves dynamically. The goal is to ensure queue stability whenever it is possible. It was proven that a strategy called max-weight scheduling ensures stability whenever possible. Maxweight scheduling takes into account both channel quality and backlog to choose which user should transmit.

## 3.1 Channel-aware scheduling

#### 3.1.1 The model

We consider N users communicating with a BS sharing the same radio resources in downlink. We adopt a *full buffer* traffic model: each user always has an infinite amount of data to transmit. The radio resources are divided into M resource units. Time is slotted, and at each time slot for each resource the BS scheduler picks a user for transmission based on their instantaneous channel conditions. A scheduling policy P is defined by the choice of a user for every scheduling instant for every resource  $(P_t^m)_{t \in \mathbb{N}, 1 \le m \le M}$ . Namely  $P_t^m = i$ means that user i is selected for transmission at the t-th time slot on the m-th resource. We define  $r_{i,t}^m$  as the instantaneous throughput of user i for the t-th time slot on resource m. We write  $r_t = (r_{i,t}^m)_{1 \le i \le N, 1 \le m \le M}$ .

We assume *perfect channel knowledge*: at the *t*-th time slot, the scheduler knows  $r_t$  and can make use of this information to choose the scheduled user.

Let  $\epsilon > 0$  denote a small averaging parameter, and define  $\overline{r}_{i,t}$  the average throughput of user *i* at time *t* by the following recursive equation:

$$\overline{r}_{i,t+1} = (1-\epsilon)\overline{r}_{i,t} + \epsilon \sum_{m=1}^{M} \delta_{P_{t+1}^m,i} r_{i,t+1}^m, \qquad (3.1)$$

where  $\delta$  denotes Kronecker's delta.

This definition for the mean allocated throughput is more relevant to reflect the QoS perceived by a user than using an arithmetic mean (which would be replacing  $\epsilon$  in (3.1) by  $\frac{1}{t}$ ) because it induces a *decay* of past observed values.  $\epsilon$  is the parameter which controls the size of the averaging window, and is related to the service we are considering. Namely, for applications such as FTP, the average data rate allocated to a user during the file transfer time, typically a few seconds, is a relevant performance indicator. For applications such as voice however, the perceived quality is related to the average data rate on a much smaller time scale, e.g 100ms, because of the play-out buffer size. Hence the value of  $\epsilon$  for FTP traffic shall be smaller than for voice traffic.

We assume that  $\overline{r}_{i,t} = 0$ , and equation (3.1) can also be written:

$$\overline{r}_{i,t} = \epsilon \sum_{t'=0}^{t} (1-\epsilon)^{t-t'} \left( \sum_{m=1}^{M} \delta_{P_{t'}^m, i} r_{i,t'}^m \right).$$
(3.2)

Assumption 3.1. (i)  $\{r_t\}_{t\in\mathbb{N}}$  is i.i.d,

(ii) There exists  $r_{max} < +\infty$  such that  $r_{i,t}^m \leq r_{max} \ \forall t, i, m$ , a.s

#### 3.1. CHANNEL-AWARE SCHEDULING

(iii)  $r_t$  has a density with respect to Lebesgue measure denoted  $p_t$ ,

The assumption that the instantaneous throughputs  $r_t$  are i.i.d is valid as long as we assume that the duration of a time slot is larger than the channel coherence time. Consider for instance Rayleigh fading, then as stated in [34], the autocorrelation of the channel fading for a single user between t and  $t + \tau$ is  $J_0(\omega_M \tau)$ , where  $J_0$  is the 0-th order Bessel function and  $\omega_M$ -the maximum Doppler shift.

We consider policies which only take into account the instantaneous throughput. Namely a scheduling policy is given by a function  $f : \mathbb{R}^{N \times M} \to \mathcal{S}^M$  with  $\mathcal{S}$  the unit simplex in  $\mathbb{R}^N$  and  $\mathbf{P}[P_t^m = i] = f_i^m(r_t)$ . With assumptions 3.1, applying scheduling policy defined by f, we have that the expected average throughput converges to:

$$E\left[\overline{r}_{i,t}\right] \xrightarrow[t \to +\infty]{} R_i(f),$$

$$R_i(f) = \int_{\mathbb{R}^{NM}} \left(\sum_{m=1}^M r_i^m f_i^m(r)\right) p(r) dr,$$
(3.3)

and that the variance of the average throughput vanishes when  $\epsilon \to 0^+$ :

$$\limsup_{t \to +\infty} E\left[ (\overline{r}_{i,t} - R_i(f))^2 \right] \underset{\epsilon \to 0^+}{\to} 0.$$
(3.4)

Hence the average throughput of user *i* converges in mean square to  $R_i(f)$  when  $\epsilon \to 0^+$ . We call R(f) the *achieved throughput* when applying scheduling policy defined by f.

**Definition 3.1.** The set of achievable throughputs  $\mathcal{R}$  is defined as the set of throughputs given by all policies:

$$\mathcal{R} = \{ r \in \mathbb{R}^N : r = R(f), f : \mathbb{R}^{N \times M} \mapsto \mathcal{S}^M \}.$$
(3.5)

**Proposition 3.1.** The achievable throughput set  $\mathcal{R}$  is a compact and convex subset of  $\mathbb{R}^N$ .

Proof. Consider the vector space of functions  $f : \mathbb{R}^{N \times M} \to \mathbb{R}^{N \times M}$  with norm  $\|f\|_{\infty} = \sup_{r \in \mathbb{R}^{N \times M} 1 \le i \le N 1 \le m \le M} \max_{f_i} |f_i^m(r)|$ . From its definition,  $f \mapsto R(f)$  is a linear application. R is continuous since  $\max_{1 \le i \le N} |R_i(f)| \le Mr_{max} \|f\|_{\infty}$ .  $\mathcal{R}$  is the image of the set of functions  $f : \mathbb{R}^{N \times M} \mapsto \mathcal{S}^M$  which is closed and convex by R. So  $\mathcal{R}$  is convex and closed.  $\mathcal{R}$  is also bounded by  $Mr_{max}$ , hence it is convex compact.

#### **3.1.2** $\alpha$ -fair scheduling

#### Definition

As introduced in [46], the  $\alpha$ -fair utility for allocation  $r \in \mathcal{R}$  and  $\alpha \in [0, +\infty)$  is:

$$U(r) = \begin{cases} \sum_{i=1}^{N} \log(d+r_i) &, \ \alpha = 1\\ \sum_{i=1}^{N} \frac{(r_i+d)^{1-\alpha}}{1-\alpha} &, \ \alpha \neq 1 \end{cases}$$
(3.6)

where d > 0 can be chosen as small as desired and is only present to avoid problematic behavior near 0.

**Definition 3.2.** The  $\alpha$ -fair allocation is the allocation which maximizes U on  $\mathcal{R}$ .

For  $\alpha > 0$ , U is a strictly concave function, and  $\mathcal{R}$  is a compact convex set, therefore the  $\alpha$ -fair allocation is unique. For  $\alpha = 0$ , there might exist several  $\alpha$ -fair allocations.

Intuitively, increasing  $\alpha$  shall result in fairer allocations, namely users with bad channel conditions get more resources, while decreasing  $\alpha$  shall result in increasing the sum throughput  $\sum_{i=1}^{N} r_i$ . However, the notion of measuring *fairness* is somehow unclear. [40] gives a formal justification to this and shows that the  $\alpha$ -fair allocation is in fact the allocation that maximizes a fairness measure while preserving Pareto optimality. Namely the  $\alpha$ -fair allocation maximizes the fairness measure:

$$\operatorname{sign}(1-\alpha) \left[ \sum_{i=1}^{N} \left( \frac{r_i}{\sum_{j=1}^{N} r_j} \right)^{1-\alpha} \right]^{\frac{1}{\alpha}}, \qquad (3.7)$$

while being Pareto optimal.

If the density p is known, then the  $\alpha$ -fair allocation can be derived using convex programming techniques. We assume that this distribution is unknown, and we use an algorithm similar to *stochastic gradient* to derive it.

#### Allocation rule

The proposed algorithm to derive the  $\alpha$ -fair allocation is to adopt the following allocation policy:

$$P_{t+1}^{m} = \arg\max_{1 \le i \le N} \frac{r_{i,t+1}^{m}}{(\overline{r}_{i,t} + d)^{\alpha}}.$$
(3.8)

#### 3.1. CHANNEL-AWARE SCHEDULING

We prove in Section 3.2 that this policy converges to the  $\alpha$ -fair allocation when  $\epsilon \to 0^+$ . We first give a heuristic justification for the scheduling rule. Let  $(\Delta U)_i^m$  denote the variation of utility if user *i* is chosen for transmitting at time t + 1 on resource *m*, which we approximate with a first-order Taylor expansion. If  $\alpha = 1$ , the increase in utility for user *i* is:

$$\log\left((1-\epsilon)\overline{r}_{i,t} + \epsilon r_{i,t+1}^m + d\right) - \log(\overline{r}_{i,t} + d) = \epsilon \frac{r_{i,t+1}^m - \overline{r}_{i,t}}{\overline{r}_{i,t} + d} + o(\epsilon).$$
(3.9)

The decrease for the other users is:

$$\log\left((1-\epsilon)\overline{r}_{i,t}+d\right) - \log(\overline{r}_{i,t}+d) = -\epsilon \frac{\overline{r}_{i,t}}{\overline{r}_{i,t}+d} + o(\epsilon).$$
(3.10)

We add (3.9) and (3.10):

$$(\Delta U)_i = \epsilon \left[ \frac{r_{i,t+1}^m}{\overline{r}_{i,t} + d} - \sum_{i'=1}^N \frac{\overline{r}_{i',t}}{\overline{r}_{i',t} + d} \right] + o(\epsilon).$$
(3.11)

If  $\alpha \neq 1$ :

$$\frac{1}{1-\alpha} \left[ \left( (1-\epsilon)\overline{r}_{i,t} + \epsilon r^m_{i,t+1} + d \right)^{1-\alpha} (\overline{r}_{i,t} + d)^{1-\alpha} \right] = \epsilon \frac{r^m_{i,t+1} - \overline{r}_{i,t}}{(\overline{r}_{i,t} + d)^\alpha} + o(\epsilon),$$
(3.12)

and:

$$\frac{1}{1-\alpha}\left[\left((1-\epsilon)\overline{r}_{i,t}+d\right)^{1-\alpha}-(\overline{r}_{i,t}+d)^{1-\alpha}\right] = -\epsilon\frac{\overline{r}_{i,t}}{(\overline{r}_{i,t}+d)^{\alpha}} + o(\epsilon).$$
 (3.13)

We add (3.12) and (3.13):

$$(\Delta U)_i = \epsilon \left[ \frac{r_{i,t+1}^m}{(\overline{r}_{i,t}+d)^\alpha} - \sum_{i'=1}^N \frac{\overline{r}_{i',t}}{(\overline{r}_{i',t}+d)^\alpha} \right] + o(\epsilon).$$
(3.14)

In both cases, for small  $\epsilon$ , the optimal choice is:

$$P_{t+1}^{m} = \arg \max_{1 \le i \le N} \frac{r_{i,t+1}^{m}}{(\overline{r}_{i,t} + d)^{\alpha}}.$$
(3.15)

 $\alpha=1$  corresponds to a PF scheduler, and  $\alpha=0$  to a Max Throughput (MTP) scheduler.

## **3.2** Convergence of $\alpha$ -fair schedulers

In this section we give a convergence analysis of  $\alpha$ -fair scheduling, using the ODE technique which has been used previously in [38] and [39] to show the convergence of the PF scheduler. We use the stochastic approximation results presented in section 2.3. We first work with  $\alpha > 0$  fixed, and the case  $\alpha = 0$  is studied separately.

The scheduling rule (3.8) is clearly a stochastic approximation scheme. We define the average drift:

$$h(\overline{r}) = \int_{\mathbb{R}^{NM}} \left( \sum_{m=1}^{M} r_i^m \mathbf{1}_{\substack{\{i = \arg\max_{i'} \frac{r_i^m}{(d+\overline{r}_{i'})^{\alpha}}\}}}(r) \right) p(r) dr, \qquad (3.16)$$

and:

$$g(\overline{r}) = h(\overline{r}) - \overline{r}.$$
(3.17)

The ODE is:

$$\dot{\overline{r}} = g(\overline{r}). \tag{3.18}$$

### 3.2.1 The mean ODE

**Proposition 3.2.** *h* is positive, bounded and Lipschitz continuous. Furthermore:

$$h(\overline{r}) = \overline{r} , \ h(\overline{r}') = \overline{r}' , \ \overline{r}' \ge \overline{r} \implies \overline{r}' = \overline{r}.$$
 (3.19)

*Proof.* h is positive and bounded by  $Mr_{max}$ .

We first assume that  $\|\overline{r}\| \leq 1$ , let  $P_{i,j,\overline{r},\overline{r}',m}$  be the following quantity:

$$P_{i,j,\overline{r},\overline{r}',m} = \mathbf{P}\left[\left\{\frac{r_i^m}{(d+\overline{r}_i)^{\alpha}} \ge \frac{r_j^m}{(d+\overline{r}_j)^{\alpha}}\right\} \cup \left\{\frac{r_i^m}{(d+\overline{r}_i')^{\alpha}} \le \frac{r_j^m}{(d+\overline{r}_j')^{\alpha}}\right\}\right],$$

which we can rewrite:

$$P_{i,j,\overline{r},\overline{r}',m} = \mathbf{P}\left[r_j^m \left(\frac{d+\overline{r}_i}{d+\overline{r}_j}\right)^{\alpha} \le r_i^m \le r_j^m \left(\frac{d+\overline{r}_i'}{d+\overline{r}_j'}\right)^{\alpha}\right].$$
(3.20)

Let  $F_{r_i^m}(x) = \mathbf{P}[r_i^m \le x],$ 

$$P_{i,j,\overline{r},\overline{r}',m} = E\left[F_{r_i^m}\left(r_j^m\left(\frac{d+\overline{r}_i'}{d+\overline{r}_j'}\right)^\alpha\right) - F_{r_i^m}\left(r_j^m\left(\frac{d+\overline{r}_i}{d+\overline{r}_j}\right)^\alpha\right)\right].$$
 (3.21)

We have assumed  $\|\overline{r}\| \leq 1$ , so there is a constant  $K_{\alpha}$  such that:

$$\left| \left( \frac{d + \overline{r}_i}{d + \overline{r}_j} \right)^{\alpha} - \left( \frac{d + \overline{r}'_i}{d + \overline{r}'_j} \right)^{\alpha} \right| \le K_{\alpha} \|\overline{r} - \overline{r}'\|.$$
(3.22)

 $x \mapsto F_{r_i^m}(x)$  is Lipschitz continuous since we have assumed  $r_i^m$  to have a density with respect to the Lebesgue measure, so for a certain constant  $K_F$ :

$$P_{i,j,\overline{r},\overline{r}',m} \le E\left[K_{\alpha}K_{F}\|\overline{r}-\overline{r}'\|r_{j}\right] \le K_{\alpha}K_{F}r_{max}\|\overline{r}-\overline{r}'\|.$$
(3.23)

We can bound the variation of h:

$$\|h(\overline{r}) - h(\overline{r}')\| \le 4r_{max} \sum_{i \ne j} \sum_{1 \le m \le M} P_{i,j,\overline{r},\overline{r}',m}.$$
(3.24)

We conclude that there exists a constant  $K_h$  so that:

$$\|h(\overline{r}) - h(\overline{r}')\| \le K_h \|\overline{r} - \overline{r}'\|.$$
(3.25)

We have proved that h is Lipschitz continuous for  $\|\overline{r}\| \leq 1$ . Let  $K_2 \geq 1$ , we have that :

$$h(\overline{r}) = h(\frac{\overline{r}+d}{K_2} - d). \tag{3.26}$$

We combine this with (3.25), with  $K_2$  large enough:

$$\|h(\overline{r}) - h(\overline{r}')\| = \|h(\frac{\overline{r} + d}{K_2} - d) - h(\frac{\overline{r}' + d}{K_2} - d)\|$$
$$\leq \frac{C_h}{K_2} \|\overline{r} - \overline{r}'\|$$
$$\leq C_h \|\overline{r} - \overline{r}'\|.$$

So we have proved that h is globally Lipschitz continuous.

The last proposition is true since all components of h cannot increase when all components of  $\overline{r}$  increase.

#### Existence of a solution to the ODE

We have to prove that the ODE has solutions on  $\mathbb{R}^+$ . We have that h is Lipschitz continuous so the *Picard-Lindelof theorem* assures us that it has a unique local solution. Furthermore, we know that there exists a unique maximal solution defined on some maximal interval  $[0, t_0[. h \text{ is bounded by} Mr_{max} \text{ so } \|\overline{r}(t)\| \leq \|\overline{r}(0)\| + tMr_{max}$ , therefore  $t_0 = +\infty$ , or else the solution is not maximal.

### 3.2.2 Convergence to a unique limit

#### Monotone dynamical systems

We first state some results from the theory of *monotone dynamical systems*, and the reader can refer to [55] for their proofs.

We denote by  $\Gamma_t(\overline{r}), \overline{r} \in (\mathbb{R}^+)^N$  the value at time t of the solution to the ODE starting at  $\overline{r}$ . We define the orbit of  $\overline{r}$  by  $O(\overline{r}) = {\Gamma_t(\overline{r}) | t \ge 0}$ and the limit set of  $\overline{r}$  by  $\omega(\overline{r}) = \bigcap_{t\ge 0} \bigcup_{s\ge t} \Gamma_s(\overline{r})$ .  $\overline{r}$  is called an equilibrium point if  $O(\overline{r}) = x$ , and we denote by E the set of equilibrium points.  $\overline{r}$  is called a quasi-convergent point if  $\omega(\overline{r}) \subset E$  and we denote by Q the set of quasi-convergent points. If  $\overline{r} \le \overline{r}' \Rightarrow \Gamma_t(\overline{r}) \le \Gamma_t(\overline{r}') \forall (\overline{r}, \overline{r}') \in (\mathbb{R}^+)^N \times (\mathbb{R}^+)^N$  $\forall t \in \mathbb{R}^+$ , then we say that  $\Gamma$  is monotone. We have the following theorems:

**Theorem 3.1.** If  $\Gamma$  is monotone and  $\overline{r} < \overline{r'}$  then either:

- (i)  $\omega(\overline{r}) < \omega(\overline{r'})$ , or
- (*ii*)  $\omega(\overline{r}) = \omega(\overline{r'}) \subset E$ .

**Theorem 3.2.** If  $\Gamma$  is monotone then Q is dense in  $(\mathbb{R}^+)^N$ .

We show that those results can be applied to the ODE we are considering, using the following comparison theorem:

**Theorem 3.3.** We consider the ODE  $\dot{\overline{r}} = g(\overline{r}')$ . Let  $g : (\mathbb{R}^+)^N \to \mathbb{R}^N$ , verifying:

- (i) g is continuous
- (ii) The solution to the ODE is unique for every initial condition
- (*iii*)  $\overline{r} \leq \overline{r}'$  and  $\overline{r}_i = \overline{r}'_i \Rightarrow g_i(\overline{r}) \leq g_i(\overline{r}')$
- (iv) For  $T \ge 0$ ,  $(\overline{r}, \delta) \in (\mathbb{R}^+)^N \times (\mathbb{R}^+)^N$ , we have that:

$$\sup_{0 \le t \le T} \left\| \Gamma_t(\overline{r}) - \Gamma_t(\overline{r} + \delta) \right\| \underset{\delta \to 0}{\to} 0.$$
(3.27)

Then  $\Gamma$  is monotone.

Condition (iii) is called the *Kamke condition*.

The ODE we are considering satisfies those conditions. (i) and (ii) have been proved previously. (iii) comes from the fact that  $\overline{r}_i \to \frac{1}{(d+\overline{r}_i)^{\alpha}}$  is decreasing. To prove (iv), let T > 0, since h is Lipschitz continuous we can apply *Gronwall's lemma*:

$$\|\Gamma_t(\overline{r}) - \Gamma_t(\overline{r} + \delta)\| \le \|\delta\| e^{K_3 t}, \qquad (3.28)$$

for a certain constant  $K_3$ . We then have that:

$$\sup_{0 \le t \le T} \|\Gamma_t(\overline{r}) - \Gamma_t(\overline{r} + \delta)\| \le \|\delta\| e^{K_3 T} \xrightarrow[\|\delta\| \to 0]{} 0.$$
(3.29)

So the conditions of the previous theorem are valid.

80

#### Convergence for $\overline{r}(0) = 0$

By noticing that g(0) > 0, the following theorem proves that the solution starting at 0 converges to a certain  $\overline{r}^*$ .

**Theorem 3.4.** If the ODE verifies the Kamke condition then any solution starting at  $\overline{r}$  with  $g(\overline{r}) > 0$  converges to an equilibrium point.

We show that all solutions converge to the same limit.

We have proved that  $\omega(0) = \{\overline{r}^*\}$ . Let  $\overline{r} > 0$  be an arbitrary initial condition, and  $\overline{r}' \geq \overline{r}$  with  $\overline{r}' \in Q$  since Q is dense in  $(\mathbb{R}^+)^N$ . We know that  $\omega(\overline{r}') \subset E$  since  $\overline{r}' \in Q$ , let us assume that  $\omega(0) < \omega(\overline{r}')$ . Let  $\overline{r}'' \in \omega(\overline{r}')$ , we have that  $h(\overline{r}'') = 0$  and  $\overline{r}'' > \overline{r}^*$ , which contradicts (3.19). So  $\omega(\overline{r}') = \omega(0) = \{\overline{r}^*\}$ , and finally  $\omega(\overline{r}) = \{\overline{r}^*\} \ \forall \overline{r} \leq 0$ , in other words all solutions converge to  $\overline{r}^*$ .

#### Optimality

Finally, we prove that the scheduling strategy is optimal, namely that any other scheduling strategy achieves lower utility. We differentiate the utility function:

$$\dot{U}\left(\overline{r}(t)\right) = \sum_{i=1}^{N} \frac{h_i(\overline{r}(t)) - \overline{r}_i(t)}{(d + \overline{r}_i(t))^{\alpha}}.$$
(3.30)

We prove that  $\theta^*$  is a local maximum of U on  $\mathcal{R}$ . Let  $f : (\mathbb{R}^+)^{NM} \to \mathcal{S}^M$  an arbitrary allocation rule, from the definition of h we have that:

$$\sum_{i=1}^{N} \frac{R_i(f)}{(d+\overline{r}_i(t))^{\alpha}} \le \sum_{i=1}^{N} \frac{h_i(\overline{r}(t))}{(d+\overline{r}_i(t))^{\alpha}}.$$
(3.31)

Let  $\overline{r}_f(t)$  and  $\overline{r}(t)$  the trajectories implied by the new and the usual (from equation (3.15)) allocation rules respectively, both starting at  $\overline{r}^*$ . By combining (3.30) and (3.31) at t = 0 we have that:

$$\dot{U}\left(\overline{r}_f(t)\right)|_{t=0} \le \dot{U}\left(\overline{r}(t)\right)|_{t=0} \le 0.$$
(3.32)

Therefore  $\overline{r}^*$  is a local maximum of U on  $\mathcal{R}$ . Since U is strictly concave on  $\mathcal{R}$ , we have proved that the scheduling rule achieves optimal utility.

#### Application to the $\alpha$ -fair scheduler, $\alpha = 0$

The case  $\alpha = 0$  is a bit different since U is linear, and not strictly concave. However the proof is simpler since the scheduling strategy does not depend on the mean throughput. The ODE is:

$$\dot{\overline{r}} = g(\overline{r}) = h(0) - \overline{r}, \qquad (3.33)$$

and the solution is  $\overline{r}(t) = e^{-t}\overline{r}(0) + (1 - e^{-t})h(0)$ , which converges to a unique limit h(0). It shall be noted that the limit is unique because  $\mathbf{P}\left[r_i^m = r_j^m, i \neq j\right] = 0$ . If  $\mathbf{P}\left[r_i^m = r_j^m, i \neq j\right] > 0$  it might not be the case, for example consider the case where all the  $r_i$  are constant and equal to 1, any point in the simplex is a limit throughput. Since we have assumed independence of the channel between two scheduling instants and that U is linear, the policy that chooses the user with the best channel also maximizes U over  $\mathcal{R}$ .

## 3.3 Calculation of scheduling gain

We are interested in calculating the gain obtained by channel-aware scheduling. The scheduling gain is the ratio between the throughput allocated to a user and the throughput allocated to this user by a Round Robin (RR) scheduler. The RR scheduler is a non-opportunistic scheduler which allocates each resource an equal amount of time to each user. To avoid confusion, we denote by  $\overline{r}^{RR}$  the RR throughput and  $\overline{r}^{\alpha}$  the  $\alpha$ -fair throughput. The RR throughput is:

$$\overline{r}_i^{RR} = \frac{1}{N} E\left[\sum_{m=1}^M r_i^m\right].$$
(3.34)

The scheduling gain is evaluated for the channel models introduced in subsection 2.2.4. For each channel model, we state the distribution of the instantaneous throughput. We work with a general link curve  $\phi$  to make our calculations applicable to practical systems (except in the MIMO case). If we are only interested in Shannon capacity,  $\phi$  can be taken according to the Shannon formula. We denote by  $L_{\phi}$  the Laplace transform of  $\phi$  and it plays an important role for our calculations. For all scheduling gain calculations, we add two assumptions.

Assumption 3.2. (i)  $\{r_i^m\}_{1 \le m \le M} \perp \{r_{i'}^m\}_{1 \le m \le M}$  if  $i \ne i'$ .

(*ii*) 
$$r_i^m \stackrel{d}{=} r_i^{m'}$$

For a general  $\alpha$ , the scheduling gain cannot be evaluated in closed form. We calculate the scheduling gain for three particular cases:  $\alpha = 0$  (MTP),  $\alpha = 1$  (PF) and  $\alpha \to +\infty$  (Max-Min Fair (MMF)).

## 3.3.1 Rayleigh-fading AWGN

For the Rayleigh-fading AWGN channel, the instantaneous throughput on a resource can be described by:

$$r_i^m = \phi(S_i Z_i^m), \tag{3.35}$$

with  $Z_i^m$  an exponentially distributed random variable with mean 1,  $S_i$  - the mean SINR of user *i* on a resource. From assumptions 3.2, we have that  $\{Z_i^m\}_{1 \le i \le N, 1 \le m \le M}$  are independent.

#### $\mathbf{R}\mathbf{R}$

The RR throughput is:

$$\overline{r}_i^{RR} = \frac{M}{NS_i} L_\phi \left(\frac{1}{S_i}\right). \tag{3.36}$$

## MTP ( $\alpha = 0$ )

For the MTP ( $\alpha = 0$ ), the probability of choosing user *i* for resource *m* is:

$$\mathbf{P}\left[S_{i}Z_{i}^{m} \ge S_{i'}Z_{i'}^{m}, \ i' \neq i\right].$$
(3.37)

The throughput is then:

$$\overline{r}_{i}^{0} = \frac{M}{S_{i}} \int_{0}^{+\infty} \phi(x) \prod_{i' \neq i} (1 - e^{-\frac{x}{S_{i'}}}) e^{-\frac{x}{S_{i}}} dx.$$
(3.38)

By developing the product, we obtain the following expression:

$$\overline{r}_{i}^{0} = \frac{M}{S_{i}} \sum_{i'=0}^{N-1} (-1)^{i'} \sum_{a_{1} < \dots < a_{i'}, a_{j} \neq i \; \forall j} L_{\phi} \left( \frac{1}{S_{i}} + \sum_{j=0}^{i'} \frac{1}{S_{a_{j}}} \right).$$
(3.39)

#### **PF** ( $\alpha = 1$ )

Results for the PF case ( $\alpha = 1$ ) were given in [13]:

$$\overline{r}_{i}^{1} = \frac{1}{S_{i}} \sum_{i'=0}^{N-1} \binom{N-1}{i'} (-1)^{i'} L_{\phi} \left(\frac{i'+1}{S_{i}}\right).$$
(3.40)

It is noted that formula (3.40) was derived based on the approximation that the scheduled user is the user who maximizes the ratio between his instantaneous SINR and mean SINR. Strictly speaking, the PF schedules the user who maximizes the ratio between instantaneous and mean throughput.

In particular, if  $\phi$  is a linear function we have the expression [11]:

$$\frac{\overline{r}_i^1}{\overline{r}_i^{RR}} = \frac{1}{S_i} \sum_{i'=0}^{N-1} \binom{N-1}{i'} (-1)^{i'} \frac{1}{S_i} \frac{S_i^2}{(i'+1)^2} = \sum_{i'=1}^N \frac{1}{i'}, \quad (3.41)$$

which is asymptotically equivalent to  $\log(N)$ .

#### MMF ( $\alpha \to +\infty$ )

The case  $\alpha \to +\infty$  is analytically tractable, and of particular interest for the dynamic alpha policy we address in the next section. Since for t > 1,  $\mathbf{P}\left[\overline{r}_{i,t} = \overline{r}_{i',t}, i \neq i'\right] = 0$  and  $\mathbf{P}\left[r_i^m = 0\right] = 0$ , the scheduling rule (3.8)) becomes:

$$P_{t+1}^m = \arg\min_{1 \le i \le N} (\overline{r}_{i,t}). \tag{3.42}$$

All users have the same throughput for a MMF scheduler. We know that  $\overline{r}_i^{\infty}$  exists. Assume that  $\overline{r}_i^{\infty} > \overline{r}_j^{\infty} \ge 0$ . So there exists T so that user i never transmits after T, and  $\overline{r}_i^{\infty} = 0$ , a contradiction. Therefore,  $\overline{r}_i^{\infty} = \overline{r}_{i'}^{\infty}$ ,  $\forall i, i'$ .

We know that when user i is alone in the cell, it's throughput is:

$$M \int_0^{+\infty} \phi(x) \frac{e^{-\frac{x}{S_i}}}{S_i} dx = \frac{M}{S_i} L_\Phi\left(\frac{1}{S_i}\right), \qquad (3.43)$$

and since the scheduling rule (3.42) does not depend on the instantaneous throughput, we have that  $\overline{r}_i^{\infty} = p_i \frac{M}{S_i} L_{\Phi}(\frac{1}{S_i})$ , with  $\sum_{i=1}^N p_i = 1$  and  $\overline{r}_i^{\infty} = \overline{r}_{i'}^{\infty}$ ,  $\forall i, i'$ . Hence:

$$\overline{r}_{i}^{\infty} = \left(\sum_{i'=1}^{N} \frac{S_{i'}}{ML_{\phi}(\frac{1}{S_{i'}})}\right)^{-1}.$$
(3.44)

This formula is useful because it enables us to determine analytically which users can be covered by adjusting the  $\alpha$ , and which users cannot be covered. Scheduling them would simply waste resources and they therefore should be ignored when deciding which  $\alpha$  to use.

#### 3.3.2 Multi-tap Rayleigh-fading AWGN

For the multi-tap Rayleigh-fading AWGN channel, we consider CDMA using a RAKE receiver as described in 2.2.4. Using formula (2.75), the instantaneous throughput on a resource can be described by:

$$r_i^m = \phi(S_i Z_i^m) \,, \tag{3.45}$$

$$Z_i^m = \sum_{l=1}^{L} Z_i^{m,l}.$$
 (3.46)

with  $\{Z_i^{m,l}\}_l L$  independent exponentially distributed random variables with means  $\{\overline{h}_l^2\}_l$ ,  $S_i$  - the mean SINR of user *i* on a resource. We recall that  $\overline{h}_l^2$  is the average relative power of the *l*-th channel tap, and  $\sum_{l=1}^L \overline{h}_l^2 = 1$ .  $S_i$  takes into account the inter-code interference through the orthogonality factor.

#### 3.3. CALCULATION OF SCHEDULING GAIN

We write  $p_Z$  the probability density function (p.d.f) of  $Z_i^m$ . Calculating the Laplace transform of  $p_Z$  and using a partial fraction expansion, [21] shows that:

$$p_Z(x) = \sum_{l=1}^{L} \frac{p_l}{\bar{h}_l^2} e^{-\frac{x}{\bar{h}_l^2}},$$
(3.47)

with:

$$p_l = \prod_{l' \neq l} \frac{\overline{h}_l^2}{\overline{h}_l^2 - \overline{h}_{l'}^2}.$$
(3.48)

We have to assume that  $\overline{h}_l^2 \neq \overline{h}_{l'}^2$ ,  $l \neq l'$ , otherwise the  $(p_l)_{1 \leq l \leq L}$  are not defined. Actually, even in the case where some eigenvalues are equal, [60] shows that it is possible to separate them artificially by a small value, with results close to the exact solution. Furthermore the cumulative distribution function (c.d.f) of the fast-fading is

$$F_Z(x) = \sum_{l=1}^{L} p_l (1 - e^{-\frac{x}{\overline{h}_l^2}}) = 1 - \sum_{l=1}^{L} p_l e^{-\frac{x}{\overline{h}_l^2}}.$$
 (3.49)

The calculation of scheduling gains is done similarly to the single-tap Rayleigh fading case. We do not develop the expression for the MTP case.

#### $\mathbf{R}\mathbf{R}$

The RR throughput is:

$$\overline{r}_i^{RR} = \frac{M}{NS_i} \sum_{l=1}^{L} \frac{p_l}{\overline{h}_l^2} L_\phi \left(\frac{1}{\overline{h}_l^2 S_i}\right).$$
(3.50)

**PF** (  $\alpha = 1$  )

In order to reduce notational complexity, we use the multi-index notation: given  $\kappa \in \mathbb{N}^L$ ,  $x \in \mathbb{R}^L$  and  $n \in \mathbb{N}$ , we write  $|\kappa| = \sum_{l=1}^L \kappa_l$ ,  $x^{\kappa} = \prod_{l=1}^L x_l^{\kappa_l}$ , and  $\binom{n}{\kappa} = \frac{n!}{\prod_{l=1}^L \kappa_l!}$ , and  $\langle \kappa, x \rangle = \sum_{l=1}^L x_l \kappa_l$ .

The probability of being chosen by the PF scheduler is the probability to have the best ratio between instantaneous SINR and mean SINR. The binomial formula gives:

$$[F_Z(x)]^{N-1} = \sum_{n=0}^{N-1} \binom{N-1}{n} (-1)^n [1 - F_Z(x)]^n.$$
(3.51)

We evaluate each term of the sum by the multinomial formula:

$$[1 - F_Z(x)]^n = [\sum_{l=1}^{L} p_l e^{-\frac{x}{\bar{h}_l^2}}]^n = \sum_{|\kappa|=n} \binom{n}{\kappa} p^{\kappa} e^{-x < \kappa, \frac{1}{\bar{h}^2} >}, \qquad (3.52)$$

where  $\frac{1}{\overline{h}^2} \in \mathbb{R}^L$  is the vector whose components are the  $\frac{1}{\overline{h}_l^2}$ ,  $1 \le l \le L$ . Summing the terms in (3.51) gives:

$$[F_Z(x)]^{N-1} = \sum_{0 \le |\kappa| \le N-1} \binom{N-1}{\kappa, N-1-|\kappa|} (-1)^{|\kappa|} p^{\kappa} e^{-x < \kappa, \frac{1}{h^2} >}, \qquad (3.53)$$

where  $\binom{N-1}{\kappa,N-1-|\kappa|} = \binom{N-1}{\kappa} \frac{1}{(N-1-|\kappa|)!}$  We multiply (3.53) by the p.d.f of the fast-fading:

$$[F_{Z}(x)]^{N-1} \sum_{l=1}^{L} \frac{p_{l}}{\overline{h}_{l}^{2}} e^{-\frac{x}{\overline{h}_{l}^{2}}} = \sum_{0 \le |\kappa| \le N-1, 1 \le l \le L} \binom{N-1}{\kappa, N-1-|\kappa|} (-1)^{|\kappa|} \frac{p_{l}p^{\kappa}}{\overline{h}_{l}^{2}} e^{-x(<\kappa, \frac{1}{\overline{h}^{2}} > +\frac{1}{\overline{h}_{l}^{2}})}.$$
 (3.54)

The scheduling throughput can then be evaluated by:

$$\overline{r}_{i,+\infty,1} = \int_{0}^{+\infty} \Phi(xS_{i}) [F_{Z}(x)]^{N-1} \sum_{l=1}^{L} \frac{p_{l}}{\overline{h}_{l}^{2}} e^{-\frac{x}{\overline{h}_{l}^{2}}} dx$$

$$= \sum_{0 \le |\kappa| \le N-1, 1 \le l \le L} \binom{N-1}{\kappa, N-1-|\kappa|} (-1)^{|\kappa|} \frac{p_{l}p^{\kappa}}{\overline{h}_{l}^{2}} L_{\Phi} \left(\frac{<\kappa, \frac{1}{\overline{h}^{2}} > +\frac{1}{\overline{h}_{l}^{2}}}{S_{i}}\right).$$
(3.55)

It is noted that (3.40) is a particular case of (3.55) for L = 1.

## MMF ( $\alpha \to +\infty$ )

Using the same argument as previously, the MMF throughput is:

$$\overline{r}_{i}^{\infty} = \left(\sum_{i'=1}^{N} \frac{S_{i'}}{M \sum_{l=1}^{L} \frac{p_{l}}{h_{l}^{2}} L_{\phi}\left(\frac{1}{\overline{h}_{l}^{2} S_{i'}}\right)}\right)^{-1}.$$
(3.56)

#### 3.3.3 MIMO Rayleigh-fading AWGN

We turn to the MIMO Rayleigh-fading AWGN channel. We assume that OFDMA is used, and that each resource is a frequency resource block with bandwidth  $W_{PRB}$ . Using the Gaussian approximation introduced in 2.2.4, the instantaneous throughput on a resource can be described by:

$$r_i^m \equiv \mathcal{N}(n_t \mu_i, \sigma_i^2), \tag{3.57}$$

with:

$$\begin{split} \zeta &= \frac{n_r}{n_t}, \\ \chi &= \frac{1}{2} \left[ 1 + \zeta + \frac{1}{S_i^m} - \sqrt{\left(1 + \zeta + \frac{1}{S_i^m}\right)^2 - 4\zeta} \right], \\ \mu_i &= \frac{W_{PRB}}{\log(2)} \left[ \zeta \log(1 + (1 - \chi)S_i^m) + \log(1 + S_i^m(\zeta - \chi) - \chi) \right], \\ \sigma_i^2 &= -\frac{W_{PRB}^2 \log\left(1 - \frac{\chi^2}{\zeta}\right)}{\log(2)^2}. \end{split}$$

with  $S_i^m$  the mean SINR on resource m. In the PF case and MTP case, the throughput is not available in a completely analytic form, however, they are given as one dimensional integrals involving the Gaussian distribution, and can be calculated rapidly using numerical integration. As shown in [32] if  $(n_t, n_r) \to +\infty$  with  $\frac{n_r}{n_t}$  bounded, we have that  $\frac{\sigma_i}{\mu_i} \to 0$ . Namely the instantaneous throughput becomes less variable when the number of antennas increase, and the benefits of channel-aware scheduling decrease. This effect is known as *channel hardening*.

#### $\mathbf{R}\mathbf{R}$

The RR throughput is:

$$\overline{r}_i^{RR} = \frac{M}{N} \mu_i. \tag{3.58}$$

We write F the c.d.f of the standard normal distribution.

#### MTP ( $\alpha = 0$ )

For the MTP scheduler, the throughput is given by the following integral:

$$\overline{r}_i^0 = \frac{M}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} (z\sigma_i + \mu_i) \left[ \prod_{j \neq i} F\left(\frac{\mu_i - \mu_j + z\sigma_i}{\sigma_j}\right) \right] e^{-\frac{z^2}{2}} dz.$$
(3.59)

**PF** (  $\alpha = 1$  )

For the PF scheduler, the throughput is calculated as the integral:

$$\overline{r}_i^1 = \frac{M}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} (z\sigma_i + \mu_i) \left[ \prod_{j \neq i} F\left(z\frac{\mu_i\sigma_j}{\mu_j\sigma_i}\right) \right] e^{-\frac{z^2}{2}} dz.$$
(3.60)

MMF (  $\alpha \rightarrow +\infty$  )

For the MMF, the throughput is:

$$\overline{r}_i^{\infty} = M\left(\sum_{i'=1}^N \frac{1}{\mu_{i'}}\right)^{-1}.$$
(3.61)

## **3.4** Numerical experiments

In this section we illustrate our scheduling gains calculations by numerical experiments. We compare our formulas with the results obtained by simulating the channel fading for all users and the  $\alpha$ -fair scheduler for 1000 time slots. A 95% confidence interval is given for any simulated value.

#### 3.4.1 Rayleigh-fading AWGN

We first consider the Rayleigh-fading AWGN, which is appropriate to model narrow-band fading and OFDMA systems such as LTE. Figure 3.1 shows the scheduling gain of a PF scheduler, with N users with  $S_i = 6dB \quad \forall i$ . Figure 3.2 shows the scheduling gain of a MTP scheduler, with N users with  $S_1 = 6dB$ ,  $S_i = 12dB$  for  $i \geq 2$ . We are interested in the gain of the first user. The gain is not the same for everyone since the scheduler allocates more resources to users in good radio conditions i.e with high mean SINR. We can see on both figures that our analytical formulas approximate the simulated values quite well, and we can also see on Figure 3.2 that the gain for user 1 decreases when N increases, since he has a smaller mean SINR.

Figure 3.3 shows the scheduling gain of an  $\alpha$ -fair scheduler, for 2 users with  $S_1 = 6dB$ ,  $S_2 = 12dB$ . This case illustrates what happens when a user is near the BS and the other one is far. By far we mean that either the user is physically far from the BS, or he is in an area with very deep shadow fading. In both cases this user has a low mean SINR. The larger  $\alpha$  is, the larger is the gain for users with poor channel conditions, and so it is possible to manage the coverage for users at cell edge by adjusting  $\alpha$  dynamically.

88



Figure 3.1: PF scheduling gain as a function of the number of users for  $S_i = 6 dB \; \forall i$ 



Figure 3.2: MTP scheduling gain as a function of the number of users for user 1 with  $S_1 = 6dB$  and  $S_i = 12dB$  for  $i \ge 2$ 



Figure 3.3: Scheduling gain as a function of  $\alpha$  for 2 users and  $S_1=6dB$  ,  $S_2=12dB$ 

We have assumed that the fast-fading of the interfering signals can be ignored since the number of interfering signals is large, meaning that the only source of variability of the instantaneous SINR is the fading of the useful signal. We simulate here the distribution of the instantaneous SINR when the useful signal as well as interfering signals are fading and follow a Rayleigh model. We compare it with the exponential distribution, using the model for path loss and shadowing described in appendix 7.1, only considering interference from first-tier neighbors. Figure 3.4 shows the p.d.f of the instantaneous SINR (normalized by the mean SINR) for cell center users and cell edge users, and we can see that those distributions are very close to the exponential distribution. Furthermore, we can see that the exponential approximation is worse when considering cell edge users compared with cell center users. This is logical since cell edge users have one or two dominant interferers, while cell center users are equally interfered by the 6 neighboring cells.

### 3.4.2 Multi-tap Rayleigh-fading AWGN

We turn to the multi-tap Rayleigh-fading AWGN channel, which is appropriate to model wide-band fading and CDMA systems such as High Speed



Figure 3.4: Impact of the interference fading on the instantaneous SINR distribution

Packet Access (HSPA). We show the scheduling gain of a PF scheduler with up to 10 users, all users having a mean SINR of 6dB. Figure 3.5 shows the scheduling gain of a PF scheduler with the Vehicular A model obtained by formula (3.55) and by simulation. Figure 3.6 shows the scheduling gain of the PF scheduler for the models stated previously, and we can see that it decreases appreciably depending on the fading model, which shows that frequency-selectivity results in smaller scheduling gains. We can conclude that frequency-selectivity is an adverse effect which diminishes the diversity gain of the PF scheduler.

#### 3.4.3 MIMO Rayleigh-fading AWGN

Lastly let us consider the MIMO Rayleigh-fading AWGN channel, which is appropriate to model narrow-band fading (OFDMA) with multiple antennas, as it is the case for LTE systems featuring MIMO. We compare the distribution of the capacity of a MIMO channel with its Gaussian approximation, for  $n_t = n_r = 2$ . We draw the channel matrix 10000 times and calculate the corresponding capacity distribution (with formula (2.77)), which we compare to the Gaussian distribution with mean and variance given by (3.57). Figure 3.7 shows the comparison of the mean of the two distributions for different val-



Figure 3.5: PF scheduling gain as a function of the number of users for  $S_i = 6dB \; \forall i$ , Vehicular A model



Figure 3.6: Comparison of PF scheduling gain for different propagation models



Figure 3.7: Mean capacity for MIMO 2x2, comparison between asymptotic distribution and simulations.

ues of the mean SINR. Figure 3.8 shows the comparison of the c.d.f of the two distributions for a mean SINR of 5dB. We can see on those two figures that the values obtained by the Gaussian approximation are very close to the simulated values obtained by drawing channel matrices. Hence approximating the distribution of the capacity by a Gaussian distribution is reasonable, even when  $n_t = n_r = 2$ . Figure 3.9 shows the throughput per user of a PF scheduler with  $S_i^m = 5dB$ ,  $\forall i, m$ . For MIMO 2x2, the PF scheduling gain is of 2.2 for 10 users, while it is of 2.7 in the single antenna case (see figure 3.1). The effect of channel hardening is already visible for MIMO 2x2 and considerably diminishes the benefits of channel-aware scheduling.

## 3.5 Coverage-capacity optimization

#### 3.5.1 Algorithm

Based on the scheduling gain calculations of the previous section, we propose a simple and efficient SON algorithm that optimizes cell-coverage while minimizing capacity losses by adjusting  $\alpha$  dynamically. We say that a user is covered if his mean throughput is higher than a certain fixed threshold  $Th_{min}$ , which is a parameter of the service we are considering, for example



Figure 3.8: c.d.f for MIMO 2x2, comparison between asymptotic distribution and simulations, for mean SINR 5dB.



Figure 3.9: PF scheduler, MIMO 2x2, scheduling gain with  $S_i^{(k)} = 5dB$ ,  $\forall i, k$ . Comparison between simulations and formula (3.60).

#### 3.5. COVERAGE-CAPACITY OPTIMIZATION

the minimal throughput to watch a video with the lowest quality.

First let us state the optimization objective: we consider a particular service with the corresponding  $Th_{min}$  and we want to change the  $\alpha$  parameter dynamically in order to cover the maximum number of users, using the above definition for coverage. However, we have to be careful since increasing  $\alpha$  can potentially increase the number of covered users, but also diminishes the global cell throughput. Therefore we want to find the minimal  $\alpha$  that covers the maximum number of users.

To this end, the formula for the scheduling gain with  $\alpha = +\infty$  is of particular interest: if  $\alpha = +\infty$  results in covering all users, this means that we can cover everyone providing that  $\alpha$  is large enough. If nobody is covered, namely the users with low mean SINR cannot be covered, and we should not allocate any resource to them.

In order to determine the users that can be covered with large enough  $\alpha$ , we ignore the user with the worst mean SINR, recalculate the  $\alpha = +\infty$  throughput and keep doing so until we are able to cover everyone.

The algorithm proceeds the following way: at each iteration it observes the number of covered users, then it determines the users that can be covered using the technique stated above, and finally the  $\alpha$  is adjusted. If some of the users that could have been covered were not covered, the  $\alpha$  is increased, and if all coverable users have been covered, the  $\alpha$  is diminished with a small probability  $P_{\epsilon}$ , and stays the same with probability  $1 - P_{\epsilon}$ . The idea is that the environment might have changed, and that the current  $\alpha$  might not be the lowest that enables us to cover all coverable users.  $P_{\epsilon}$  therefore shall be chosen to reflect the speed at which the environment changes.

The following notations are used: we consider BS s;  $\alpha_s$  is the value of  $\alpha$  for s,  $N_s$  - the number of users that s can cover and  $\tilde{N}_s$  - the number of users effectively covered at the last period.  $(\alpha^{(j)})_{1 \leq j \leq J_{max}}$  is the allowed set of values of  $\alpha$ , e.g.  $\{1, ..., 5\}$  in the present work.  $j_s$  is the index of the current  $\alpha$ , namely  $\alpha_s = \alpha^{(j_s)}$ . The algorithm is described in Table 3.1.

It is noted that in Table 3.2 it is sufficient to calculate the throughput of a user in  $i \in \{1, ..., N\} \setminus I$  since the MMF scheduler allocates the same throughput to all users in  $i \in \{1, ..., N\} \setminus I$  and allocates 0 to users in I.

It is noted that this algorithm has all the necessary features to be a robust and implementable SON algorithm: it is decentralized since each station adjusts its own parameters according to its own KPIs without any communication with neighboring cells; it is not computationally demanding; and it is scalable since the introduction of new BSs does not disturb its functioning. For each BS s: Initial phase: 1. Calculate  $N_s$  using (Table 3.2) 2. Try every  $\alpha_s \in (\alpha^{(j)})_{1 \leq j \leq J_{max}}$  once 3. Choose the minimal  $j_s$  so that  $\alpha_s = \alpha^{(j_s)}$ that covers  $N_s$  users. Repeat: 4. Calculate  $N_s$  using (Table 3.2) 5. Set  $\alpha_s = \alpha^{(j_s)}$  and observe resulting  $\tilde{N}_s$ If  $\tilde{N}_s < N_s$ : 6.  $j_s \leftarrow \min(j_s + 1, J_{max})$ If  $n_k = N_k$ : 7.  $j_s \leftarrow \begin{cases} \max(j_s - 1, 1) \text{ with probability } P_\epsilon \\ j_s \text{ with probability } (1 - P_\epsilon) \end{cases}$ 

Table 3.1:	Capacity	coverage	algorithm
------------	----------	----------	-----------

Initial phase: 1.  $I = \emptyset$ 2. Calculate  $\overline{r}_i^{\infty}$  for a certain  $i \in \{1, ..., N\} \setminus I$ While  $\overline{r}_i^{\infty} < Th_{min}$ : 3.  $i^* = \arg\min_{i \in \{1,...,N\} \setminus I} S_i$ 4. Add  $i^*$  to I5. Calculate  $\overline{r}_i^{\infty}$  for a certain  $i \in \{1, ..., N\} \setminus I$ ignoring users in IResult:  $6.N_k = N - |I|$ 



### 3.5.2 Admission Control

It shall be noted that the MMF throughput is also useful to define an *ad*mission control rule. Given the mean SINR of the users in a cell, if a new user arrives, we can calculate the throughput of the MMF scheduler and determine whether we are able to cover this user with  $\alpha$  sufficiently large. If it is not the case the new user shall not be admitted. The benefit of such an admission rule over traditional methods is that we can be sure that we are always able to cover all users if they do not move too fast, so that their mean SINR does not change too drastically over time. Furthermore since calculating the MMF throughput simply involves looking at most N times in a table of values, N being the number of users in the cell, this is a practically implementable admission rule.

#### 3.5.3 Simulation

We implement the coverage-capacity algorithm described above in a realistic OFDMA network simulator with 33 stations to observe its average performance. We use a *semi-dynamic network simulator* with time resolution of 1s (see [53] for a detailed description of a semi-dynamic simulator). The propagation model is explained in appendix 7.1. Users arrive in the network according to a Poisson process. Admission control is done with the algorithm described previously. We consider a streaming service with constant session length. A user quits the service if he is not covered during 10 consecutive seconds. The number of users that quit the service in such a way is a measure of coverage, and we show that the proposed algorithm reduces it appreciably.

We compare the proposed algorithm to a reference scenario in which BSs apply a PF scheduler all the time, that is  $\alpha_s = 1$ ,  $\forall s$ . It is noted that admission control is the same for both algorithms for the comparison between the proposed algorithm and the reference one, hence the coverage improvement is not related to the admission control strategy. Table 3.3 summarizes the relevant simulation parameters.

#### 3.5.4 Simulation Results

Figure 3.10 shows the evolution of  $\alpha$  during the simulation for a particular BS, and Figure 3.11 the number of users served by this BS. We can clearly see that the algorithm keeps  $\alpha$  low when the number of users is small, in order not to loose capacity, and increases  $\alpha$  when the number of users increases in order to keep all users covered. Figure 3.12 shows the percentage of users that have left the network because of a lack of coverage, namely because they did not receive the minimal bitrate for 10 consecutive seconds as described above. The proposed algorithm allows to reduce the percentage of users leaving the network from 4% which is generally considered unacceptable in terms of QoS to less than 1%. Figure 3.13 shows the average BS throughput. The capacity loss caused by the coverage improvement is on average 4%, which is a relatively small price to pay for the important reduction of calls dropped because of coverage loss. It is noted that the drop call rate is a much more important QoS metric than the global system throughput.



Figure 3.10: Evolution of  $\alpha$  as a function of time for a BS.



Figure 3.11: Number of users in a BS as a function of time.



Figure 3.12: Number of users leaving because of lack of coverage as a function of arrival rate.



Figure 3.13: Average BS throughput as a function of arrival rate.

Simulator parameters			
Spatial resolution	$25m \times 25m$		
Time resolution	1s		
Simulation time	10000s		
User speed	5km/h		
Average session length	120s		
Coverage threshold	256kb/s		
Network parameters			
Number of Physical Resource Blocks (PRBs)	12		
Size of a PRB	180kHz		
Number of stations	33		
Cell layout	$11 \text{ eNB's} \times 3 \text{ sectors}$		
Average inter-cell distance	1km		
Type of service	Streaming		
Propagation			
Thermal noise	-174dBm/Hz		
Path $loss(d \text{ in } km)$	$128 + 37.6 \log_{10}(d)  \mathrm{dB}$		
Shadowing standard deviation	6 dB		
Antenna configuration	MIMO $2 \times 2$		

Table 3.3: Model parameters

## Chapter 4

## Load balancing

In this chapter we study the problem of load balancing in cellular networks. In operational networks, certain BSs are noticeably more loaded than others due to the scarcity of available locations for deploying BSs and the nonuniformity of mobile traffic. Those imbalances in BSs load evolves through the day due to daily traffic patterns. Typical situations include:

- business districts at the end of the afternoon when customers leave their office to go back home,
- a stadium at the end of a football match, where a large amount of customers call as soon as the match ends,
- train and subway stations during busy hours.

In these situations, the loads of a small group of BSs become too high to ensure good QoS in the area and a large amount of dropped/blocked calls occur. The simplest solution is to over-dimension the network in those areas and deploy a large number of BSs to cope with the traffic peak. This solution is simple but costly and might not be practical if sites with good propagation conditions are scarce.

The solution we explore here is based on SON: BSs can monitor their loads and control the amount of traffic they absorb to avoid overload. Users attach themselves to the BS with the strongest received pilot power. If a BS is more loaded than its neighbors then it diminishes its transmitted pilot power, which causes the geographical area it serves to shrink, and reduces its load by letting its neighbors absorb more traffic. An update equation for the pilot powers is given to equalize the loads. The load balancing algorithm is distributed since it requires that BSs exchange their loads with a single reference BS. In practical settings, a load balancing procedure can be triggered by an alarm based on BSs loads. The load balancing can then be applied to the overloaded BS and group of its neighbors. If used this way, our algorithm is fully scalable.

We consider *flow-level dynamics*: users enter the network dynamically according to a random arrival process, download a file and exit the network when the download is complete. The *load estimates* depend on random arrivals and departures, so they are by nature noisy. Using stochastic approximation, we prove that the proposed algorithm converges to a set on which all loads are equal in spite of the load estimation noise. Our algorithm ensures that the network is stable (users do not accumulate to infinity in at least one of the BSs) whenever it is possible.

Our algorithm is designed to work on a time scale which is slower than users arrivals and departures, but faster than the speed at which traffic intensity and distribution changes. Namely, each BS estimates its load and changes its transmit power every T seconds. T should be larger than the typical flow duration (a few seconds), and faster than the time scale on which traffic changes (at least several tens of minutes). If T is too small, users which arrive at the cell edge undergo a large amount of handovers during their sojourn in the network which creates both high overhead and dropped calls. If T is too large, the algorithm will not be able to track the changes in traffic, which is the main point of SON. The order of magnitude of T should be of 1 minute to ensure good performance.

The algorithm proposed here is a *local* algorithm, since it is a applied to a station and its neighbors only. An interesting extension of this work would be to analyze the convergence of a gossip-like ([19]) procedure where this local algorithm is applied at multiple locations asynchronously.

The material of this chapter is based on our contributions [26, 8]. The load balancing was introduced in [26] in the context of macro-cell networks, and load balancing in the context of femto-cells networks was analyzed in [8]. The load balancing mechanism is extended to relay-enhanced networks in chapter 5. Flow-level analysis for wireless networks was introduced in [14]. [17] showed that gains of channel-aware scheduling could be included in the model. This in particular makes the calculations of scheduling gains of chapter 3 fully applicable. The proofs for all the results can be found in section 4.4.

## 4.1 Flow-level dynamics

#### 4.1.1 Traffic model

We consider a wireless network in a downlink scenario, serving elastic traffic, without user mobility. The network area  $\mathbb{A} \subset \mathbb{R}^2$  is bounded and convex. Users enter the network according to a Poisson process on  $\mathbb{A} \times \mathbb{R}$  with measure  $\lambda(dr \times dt) = \lambda(r)dr \times dt$ ,  $r \in \mathbb{A}$ . This point process models the instants of arrivals and their locations. Users download a file of size  $\sigma$ , with  $E[\sigma] < +\infty$ and we assume independence between the arrival process and the file sizes. Users do not move while they are downloading their file and they leave the network when the download finishes. There are  $N_{BS}$  BSs, with  $\mathbb{A}_s$  the area served by BS s. We write  $R_s(r)$  the data rate of a user located at r served by BS s when there are no other users in s.

The time scale on which users arrive and depart is much slower than fastfading, so that in each state of the network, each user experiences a data rate which is equal to their data rate averaged on the fast-fading. The effect of fast-fading is only visible in our model through its impact on the average data rate.

As done in [14], the network can be modeled by  $N_{BS} M/G/1/PS$  (see 2.1.2) queues and the stability region of the network is given by Theorem 4.1.

**Theorem 4.1.** Define the load of BS s,  $\rho_s$  by:

$$\rho_s = E\left[\sigma\right] \int_{\mathbb{A}_s} \frac{\lambda(r)}{R_s(r)} dr, \qquad (4.1)$$

and BS s is stable if  $\rho_s < 1$ , and unstable if  $\rho_s > 1$ . The network is stable if  $\max_s \rho_s < 1$ , and unstable if  $\max_s \rho_s > 1$ .

A BS is *stable* if the probability distribution of the number of active users in this BS tends to a stationary distribution. A BS is *unstable* if the number of active users grows to infinity. The total *traffic intensity* is  $E[\sigma] \int_{\mathbb{A}} \lambda(r) dr$ , and the *network capacity* is the maximal value of the total traffic intensity which ensures stability of the network.

Assume that when  $n_s$  users are simultaneously served by BS s, the data rate of a user served by BS s and located at r is  $\frac{R_s(r)g(n_s)}{n_s}$ , with  $n \to g(n)$  a non-decreasing function. We define the maximal gain:

$$g^* = \lim_{n \to +\infty} g(n). \tag{4.2}$$

The function g(n) stands for the multi-user diversity gain which is a characteristic of wireless networks, due to fast-fading. We have shown in chapter 3 that g(n) could be calculated for most physical layer models of interest. Then Theorem 4.1 applies when replacing  $\rho_s$  by  $\frac{\rho_s}{q^*}$ . See for example ([17]).

#### 4.1.2 Load estimation

A user arriving in the network attaches himself to the BS with the strongest received pilot signal, and the BSs can modify their transmit pilot power in order to adjust their loads. We show a method to do so based solely on network measurements (user feedback), in a distributed way, and with minimal information exchange between BSs. BSs do not change the power they transmit on data channels, so that the data rates  $R_s(r)$  are constant. Only the pilot power can change, and hence the zones  $\mathbb{A}_s$  served by the BSs. We define  $P_s$  the transmitted pilot power of BS s, and  $P = \{P_s\}_{1 \leq s \leq N_{BS}}$  - the corresponding vector.

We show that the cell load can be estimated without bias, and propose a power update mechanism. The convergence of the power update mechanism to an optimal configuration is demonstrated by studying an associated ODE.

Time is divided into slots of size T > 0, and the k-th time slot is [kT, (k+1)T). Let  $P_s[k]$  be the pilot power transmitted by BS s during the k-th time slot,  $P[k] = \{P_s[k]\}_{1 \le s \le N_{BS}}$  - the corresponding vector. We write  $\{T_n, r_n, \sigma_n\}_{n \in \mathbb{Z}}$  the instants of arrival of users, their location and their file size respectively.

A load estimate for BS s is the amount of workload arriving in BS s during [kT, (k+1)T), divided by T:

$$\rho_s[k] = \frac{1}{T} \sum_{n \in \mathbb{Z}} \frac{\sigma_n}{R_s(r_n)} \mathbf{1}_{[kT,(k+1)T) \times \mathbb{A}_s}(T_n, r_n).$$

$$(4.3)$$

The estimate (4.3) is unbiased and has finite variance, as stated by Theorem 4.2.

#### Theorem 4.2.

$$E\left[\rho_s[k]\right] = \rho_s(P[k]),\tag{4.4}$$

and if  $E[\sigma^2] < +\infty$ ,

$$\sup_{k\in\mathbb{N}} E\left[\rho_s[k]^2\right] < +\infty. \tag{4.5}$$

## 4.2 Load balancing mechanism

#### 4.2.1 Update equation

We propose the following load balancing mechanism:

$$P_s[k+1] = P_s[k](1 + \epsilon_k(\rho_1[k] - \rho_s[k])).$$
(4.6)

#### 4.2. LOAD BALANCING MECHANISM

Since load estimates are unbiased,

$$E[\rho_1[k] - \rho_s[k]] = \rho_1(P[k]) - \rho_s(P[k]), \qquad (4.7)$$

and this suggests looking at the mean ODE:

$$\dot{P}_s = P_s[\rho_1(P) - \rho_s(P)].$$
(4.8)

We have chosen BS 1 as the reference BS without loss of generality i.e.  $P_1[k] = P_1[0]$ ,  $k \in \mathbb{N}$ , since the reference BS can be changed by permutation of indices. The rationale behind such a mechanism is that if BS s is less loaded than BS 1 then  $P_s$  increases, and  $\rho_s(P)$  should (intuitively) increase. We prove that all solutions of the ODE converge to a set on which the loads of all BSs is equal.

#### 4.2.2 The mean ODE

We demonstrate several properties of the ODE (4.8). We use the model described in section 7.1 for signal attenuation, without shadowing. The location of BS s is  $r_s \in \mathbb{A}$ , and the signal attenuation between BS s and location  $r \in \mathbb{A}$  is  $L(r_s, r)$ . Users attach themselves to the BS with the strongest received pilot signal:

$$\mathbb{A}_s = \{ r \in \mathbb{A} : s \in \operatorname*{arg\,max}_{s'} L(r_{s'}, r) P_{s'} \}.$$

$$(4.9)$$

Assumption 4.1. Data rates are upper and lower bounded:

$$0 < R_{min} \le R_s(r) \le R_{max} < +\infty, \,\forall s, r.$$

$$(4.10)$$

As a consequence:

$$\frac{E\left[\sigma\right]}{R_{max}} \int_{\mathbb{A}_s} \lambda(r) dr \le \rho_s \le \frac{E\left[\sigma\right]}{R_{min}} \int_{\mathbb{A}_s} \lambda(r) dr.$$
(4.11)

**Theorem 4.3.** Under Assumptions 4.1,  $P \rightarrow \rho_s(P)$  is Lipschitz continuous on:

$$\mathcal{P} = [P_{min}, +\infty)^{N_{BS}},\tag{4.12}$$

with  $P_{min} > 0$ .

This result is used to prove unicity of the solutions to the ODE.

#### 4.2.3Convergence of the load balancing mechanism

An important property is that the maximal load decreases on the trajectories of the ODE, as shown by theorem 4.4.

**Theorem 4.4.** Under Assumptions 4.1:

(i) Given an initial condition, the ODE (4.8) has a unique solution defined on  $\mathbb{R}^+$ . Furthermore it verifies

$$0 < \inf_{t \in \mathbb{R}^+} P_s(t) \le \sup_{t \in \mathbb{R}^+} P_s(t) < +\infty.$$

$$(4.13)$$

(ii) The set:

$$\mathcal{L} = \{P : \max_{o} \rho_s(P) = \min_{o} \rho_s(P)\}, \qquad (4.14)$$

is a Lyapunov stable attractor of the ODE (4.8).

This theorem has the following consequences: first since the solution is unique given an initial condition, the asymptotic behaviour of the system can be evaluated numerically by standard numerical analysis techniques. Furthermore the transmit power of each BS can never be 0 and remains bounded. Finally, the solution converges to a set on which the loads of all BSs are equal, namely it performs *load balancing*. We write

$$\rho_{\infty} = \sup_{P \in \mathcal{L}} \max_{s} \rho_{s}(P).$$
(4.15)

 $\rho_{\infty} < 1$  implies that the algorithm achieves stability regardless of the initial condition. The algorithm increases the capacity of the network, and the increase in capacity can be computed by evaluating  $\rho_{\infty}$  numerically.

Finally we show the link between the asymptotic behaviour of the discrete algorithm (4.6) and the previously studied ODE (4.8) through a stochastic approximation result.

#### **Theorem 4.5.** Decreasing step sizes Assume:

$$\sum_{k\in\mathbb{N}}\epsilon_k = +\infty, \ \sum_{k\in\mathbb{N}}\epsilon_k^2 < +\infty,$$
(4.16)

then

$$\max_{s} \rho_{s}[k] - \min_{s} \rho_{s}[k] \xrightarrow[k \to +\infty]{a.s} 0, \qquad (4.17)$$

and:

$$\limsup_{k \to +\infty} \max_{s} \rho_{s}[k] \le \rho_{\infty}.$$
(4.18)

Constant step sizes When  $\epsilon_k = \epsilon > 0$  a constant, then, for all  $\mu > 0$ :

$$\limsup_{k \to +\infty} \mathbf{P}\left[ |\max_{s} \rho_{s}[k] - \min_{s} \rho_{s}[k]| > \mu \right] = o(\epsilon), \tag{4.19}$$

and:

$$\limsup_{k \to +\infty} \mathbf{P}\left[\max_{s} \rho_{s}[k] > \mu + \rho_{\infty}\right] = o(\epsilon).$$
(4.20)

#### 4.2.4 Extension to constant data rate traffic

This work was mainly motivated by *elastic traffic*, but most of the underlying ideas are applicable to *constant data rate traffic* (streaming and voice traffic) as well. Constant data rate traffic is modeled using the *multi-rate Erlang model* described in subsection 2.1.2. The radio resources are divided into C resource blocks, and users are allocated enough resource blocks to reach a fixed target data rate  $R_{st}$ . A user located at r served by BS s requires  $\left\lceil \frac{CR_{st}}{R_s(r)} \right\rceil > C$  then all users arriving at r are blocked. We remove such points from the network area  $\mathbb{A}$ . Users stay in the network an exponentially distributed amount of time with parameter  $\mu$ . The streaming load for BS s denoted  $\rho_s^{(st)}$  is calculated by:

$$\rho_s^{(st)} = \frac{1}{C\mu} \int_{\mathbb{A}_s} \left[ \frac{CR_{st}}{R_s(r)} \right] \lambda(r) dr.$$
(4.21)

In particular if C is large so that the granularity of resources is small, the streaming load can be approximated by:

$$\rho_s^{(st)} \approx \frac{R_{st}}{\mu} \int_{\mathbb{A}_s} \frac{\lambda(r)}{R_s(r)} dr.$$
(4.22)

which is the same expression as for elastic traffic, where the mean flow size  $E[\sigma]$  has been replaced by  $\frac{R_{st}}{\mu}$  which is the average amount of data received by a user during a session.

The loads can be estimated without bias in the same way as before:

$$\rho_s^{(st)}[k] = \frac{1}{T} \sum_{n \in \mathbb{Z}} \frac{\Gamma_n}{C} \left[ \frac{CR_{st}}{R_s(r_n)} \right] \mathbf{1}_{[kT,(k+1)T) \times \mathbb{A}_s}(T_n, r_n), \qquad (4.23)$$

where  $\Gamma_n$  denotes the amount of time that the *n*-th user stays in the network with  $E[\Gamma_n] = \frac{1}{\mu}$ . Using the same power update equation 4.6, the loads can be equalized.

There is one difference with the elastic traffic case: for constant data rate traffic, the main performance indicator is the *blocking rate*, and there is no instability for loads higher than 1. There exists pathological cases where the blocking rate is not a monotonously increasing function of the load. Increasing the load can cause to block more users who need a large amount of resources to reach the target data rate  $R_{st}$ , and reduce the blocking rate for classes of users who need a small amount of resources.

In the elastic traffic case, the rationale behind equalizing the loads is to ensure *stability* of all BSs whenever it is possible. For constant data rate traffic, we cannot prove mathematically that equalizing the loads reduces the network blocking rate. However, in practical scenarios, if some BSs are much more congested than their neighbors, it is fair to say that equalizing loads should improve the network performance by offloading the critically loaded BSs.

## 4.3 Numerical experiments

We assess the performance gains of the proposed algorithm numerically. The parameters of the network model are given in Table 4.1. We apply a small random perturbation to the BSs locations, because in the case of a perfectly hexagonal network, all cells have the same load, and there is no point in trying to perform load balancing. The asymptotic behaviour of the proposed algorithm (Theorem 4.5) is described by the ODE (4.8). To evaluate the performance gains numerically, we choose an initial power configuration uniformly distributed in  $\mathcal{P}$ , and find the corresponding limit set numerically. We repeat the process several times to obtain several limit sets, and for each of them we calculate the network capacity. Figure 4.1 shows the complementary cumulative distribution function (c.c.d.f) of the network capacity improvement on the limit sets obtained by the procedure described above. The capacity improvement is calculated with respect to a reference scenario in which all BSs transmit the same power. We observe a performance gain of 36% in the worst case and 45% in the best case. The difference between the best and worst case is not very large, suggesting that the proposed method achieves a good performance without a global search. The gain in term of network capacity is considerable. Figure 4.2 compares the behaviour of the discrete time algorithm obtained by simulating user arrivals with the corresponding trajectory of the ODE. The asymptotic behaviour of the discrete time algorithm is indeed described by the ODE.

108
Model parameters	
Network layout	Hexagonal
Antenna type	Tri-sector
Number of BSs	16 sites $\times$ 3 sectors
Inter-site distance	500m
Network Area	$1km \times 1km$
Access technology	OFDMA
Link Model	SISO, AWGN + Rayleigh fading
Number of resource blocks	12
Resource block size	180kHz
BS maximal transmit power	46dBm
Thermal noise	-174 dBm/Hz
Path loss model	$128 + 37.6 \log_{10}(d) \text{ dB}, d \text{ in km}$
File size	10Mbytes

Table 4.1: Model parameters

# 4.4 Proofs

# 4.4.1 Proof of Theorem 4.2

*Proof.* Applying the Campbell formula we have that:

$$E\left[\rho_s[k]\right] = \frac{E\left[\sigma\right]}{T} \int_{[kT,(k+1)T) \times \mathbb{A}_s(P[k])} \frac{\lambda(r)}{R_s(r)} dt dr = \rho_s(P[k]). \tag{4.24}$$

The number of users entering the network in a finite time interval has finite second moment, so that we obtain the bound:

$$E\left[\rho_s[k]^2\right] \le \frac{1}{R_{min}^2} \left(\int_{\mathbb{A}} \lambda(r) dr\right)^2 \left(E\left[\sigma\right]^2 + \frac{E\left[\sigma^2\right]}{T}\right),\tag{4.25}$$

and

$$\sup_{k \in \mathbb{N}} E\left[\rho_s[k]^2\right] < +\infty,\tag{4.26}$$

concluding the demonstration.

# 4.4.2 Proof of Theorem 4.3

Proof. The 2 BSs case



Figure 4.1: Cell size optimization: c.c.d.f of performance gains on limit sets of the ODE

We first consider  $\mathbb{A} = [-X_{max}, X_{max}]^2$ , and two BSs. BS 1 is located at  $(-\frac{d}{2}, 0)$ , and BS 2 at  $(\frac{d}{2}, 0)$ . By solving the algebraic equation:

$$P_1((x+\frac{d}{2})^2+y^2)^{-\frac{\alpha}{2}} = P_2((x-\frac{d}{2})^2+y^2)^{-\frac{\alpha}{2}},$$
(4.27)

we have that:

- If  $P_1 = P_2$ ,  $\mathbb{A}_1 = \{(x, y) | -X_{max} \le x \le 0, -X_{max} \le y \le X_{max} \}$
- -~ If  $P_1 < P_2$  ,  $\mathbbm{A}_1$  is the intersection between  $\mathbbm{A}$  and a disk of radius

$$r(P_1, P_2) = d \frac{P_1^{-\frac{1}{\alpha}} P_2^{-\frac{1}{\alpha}}}{|P_1^{-\frac{2}{\alpha}} - P_2^{-\frac{2}{\alpha}}|},$$
(4.28)

centered at  $(-c(P_1, P_2), 0)$  with:

$$c(P_1, P_2) = \frac{d}{2} \frac{P_1^{-\frac{2}{\alpha}} + P_2^{-\frac{2}{\alpha}}}{|P_1^{-\frac{2}{\alpha}} - P_2^{-\frac{2}{\alpha}}|}.$$
(4.29)

– If  $P_1 > P_2$ ,  $\mathbb{A}_2$  is the intersection between  $\mathbb{A}$  and a disk of radius  $r(P_1, P_2)$  centered at  $(c(P_1, P_2), 0)$ .



Figure 4.2: Cell size optimization: comparison between the discrete time algorithm and the ODE

Assume that  $P_1 < P_2$ :

$$|\mathbb{A}_1(P_1, P_2)| = \int_0^\pi R(\theta, P_1, P_2)^2 d\theta, \qquad (4.30)$$

with:

$$R(\theta, P_1, P_2)^2 = \min(r(P_1, P_2)\sin(\theta), X_{max})^2 + \max(r(P_1, P_2)\cos(\theta), c(P_1, P_2) - X_{max})^2.$$
(4.31)

Since both  $(P_1, P_2) \to r(P_1, P_2)$  and  $(P_1, P_2) \to c(P_1, P_2)$  are bounded with bounded derivatives in a neighborhood of  $(P_1, P_2), (P_1, P_2) \to |\mathbb{A}_1(P_1, P_2)|$  is locally Lipschitz continuous at  $(P_1, P_2)$ . By symmetry, the same is true for  $P_1 > P_2$ .

Assume that  $P_2 > P_1 > 0$  and  $|P_2 - P_1| \le \epsilon$ , then there exists  $K_4 > 0$  such that:

$$||\mathbb{A}_1(P_1, P_2)| - |\mathbb{A}_1(P_1, P_1)|| \le K_4 \frac{\epsilon}{P_1} + o(\epsilon),$$
(4.32)

hence Lipschitz continuity is valid on  $\mathcal{P}$ .

Arbitrary number of BSs

We consider the general case with an arbitrary number of BSs. We consider  $N_{BS}$  BSs,  $P^{(1)} \in \mathcal{P}$ ,  $P^{(2)} \in \mathcal{P}$ , and without loss of generality  $\rho_s(P^{(1)}) \geq \rho_s(P^{(2)})$ . Let  $P^{(3)} \in \mathcal{P}$ ,  $P^{(4)} \in \mathcal{P}$  with

$$P_{s'}^{(3)} = \min(P_{s'}^{(1)}, P_{s'}^{(2)}), \ s' \neq s$$
(4.33)

$$P_s^{(3)} = \max(P_s^{(1)}, P_s^{(2)}), \tag{4.34}$$

and:

$$P_{s'}^{(4)} = \max(P_{s'}^{(1)}, P_{s'}^{(2)}), \qquad (4.35)$$

$$P_s^{(4)} = \min(P_s^{(1)}, P_s^{(2)}). \tag{4.36}$$

We use the notation  $||P||_{\infty} = \max_{s} |P_{s}|$ . It is noted that  $||P^{(2)} - P^{(1)}||_{\infty} = ||P^{(3)} - P^{(4)}||_{\infty}$ . Since:

$$|\rho_s(P^{(1)}) - \rho_s(P^{(2)})| \le |\rho_s(P^{(3)}) - \rho_s(P^{(4)})|$$
(4.37)

and:

$$\mathbb{A}_s(P^{(4)}) \subset \mathbb{A}_s(P^{(3)}), \tag{4.38}$$

then:

$$|\rho_s(P^{(3)}) - \rho_s(P^{(4)})| \le K_2 |\mathbb{A}_s(P^{(3)}) \setminus \mathbb{A}_s(P^{(4)})|.$$
(4.39)

We write:

$$\mathbb{A}_{s,s'}(P) = \{ r | L(r_s, r) P_s \ge L(r_{s'}, r) P_{s'} \},$$
(4.40)

and

$$\mathbb{A}_{s}(P) = \bigcap_{s' \neq s} \mathbb{A}_{s,s'}(P). \tag{4.41}$$

Furthermore:

$$\mathbb{A}_s(P^{(3)}) \setminus \mathbb{A}_s(P^{(4)}) \subset \bigcup_{s' \neq s} (\mathbb{A}_{s,s'}(P^{(3)}) \setminus \mathbb{A}_{s,s'}(P^{(4)})).$$
(4.42)

Hence we have that:

$$|\rho_s(P^{(1)}) - \rho_s(P^{(2)})| \le \sum_{s' \ne s} |\mathbb{A}_{s,s'}(P^{(3)}) \setminus \mathbb{A}_{s,s'}(P^{(4)})|, \qquad (4.43)$$

which proves the result, since there exists  $K_3$  so that

$$|\mathbb{A}_{s,s'}(P^{(3)}) \setminus \mathbb{A}_{s,s'}(P^{(4)})| \le K_3 ||P^{(3)} - P^{(4)}||, \qquad (4.44)$$

by using the result obtained for two BSs.

112

### 4.4.3 Lemma 4.1

The following lemma is a direct consequence of the *enveloppe theorem*.

**Lemma 4.1.** Let  $x : \mathbb{R} \to \mathbb{R}^n$ , absolutely continuous with almost everywhere (a.e) derivative  $\dot{x}(t)$ . Then  $t \to \min_s x_s(t)$  and  $t \to \max_s x_s(t)$  are absolutely continuous, with derivatives:

$$\dot{x}_{s(t)}(t), \ s(t) \in \{ \arg\min_{s} x_s(t) \},$$
(4.45)

and:

$$\dot{x}_{s(t)}(t), \ s(t) \in \{ \arg\max_{s} x_s(t) \}.$$
 (4.46)

# 4.4.4 Proof of Theorem 4.4

*Proof.* (i) Since  $\min_{s} P_{s}(0) > 0$ , Theorem 4.3 states that the cell loads are Lipschitz continuous in a neighbourhood of P(0). Hence  $P \to (\rho_{1}(P) - \rho_{s}(P))$  is Lipschitz continuous in a neighbourhood of P(0), and the Picard-Lindelöf theorem ensures that there exists a unique local solution given an initial condition in  $\mathcal{P}$ .

Upper bound

Consider such a local solution defined on  $[0, \delta)$ ,  $t \in [0, \delta)$ , and assume  $P_s(t) = \max P_s(t) > P_{max}$ , then:

$$\rho_1(P(t)) \le \rho_1(P_1(0), 0, \cdots, 0, P_{max}, 0, \cdots, 0), \tag{4.47}$$

and

$$\rho_s(P(t)) \ge \rho_s(P_{max}, \cdots, P_{max}) = \rho_s(1, \cdots, 1). \tag{4.48}$$

Since:

$$\rho_1(P_1(0), 0, \cdots, 0, P_{max}, 0, \cdots, 0) \xrightarrow[P_{max} \to +\infty]{} 0, \qquad (4.49)$$

and  $\rho_s(1, \dots, 1) > 0$ , there exists a value of  $P_{max}$  such that if  $P_s(t) = \max P_s(t) > P_{max}$  then  $\dot{P}_s(t) \le 0$  (using Lemma 4.1).

Assume that there exists  $t_1$  such that  $\max_s P_s(t_1) > P_{max}$ , there also exists  $t_0$  such that  $\max_s P_s(t_0) = P_{max}$  and  $\max_s P_s(t) > P_{max}$ ,  $t \in [t_0, t_1]$ . Applying Lemma 4.1 we obtain  $P_{max} < \max_s P_s(t_1) \le P_{max}$  which is impossible. Hence  $\sup_{t \in [0,\delta)} P_s(t) < +\infty$ .

Lower bound

We write  $P_{max} = \sup_{t \in [0,\delta)} \max_{s} P_s(t)$ . Assume that  $P_s(t) = \min_{s} P_s(t) < P_{min}$ ,

then:

$$\rho_1(P(t)) \ge \rho_1(P_1(0), P_{max}, \cdots, P_{max}),$$
(4.50)

and

$$\rho_s(P(t)) \le \rho_s(P_1(0), P_{min}, \cdots, P_{min}).$$
(4.51)

Since:

$$\rho_s(P_1(0), P_{min}, \cdots, P_{min}) \xrightarrow[P_{min} \to +\infty]{} 0, \qquad (4.52)$$

there exists a value of  $P_{min}$  such that if  $P_s(t) = \min_s P_s(t) < P_{min}$  then  $\dot{P}_s(t) \geq 0$ . Using Lemma 4.1 and the same argument as above, we obtain that  $\inf_{t \in [0,\delta)} P_s(t) > 0$ .

Maximality

Since  $0 < \inf_{t \in [0,\delta)} P_s(t) \leq \sup_{t \in [0,\delta)} P_s(t) < +\infty$ , and assuming that  $\delta < +\infty$  the considered local solution can be extended to  $[0,\delta')$  with  $\delta < \delta'$ . This proves that the ODE has a unique solution defined on  $\mathbb{R}^+$  and that  $0 < \inf_{t \in \mathbb{R}^+} P_s(t) \leq \sup_{t \in \mathbb{R}^+} P_s(t) < +\infty$ .

(*ii*) Since  $t \to P(t)$  is absolutely continuous, and  $P \to \rho_s(P)$  is Lipschitz continuous,  $t \to \rho_s(P(t))$  is absolutely continuous and has a derivative a.e, and we write Z the set on which the function is non-differentiable.

Let  $t_0 \notin Z$ , and  $s \in \{ \arg \max_{o} \rho_s(P(t_0)) \}$ ,

$$\frac{d}{dt}\frac{P_{s'}(t)}{P_s(t)}|_{t=t_0} = \frac{P_{s'}(t_0)}{P_s(t_0)}[\rho_s(P(t_0)) - \rho_{s'}(P(t_0))] \ge 0,$$
(4.53)

with equality if  $s' \in \{\arg \max \rho_s(P(t_0))\}$ .

Using homogeneity of  ${}^{s}P \to \rho_{s}(P)$ :

$$\rho_s(P(t_0 + \epsilon)) = \rho_s\left(\frac{P(t_0 + \epsilon)}{P_s(t_0 + \epsilon)}\right).$$
(4.54)

Using Lipschitz continuity of  $\rho_s$ :

$$\rho_s \left( \frac{P(t_0 + \epsilon)}{P_s(t_0 + \epsilon)} \right)$$

$$= \rho_s \left( \frac{P(t_0)}{P_s(t_0)} (1 + \epsilon [\rho_s(P(t_0)) - \rho(P(t_0))]) \right) + o(\epsilon)$$

$$\leq \rho_s \left( \frac{P(t_0)}{P_s(t_0)} \right) + o(\epsilon) = \rho_s(P(t_0)) + o(\epsilon)$$
(4.55)

Hence :

$$\lim_{\epsilon \to 0} \frac{\rho_s(P(t_0 + \epsilon)) - \rho_s(P(t_0))}{\epsilon} \le 0.$$
(4.56)

It is noted that the limit exists because of differentiability at  $t_0$ . Define the Lyapunov function  $V(P) = \max_s \rho_s(P) - \min_s \rho_s(P)$ . By the reasoning

#### 4.4. PROOFS

above we have proven that  $t \to \max_{s} \rho_s(P(t))$  is non-increasing and similarly that  $t \to \min_{s} \rho_s(P(t))$  is non-decreasing. This proves that  $t \to V(P(t))$  is non-increasing.

It remains to show that  $t \to V(P(t))$  is strictly decreasing when it is not equal to 0. We consider two sub-cases depending on the number of BSs whose load equals the maximal load max  $\rho_s(P)$ .

$$|\{\arg\max_{s}\rho_{s}(t_{0})\}|=1$$

Consider  $s_0 = \{ \arg\max_s \rho_s(t_0) \}$  and  $t_0 < t_1$ . For  $t_1$  sufficiently close to  $t_0$ , we have that  $\frac{P_{s_0}(t)}{P_s(t)}$  is strictly decreasing on  $[t_0, t_1]$  for  $s \neq s_0$ . Hence  $t \to \max_s \rho_s(P(t)) = \rho_{s_0}(P(t))$  is strictly decreasing on  $[t_0, t_1]$ . Since we have already proven that  $t \to \min_s \rho_s(P(t))$  is non-decreasing, we have that  $t \to V(P(t))$  is strictly decreasing on  $[t_0, t_1]$ .

 $|\{\arg\max_{s}\rho_s(t_0)\}| > 1$ 

Now consider the situation  $|\{\arg\max_s \rho_s(t_0)\}| = n > 1$ . There are two possibilities: either there exists  $t_1 > t_0$  such that  $\{\arg\max_s \rho_s(t)\} = \{\arg\max_s \rho_s(t_0)\}, t \in [t_0, t_1]$  or  $|\{\arg\max_s \rho_s(t)\}| < n$  for all  $t \in [t_0, t_1]$  with  $t_1$  in a sufficiently small neighborhood of  $t_0$ .

Consider the first case. We must have that  $t \to \max_{s} \rho_s(t)$  is strictly decreasing on  $[t_0, t_1]$ , since  $t \to \frac{P_{s_0}(t)}{P_{s_1}(t)}$  is strictly decreasing on  $[t_0, t_1]$  for  $s_0 \in \{\arg\max_s \rho_s(t_0)\}$  and  $s_1 \notin \{\arg\max_s \rho_s(t_0)\}$ . Since  $t \to \min_s \rho_s(P(t))$  is non-decreasing, we have that  $t \to V(P(t))$  is strictly decreasing on  $[t_0, t_1]$ .

The second case is proven by recurrence on n.

We have proven that  $t \to V(P(t))$  is strictly decreasing whenever V(P(t)) > 0 which concludes the demonstration.

## 

### 4.4.5 Proof of theorem 4.5

*Proof.* Theorem 4.2 states that:

$$E\left[\rho_s[k]\right] = \rho_s(P[k]) , \ \sup_k E\left[\rho_s[k]^2\right] < +\infty.$$
(4.57)

Also  $\{\rho_s(P[k]) - E[\rho_s[k]]\}_k$  is a martingale difference noise sequence. We know that  $\mathcal{L}$  is an asymptotically stable attractor of the ODE (4.8). The update equation for powers is a stochastic approximation scheme and applying theorem 2.7 and theorem 2.8 proves the result.

# Chapter 5 Relay networks

In this chapter we address SON for traffic management in relay-enhanced cellular networks. A Relay Station (RS) is a node which is connected to the BS through a wireless link. Communication between the BS and a user involves two hops: the BS transmits data to the RS via air interface, the RS processes it and transmits it to the user, via air interface once again. The benefit of adding RSs to a wireless network is that users are closer to the RS than to the BS, and the power of the signal received by the users is stronger. We use the term "station" to refer to a BS or a RS indifferently.

There is no wired link between the BS and the RS, so deploying a relay is much less costly than deploying a BS. There is a price to pay though, since the available spectrum for wireless communication must be shared between the BS to RS links and the stations to users links. RSs create additional *inter-cell interference* in the network, so that a user located at the frontier between two RSs receives high interference and experiences low data rate. There is a capacity trade-off between gains from increasing the strength of the received signal, and the losses due to resource sharing between the BS to RS links and the stations to users links and increased inter-cell interference.

Future wireless networks such as LTE-Advanced networks are expected to feature RSs. RSs are part of the concept of HetNet. HetNets comprise a high number of low power nodes deployed in high traffic areas to increase capacity, namely pico-cells, femto-cells and RSs. Autonomous management of HetNets is an important research topic because of the sharp increase of the number of nodes. Some HetNet nodes such as femto-cells are deployed directly by subscribers and must be configured without the help of a skilled network engineer. The SON approach seems particularly adapted to HetNets.

As in chapter 4, we take into account *flow-level dynamics* where users enter the network and leave dynamically. We are concerned with three topics:

The capacity gains of relays at the flow level.

- Algorithms to optimize the relays transmit power and the resource allocation between BS to RS and RSs to users links based on traffic measurements.
- Dynamic resource sharing between BS to RS and RSs to users links, i.e taking into account the number of active users and their location in the network.

We provide simple closed-form formulas for dimensioning and evaluating the capacity gains at the flow level using a queuing analysis. An algorithm to optimize the relay transmit powers and resource sharing simultaneously is provided, and its convergence is proven using *stochastic approximation* theorems. The proposed algorithm is partly based on the load balancing algorithm proposed in chapter 4. We extend the results of chapter 4 to more general arrival processes than Poisson arrivals. Our results apply when there is a long-range dependency between the arrivals, for instance for *Markov-modulated Poisson* arrivals. Dynamic resource sharing is modeled as a MDP. For a small number of RSs, the optimal controller is found using *value iteration*. The structure of this optimal controller is to be used as expert knowledge. A set of parameterized policies (the expert knowledge) is introduced. Finally, we use policy gradient reinforcement learning to tune the policy parameter without knowledge about the traffic dynamics.

This chapter is based on our contributions [25], [27]. The proofs for all the results can be found in section 5.5.

# 5.1 Dimensioning

## 5.1.1 System model

We consider a wireless network in downlink. Users arrive at random times and locations, to receive a file of random size  $\sigma$ , with  $E[\sigma] < +\infty$ . We assume that there is no user mobility and that users leave the network upon service completion. We denote by  $\mathbb{A} \subset \mathbb{R}^2$  the network area which we assume to be bounded and convex. A contains a BS (alternatively denoted as macrocell) and several RSs. We denote by  $N_R$  the number of RSs, and we use the convention that station 0 is the BS and station s,  $1 \leq s \leq N_R$  is the *s*-th RS.

We use the terminology of point processes introduced in section 2.1. We denote by  $\{T_k, r_k, \sigma_k\}_{k \in \mathbb{Z}}$  the users' instants of arrival, their location and their file size. For  $B \subset \mathbb{R} \times \mathbb{A}$  a Borel set, we define the number of users who

#### 5.1. DIMENSIONING

arrive in B:

$$N(B) = \sum_{k \in \mathbb{Z}} \mathbf{1}_B(T_k, r_k), \tag{5.1}$$

and the *first-order measure* of the arrival process m:

$$m(B) = E[N(B)].$$
(5.2)

We define the *filtration*  $\mathcal{F}_t$  as the  $\sigma$ -algebra generated by:

$$(N(B): B \subset (-\infty, t) \times \mathbb{A} \text{ Borel set}), \qquad (5.3)$$

which represents the available information when observing the arrival process up to time t. To ease the notation, we define  $\xi_t \in \Xi$  the *effective memory* of the arrival process, with  $\Xi$  a compact metric space, so that  $E[.|\mathcal{F}_t] = E[.|\xi_t]$ . Informally,  $\xi_t$  contains all the  $\mathcal{F}_t$ -measurable random variables which are needed to compute the distribution of the number of arrivals after t. Those variables represent the information available at time t which is relevant to the law of arrivals after t.

Finally, we define the *conditional first-order measure* of the arrival process at time t by:

$$m(B|\xi_t) = E[N(B)|\xi_t].$$
(5.4)

We use three sets of assumptions for the arrival process:

Assumption 5.1 (stationary ergodic traffic). The arrival process satisfies:

- Time-stationary: for  $t \in \mathbb{R}$ ,  $\{T_k - t, r_k, \sigma_k\}_{k \in \mathbb{Z}} \stackrel{d}{=} \{T_k, r_k, \sigma_k\}_{k \in \mathbb{Z}}$
- Independence between arrivals and file sizes  $\{T_k, r_k\}_{k \in \mathbb{Z}} \perp \{\sigma_k\}_{k \in \mathbb{Z}}$
- Ergodicity: the transformation  $\{T_k, r_k, \sigma_k\}_{k \in \mathbb{Z}} \mapsto \{T_k - t, r_k, \sigma_k\}_{k \in \mathbb{Z}}$  is ergodic
- Continuity with respect to Lebesgue measure in space:  $m(dr \times dt) = \lambda(r)dr \times dt$ .
- $\ \textit{Bounded intensity: } \sup_{r \in \mathbb{A}} \lambda(r) < +\infty$

Assumption 5.2 (stationary ergodic light traffic). The arrival process satisfies assumptions 5.1 and:

- Finite second-moment measure: for  $T \ge 0$ ,  $E[N([0,T] \times \mathbb{A})^2] < +\infty$ 

- Conditional continuity with respect to Lebesgue measure in space:  $\exists \overline{\lambda}, m(dr \times [0,T)|\xi_0) = \overline{\lambda}(r,[0,T),\xi_0) dr.$
- Bounded conditional intensity:  $\sup_{\xi_0 \in \Xi} \sup_{r \in \mathbb{A}} \overline{\lambda}(r, [0, T), \xi_0) < +\infty$

**Assumption 5.3** (Poisson light traffic). The arrival process satisfies assumptions 5.2 and is a Poisson process:

- N(B) is a Poisson random variable with mean m(B)
- $(N(B_1), \ldots, N(B_N))$  are independent if  $\bigcap_{n=1}^N B_n = \emptyset$ .

It is noted that assumptions 5.1 are the most general, allowing for *correlated arrivals* in both time and space, while 5.3 are the most restrictive. A special case of assumptions 5.2 is *Markov-modulated Poisson* arrivals. For Markov modulated Poisson arrivals,  $t \to \xi_t$  is a Markov process, and its evolution does not depend on the arrival process. Conditional to  $\{\xi_t\}_{t\in\mathbb{R}}$ , the arrival process is a Poisson process. It is also noted that we do not assume that  $\sigma$  has finite variance so that our results hold for *heavy-tailed traffic*.

As mentioned earlier, RSs have no direct link to the core network, and are connected to the BS by a wireless link. This wireless link uses the same radio resources as the station to users' links and we are interested in finding an appropriate resource sharing method. This mechanism is often called in-band relaying. Depending on the multi-access radio technology, the radio resources can refer to codes in CDMA, to time slots in Time Division Multiple Access (TDMA) or to time-frequency blocks in OFDMA. We ignore the granularity of resources and we denote by  $x \in [0, 1]$  the proportion of resources allocated to the link between the BS and RSs. We further assume that RR scheduling applies in all links: the link between the BS and RSs is shared in a PS way among the RSs, and that each link between a station and the users it serves is shared in a PS way among those users.

### 5.1.2 System capacity

Let  $\mathbb{A}_s \subset \mathbb{A}$  denote the area covered by station s. For a given  $x \in [0, 1]$ we calculate the capacity of the system, and the optimal resource sharing strategy  $x^*$  which ensures *stability* whenever it is possible. We assume until the end of this section that the traffic is uniform  $m(dr \times dt) = \lambda_0 dr \times dt$ . Namely, we denote by C the capacity of the system defined as the maximal value of  $\lambda_0 E[\sigma]$  that keeps the system stable i.e the number of users in the system does not grow to infinity. We write  $R_{rel,s}$ ,  $1 \leq s \leq N_R$  the data rate

#### 5.1. DIMENSIONING

of the link between BS and RS s when it is the only active link, and  $R_s(r)$ ,  $r \in A_s$  the data rate between station s and a user located at r when he is the only user served by station s. The effect of *inter-cell interference* is incorporated in  $R_{rel,s}$  and  $R_s(r)$ , hence the results given here hold regardless of the amount of inter-cell interference.

**Theorem 5.1.** The capacity C of the system is:

$$C(x) = \min\left(C_{rel}(x), \min_{0 \le s \le N_R} C_s(x)\right), \qquad (5.5)$$

with:

$$C_{rel}(x) = x \left( \sum_{s=1}^{N_R} \frac{|\mathbb{A}_s|}{R_{rel,s}} \right)^{-1}, \qquad (5.6)$$

$$C_s(x) = (1-x) \left( \int_{\mathbb{A}_s} \frac{1}{R_s(r)} dr \right)^{-1}.$$
 (5.7)

Furthermore, there exists a unique  $x^* \in [0, 1]$  which maximizes the capacity,

$$x^{*} = \frac{\left(\max_{0 \le s \le N_{R}} \int_{\mathbb{A}_{s}} \frac{1}{R_{s}(r)} dr\right)^{-1}}{\left(\max_{0 \le s \le N_{R}} \int_{\mathbb{A}_{s}} \frac{1}{R_{s}(r)} dr\right)^{-1} + \left(\sum_{s=1}^{N_{R}} \frac{|\mathbb{A}_{s}|}{R_{rel,s}}\right)^{-1}},$$
(5.8)

with  $C(x^*)$  the maximal capacity.

It is noted that this result applies regardless of the underlying packet dynamics. More precisely, consider two scenarios:

- 1. Small files: When a user served by a RS arrives in the network, the file he wants to receive enters the BS to RSs link and once the whole file has gone through that link, it enters the corresponding RS to user link and is transmitted. This model is reasonable for small files.
- 2. Larger files: In a more realistic setting, when a user served by a RS arrives in the network, the file he wants to receive arrives as small packets which enter the BS to RSs link, possibly with delays between packets. Once a packet has gone through the BS to RSs link it immediately enters the RS to user link. Here the file can be "split" between the two successive links.

For both traffic models the demonstration remains the same, and the system capacity does not change.

# 5.1.3 Relay gain

We introduce the concept of RS placement gain, and give a method to evaluate the resulting capacity improvement. We use the propagation model described in section 7.1. We assume that the signal attenuation per distance unit is smaller for the useful signal between the BS and RSs than for interfering signals. This can be achieved by placing RSs high enough so that the propagation between the BS and RSs is close to the line-of-sight case, while taking advantage of buildings to increase the attenuation of interfering signals. The path loss parameters are  $(A, \eta_r)$  for the useful signal between the BS and RSs, and  $(A, \eta)$  for all other signals with  $2 \leq \eta_r \leq \eta$ . The case  $\eta_r = 2$  corresponds to line-of-sight propagation between BS and RSs. We call  $\eta - \eta_r$  the relay gain, and  $\eta_r = 2$  gives an upper bound on the achievable capacity by intelligent relay placement.

# 5.1.4 Numerical experiments

We evaluate the influence of the system parameters on the performance. The model parameters are given in Table 5.1, and Figure 5.1 represents the network layout. Interference from neighbouring cells is taken into account. Data rates  $R_s(r)$  are calculated for single-tap Rayleigh fading as explained in chapter 3. We choose a large cell radius since [52] had shown that relays are only beneficial in such a setting.

Model parameters	
Cell layout	Hexagonal
Antenna type	Omni-directional
Cell Radius	2km
Access technology	OFDMA
Fast-fading model	Rayleigh
$N_{RB}$	10
Resource block size	180kHz
BS transmit power	46 dBm
RS maximum transmit power	30 dBm
Thermal noise	-174 dBm/Hz
Path loss model	$128 + 37.6 \log_{10}(d) \text{ dB}, d \text{ in km}$
File size	10 Mbytes

Table 5.1: Model parameters



Figure 5.1: Relay placement

Figure 5.2 and 5.3 show the capacity of the system and the optimal relay transmit power respectively as the number of relays grows, with and without relay gain. The optimal relay transmit powers are determined using an exhaustive search for a discrete set of possible values (  $\{-10, \ldots, 60\}$  dBm ), all relays having the same transmit power. The case without relay gain is denoted "bad planning" (with  $\eta_r = \eta = 3.5$ ) and with relay gain "good planning" (with  $\eta_r = 2$  and  $\eta = 3.5$ ). It is noted that the value of the optimal relay transmit power in the "bad planning" case is 0mW for all number of relays (below the x-axis). It demonstrates that the impact of relay gain is fundamental since without relay gain it is actually detrimental to deploy relays. With relay gain however, the system capacity increases sharply.

Figure 5.4 shows the impact of the relay gain on the system capacity for a fixed number of relays (15 in this case), and we can see that the capacity increases almost linearly in the relay gain. This can be explained by the fact that  $\log_2(1+S||r||^{\eta-\eta_r})$  is close to  $\log_2(S) + (\eta - \eta_r) \log_2(||r||)$  when  $S||r||^{\eta-\eta_r}$ is large. It shows that if one is able to evaluate the relay gain prior to deployment (by measuring the value of the path loss exponent in candidate sites for relay placement), one can actually determine if relay deployment is beneficial and the expected benefit. Furthermore the point where the two curves intersect represents the minimal relay gain needed for any benefit from relay deployment to appear.



Figure 5.2: System capacity as a function of the number of relays, for different planning strategies

# 5.2 Self-Optimization

We have given a procedure for network dimensioning and we show that the network can adapt itself to traffic variations based solely on measurements and perform automatic load balancing. Two critical parameters are tuned: the pilot powers of the RSs which control the zones served by the RSs and the resources allocated to the backhaul links. Both parameters are updated simultaneously, and we show that the proposed mechanism ensures their coordination. We extend the load balancing mechanism of chapter 4 to relay enhanced networks, and we tune the transmitted pilot powers and the resource allocation to the backhaul to converge to an optimal configuration. Unlike the previous section, we consider a slightly more general model: the resources allocated to the backhaul links are not shared in a PS manner any more.

Instead of sharing  $\sum_{s=1}^{N_R} x_s$  resources among the backhaul links in a PS manner, for each s, a quantity  $x_s$  is allocated to the link between the BS and RS s, which does not require a scheduler to share the resources among the different backhaul links. If PS applies for the backhaul links then, the quantity allocated to the backhaul is simply  $\sum_{s=1}^{N_R} x_s$ .



Figure 5.3: Optimal relay transmit power as a function of the number of relays, for different planning strategies

# 5.2.1 Traffic estimation

In appendix 5.5.2, we show that quantities of interest can be estimated by traffic measurements. We do not assume the traffic to be uniform. We write  $\overline{\rho_s}$  the load of station s and  $\overline{\rho}_{rel,s}$  the load of the backhaul between the BS and RS s, which can be expressed as:

$$\overline{\rho}_{s} = \frac{E\left[\sigma\right]}{1 - \sum_{s'=1}^{N_{R}} x_{s'}} \int_{\mathbb{A}_{s}} \frac{\lambda(r)}{R_{s}(r)} dr \quad , \quad \overline{\rho}_{rel,s} = \frac{E\left[\sigma\right] \int_{\mathbb{A}_{s}} \lambda(r) dr}{x_{s} R_{rel,s}}.$$
(5.9)

Define  $\rho_s$  and  $\rho_{rel,s}$  by :

$$\rho_s = \int_{\mathbb{A}_s} \frac{\lambda(r)}{R_s(r)} dr \quad , \quad \rho_{rel,s} = \frac{\int_{\mathbb{A}_s} \lambda(r) dr}{R_{rel,s}}. \tag{5.10}$$

then the loads can be expressed in the reduced form:

$$\overline{\rho}_{s} = \frac{E\left[\sigma\right]\rho_{s}}{1 - \sum_{s'=1}^{N_{R}} x_{s'}} \quad , \quad \overline{\rho}_{rel,s} = \frac{E\left[\sigma\right]\rho_{rel,s}}{x_{s}}.$$
(5.11)

The condition for load balancing is  $\overline{\rho}_{rel,s} = \overline{\rho}_s = \overline{\rho}_0$ , which reduces to:

$$\frac{\rho_{rel,s}}{x_s} = \frac{\rho_s}{1 - \sum_{s'=1}^{N_R} x_{s'}} = \frac{\rho_0}{1 - \sum_{s'=1}^{N_R} x_{s'}}.$$
(5.12)



Figure 5.4: Impact of the relay gain on the system capacity

The mean flow size  $E[\sigma]$  has disappeared, so that load balancing can be achieved without estimating it.

Time is slotted, with T the time slot size. The n-th time slot is [nT, (n + 1)T). We write  $\xi[n] = \xi_{Tn}$ . Loads can be estimated using theorem 5.4 given in section 5.5. The loads are estimated by:

$$\rho_s[n] = \frac{1}{T} \sum_{k \in \mathbb{Z}} \frac{1}{R_s(r_k)} \mathbf{1}_{\mathbb{A}_s}(r_k) \mathbf{1}_{[nT,(n+1)T)}(T_k),$$
(5.13)

$$\rho_{rel,s}[n] = \frac{1}{T} \sum_{k \in \mathbb{Z}} \frac{1}{R_{rel,s}} \mathbf{1}_{\mathbb{A}_s}(r_k) \mathbf{1}_{[nT,(n+1)T)}(T_k)$$
(5.14)

Assumption 5.4. Data rates are lower bounded:  $\inf_{r \in \mathbb{A}} \min_{s} R_s(r) = R_{min} > 0$ 

We recall that  $P \mapsto |\mathbb{A}_s(P)| = \int_{\mathbb{A}_s(P)} dr$  is Lipschitz continuous as a particular case of theorem 4.3. Hence  $P \mapsto \rho_{rel,s}(P)$  and  $P \mapsto \rho_{rel,s}(P)$  are both Lipschitz continuous.

**Property 5.1.**  $P \to |\mathbb{A}_s(P)|$  is Lipschitz continuous on  $\mathcal{P} = [P_{min}, P_{max}]^{N_R+1}$  with  $0 < P_{min} \leq P_{max} < +\infty$ .

Equation (5.4) is valid as long as there is an *admission control* rule on the minimal data rate for a user to enter the system. Theorem 5.4 states

that the load estimates are unbiased:

$$E[\rho_s[n]] = \rho_s , \quad E[\rho_{rel,s}[n]] = \rho_{rel,s}.$$
 (5.15)

# 5.2.2 Traffic balancing for the backhaul

First assume that the RSs transmit powers are fixed, so that the zones they serve do not change. We want to balance the traffic based on measurements, starting from an arbitrary allocation. If  $A_s$  has Lebesgue measure 0 we can simply ignore RS s, hence we assume, without loss of generality, that  $\min \rho_s > 0$  and  $\min \rho_{rel,s} > 0$ .

**Proposition 5.1.** (i) The unique solution (5.12) is  $x^*(\rho)$ :

$$x_{s}^{*}(\rho) = \frac{\frac{\rho_{rel,s}}{\rho_{s}}}{1 + \sum_{s'=1}^{N_{R}} \frac{\rho_{rel,s'}}{\rho_{s'}}}$$
(5.16)

- (ii) We have that  $0 < \sum_{s'=1}^{N_R} x_{s'}(\rho) < 1$
- (iii)  $\rho \to x^*(\rho)$  is locally Lipschitz continuous

*Proof.* (i) is proven by noticing that for any solution we must have that

$$s \to \frac{x_s \rho_s}{\rho_{rel,s}} = 1 - \sum_{s'=1}^{N_R} x_{s'}$$
 (5.17)

is constant. (ii) is straightforward since  $0 < \frac{\rho_{rel,s}}{\rho_s} < +\infty$  and equation (5.16). (iii) Is true since we have assumed  $\rho_s > 0$ .

Write  $x_s[n]$  the proportion of resources allocated to the link between the BS and RS s during the *n*-th time slot, and  $\epsilon_n > 0$  a step size. We consider two types of steps sizes:

- (constant step sizes)  $\epsilon_n = \epsilon > 0$
- (decreasing step sizes)  $\epsilon_n = \frac{1}{n^{\gamma}}$  with  $\gamma_0 < \gamma \leq 1$ .

We define H the *admissible set* which is convex:

$$H = \{ x : x_s \ge 0 , \ 0 \le \sum_{s=1}^{N_R} x_s \le 1 \}.$$
(5.18)

We write  $[.]_{H}^{+}$  the projection on H. We consider the following iterative scheme for load balancing:

$$x_s[n+1] = [x_s[n] + \epsilon_n g_s(\rho[n], x[n])]_H^+, \qquad (5.19)$$

$$g_s(\rho, x) = \rho_{rel,s} (1 - \sum_{s'=1}^{N_R} x_{s'}) - \rho_s x_s.$$
 (5.20)

The convergence to the unique optimal point is given by the following theorem. The proof is based on *stochastic approximation*: we associate an ODE to the iterative scheme and study its asymptotic behaviour. We then prove that the iterates converge to Lyapunov stable attractors of the ODE.

**Theorem 5.2.** With assumptions 5.2 and 5.5, the sequence  $\{x[n]\}_n$  converges to  $x^*(\rho)$ . The convergence occurs a.s for decreasing step sizes, and in distribution for constant step sizes with  $\epsilon \to 0^+$ .

# 5.2.3 Coordination between backhaul and cell sizes

We assume that both the resource allocation to the backhaul, and the zones served by the relays are adapted simultaneously, and we propose a coordination mechanism. The idea is to make the two mechanisms operate on a "different time scale", namely, the backhaul adaptation is sufficiently fast compared to the cell sizes so that it appears as *quasi-static*. Relevant twotime scales stochastic approximation results are used to prove convergence.

We assume that users attach themselves to the station with the strongest received pilot power. Let  $P_s$  denote the power of the pilot signal transmitted by station s and  $L(r_s, r)$  the signal attenuation between station s and location  $r \in \mathbb{A}$ , the zones covered by stations can be written:

$$\mathbb{A}_{s}(P) = \{r : s \in \operatorname*{arg\,max}_{s'} P_{s'}L(r_{s'}, r)\}.$$
(5.21)

We write  $P_s[n]$  the power of the pilot signal transmitted by station s during the *n*-th time slot. Let  $\delta_n > 0$  denote another step size sequence. As previously, we distinguish two cases:

- (constant step sizes)  $\epsilon_n = \epsilon > 0$ ,  $\delta_n = \delta(\epsilon) > 0$ , with  $\frac{\delta(\epsilon)}{\epsilon} \xrightarrow[\epsilon \to 0^+]{} 0$ 

- (decreasing step sizes) 
$$\epsilon_n = \frac{1}{n^{\gamma_1}}, \, \delta_n = \frac{1}{n^{\gamma_2}}, \, \text{with } \gamma_0 < \gamma_1 < \gamma_2 \le 1$$

We consider the constraint set for the pilot powers  $\mathcal{P} = [P_{min}, P_{max}]^{N_R+1}$ with  $0 < P_{min} \leq P_{max} < +\infty$ . The update equations are:

$$x_s[n+1] = [x_s[n] + \epsilon_n g_s(\rho[n], x[n])]_H^+$$
(5.22)

$$P_s[n+1] = [P_s[n] + \delta_n h_s(\rho[n], P[n])]_{\mathcal{P}}^+, \qquad (5.23)$$

$$h_s(\rho, P) = P_s(\rho_0(P) - \rho_s(P)).$$
(5.24)

The convergence to a network configuration where the loads of all links are equal is given by the next result.

**Theorem 5.3.** With assumptions 5.2 and 5.5, the sequence  $\{(x[n], P[n])\}_n$ converges to a set on which the loads of all links are equal, for  $P_{min}$  sufficiently small and  $P_{max}$  sufficiently large. As in the previous theorem, the convergence occurs a.s for decreasing step sizes, and in distribution for constant step sizes with  $\epsilon \to 0^+$ .

# 5.2.4 Numerical experiments

We show some numerical experiments to assess the efficiency of the proposed method. We have proven mathematically that, for a given stationary traffic, the proposed algorithms converge to the optimal configuration. However, in practical situations, the traffic changes over the course of a day, with traffic peaks and periods during which the served traffic is low, for example during the night.

Our numerical experiments show that when the traffic is not stationary, the algorithm is able to adapt itself and successfully *track* the changing traffic patterns. One BS and 4 RSs are considered. To demonstrate the tracking properties, we adopt the following traffic configuration: a uniform traffic of 50 Mbps which does not change during time, and a *hot-spot* i.e a limited zone with high traffic, located next to RS 1. The hot-spot traffic varies between 0 Mbps and 30 Mbps, and the time interval between the maximal traffic and minimal traffic is 2 hours. We show that the algorithm adapts both cell sizes and backhaul resources allocation in order to handle the variation in the traffic pattern. We compare the proposed algorithm with a reference scenario in which the network parameters are static. The network parameters are the optimal static parameters for the period in which the hot-spot traffic is 10 Mbps, the second hour with the highest load. The motivation behind such a model is a scenario in which a network engineer has chosen the optimal network parameters for a uniform traffic, and an unexpected traffic pattern appears for a few hours. Such traffic variations are too fast for a human operator to modify the network parameters accordingly. This situation shows

what kind of gains can be expected from network equipments that can adapt themselves automatically to hourly traffic patterns.

Figure 5.5 illustrates the chosen network setup. Figure 5.6 shows the total served traffic by the network, which is the sum of the uniform traffic (50 Mbps) and the hot-spot traffic (between 0 and 30 Mbps). Figure 5.7 shows the evolution of the pilot power of two relay stations scaled to their total transmitted power as a function of time, when the proposed SON algorithm is used. At low traffic periods, RS 1 transmits at a high power and covers a large area. At high traffic periods RS 1 transmits at low power in order to serve a smaller area and avoid being overloaded since it absorbs most of the hot-spot traffic. Figure 5.8 and 5.9 compare the loads of links between the proposed SON algorithm and the reference scenario. In the reference scenario, the loads of the BS and of RS 1 are unbalanced, and during the high traffic periods RS 1 absorbs too much traffic, its load being close to 100%. This is highly problematic: without admission control, the average file transfer time becomes infinite when the load goes to 100%. With admission control, a load close to 100% results in unacceptably high blocking rate. With the proposed algorithm, the loads of all links are very close to each other, and are lower than in the reference scenario. At high traffic periods, the worst load is 70% which is a large improvement with respect to the reference scenario. This shows that the proposed algorithm successfully balances the loads and reduces congestion by adapting to the changing traffic pattern.

# 5.3 Dynamic resource allocation

In the previous sections, our approach was to adapt the network to the traffic configuration, defined in terms of arrival rates. The aim was to find the best parameters for a given traffic. We turn to a case in which we act on a *faster time scale*, and instead of adapting to the arrival rates, we adapt to the current number and locations of active users. It is indeed a faster time scale since the arrival rates change on the time scale of minutes to hours, whereas the configuration of active users changes on a time scale of seconds. The BS observes the current state of the network and decides whether to activate the BS to RSs links or the stations to users' links.

### 5.3.1 Infinite buffer case: stabilizing policy

We partition each  $\mathbb{A}_s$  into N regions  $\mathbb{A}_{s,i}$ ,  $1 \leq i \leq N$ , each associated with a different radio condition. We call *i*-th *traffic class* in station *s* the users who arrive in  $\mathbb{A}_{s,i}$ . The state of the system can then be described by a vector



Figure 5.5: Hot-spot traffic model

 $\mathbf{S} \in \mathbb{N}^{(2N_R+1)N}$ :

$$\mathbf{S} = ((S_{s,i})_{0 \le s \le N_R, 1 \le i \le N}, (S_{rel,s,i})_{1 \le s \le N_R, 1 \le i \le N}).$$
(5.25)

In the small files framework we count the number of users present in the links, otherwise we count the number of packets. Hence  $S_{s,i}$  is the number of users (packets respectively) of class *i* served by the station to user link in station *s*, and  $S_{rel,s,i}$ ,  $s \ge 1$  - the number of users (packets respectively) of class *i* served by the BS to RS *s* link. We write  $R_{s,i}$  the data rate of a user of class *i* served by station *s*.

We first assume infinite buffer lengths and we want to find the policy that keeps the system stable whenever that is possible. The problem is in fact a particular case of the constrained queuing systems considered by [57]. It has been proven that such a policy exists and that it is a *max-weight* policy. We define the weights:

$$D_s = \max_{1 \le i \le N} (S_{s,i} R_{s,i}), \ 0 \le s \le N_R$$
(5.26)

$$D_{s,rel} = \max_{1 \le i \le N} ((S_{rel,s,i} - S_{s,i})R_{rel,s}), \ 1 \le s \le N_R$$
(5.27)

The max-weight policy is then:

- If 
$$\sum_{1 \le s \le N_R} D_{s,rel} \ge \sum_{0 \le s \le N_R} D_s$$
: activate the BS to RS  $s^*$  link with  $s^* = \underset{1 \le s \le R_S}{\operatorname{max}} D_{s,rel}$ ,



Figure 5.6: Total served traffic as a function of time

- *Else:* activate the stations to users' links, and in each station s serve the class of users  $i_s^* = \underset{i}{\arg \max} n_{s,i} R_{s,i}$ 

# 5.3.2 Finite buffer case: MDP formulation

We assume that the system state **S** is restrained to  $\mathcal{S} \subset \mathbb{N}^{(2N_R+1)N}$  with  $\mathcal{S}$  finite due to admission control mechanisms. We formulate the problem as a CTMDP and optimize QoS metrics such as blocking rate or file transfer time. We formulate the problem in the small files framework since we want to solve the MDP iteratively, in order to keep the state space relatively small. The learning approach of the next section however can handle large state spaces as demonstrated later.

#### State and action spaces

We assume that each link has a maximal number of simultaneous active users.

$$\mathcal{S} = \left\{ \mathbf{S} : S_{rel,s,i} \le \overline{S_{rel,s,i}}, \ 1 \le s \le N_R, \ 1 \le i \le N \right.$$
  
and  $S_{s,i} \le \overline{S_{s,i}}, \ 0 \le s \le N_R, \ 1 \le i \le N \left. \right\}$ 

We define  $\mathcal{A} = \{0, 1\}$  the action space, with the convention:



Figure 5.7: SON algorithm: scaled relay pilot power as a function of time

-a = 0: activate BS to RSs links and share them in a PS manner

-a = 1: activate stations to users' links and share them in a PS manner

### Transition probabilities

Assuming that the file size  $\sigma$  is exponentially distributed, the system is a CTMDP. Transitions from **S** to **S'** given action *a* have the following intensities:

- Arrival of a user from class i in the BS:  $\mathbf{1}_{\mathcal{S}}(\mathbf{s}') \int_{\mathbb{A}_{0,i}} \lambda(r) dr$
- Arrival of a user from class i in the BS to RS s link:  $\mathbf{1}_{\mathcal{S}}(\mathbf{s}') \int_{\mathbb{A}_{s,i}} \lambda(r) dr$
- Departure of a user from class i in station s:  $\mathbf{1}_{\{1\}}(a)\mathbf{1}_{\mathcal{S}}(\mathbf{s}')\frac{R_{s,i}S_{s,i}}{E[\sigma]\sum_{i=1}^{N}S_{s,i}}$
- Movement of a user of class i from BS to RS s link to RS s to users' link:  $\mathbf{1}_{\{0\}}(a)\mathbf{1}_{\mathcal{S}}(\mathbf{s}') \frac{S_{rel,s,i}R_{rel,s}}{E[\sigma]\sum_{i=1}^{N}\sum_{s=1}^{N_R}S_{rel,s,i}}$

### Average reward

We call *policy* a mapping  $S \to \mathcal{D}(\mathcal{A})$ , with  $\mathcal{D}(\mathcal{A})$  the set of probability distributions on  $\mathcal{A}$ . We write  $(\mathbf{S}(t), a(t), r(t))_{t \in \mathbb{R}^+}$  a sample path of the CTMDP



Figure 5.8: Reference scenario: loads as a function of time

with  $\mathbf{S}(t)$  the state, a(t) the action, and r(t) the reward at time t respectively. We are interested in the average reward criterion of a policy P:

$$J_{\mathbf{S}_0}(P) = \lim_{T \to +\infty} \frac{1}{T} E_{P,\mathbf{S}_0} \left[ \int_0^T r(t) \right]$$
(5.28)

with  $E_{P,\mathbf{S}_0}$  the expectation with respect to the probability generated by P, starting at  $\mathbf{S}_0$ , which does not depend on  $\mathbf{S}_0$  if the system is ergodic under policy P.

### Performance criteria

We consider two performance criteria: mean file transfer time and blocking rate(considering admission control). For each performance criterion we can define a corresponding instantaneous reward for each state-action pair, and finding the optimal policy for the resulting MDP yields the best policy with respect to the considered performance criterion.

To optimize the mean file transfer time, we define the reward in state  $\mathbf{S}$  as the number of users divided by the arrival rate

$$\frac{\sum_{i=1}^{N} (S_{0,i} + \sum_{s=1}^{N_R} (S_{s,i} + S_{rel,s,i}))}{\int_{\mathbb{A}} \lambda(r) dr},$$
(5.29)



Figure 5.9: SON algorithm: loads as a function of time

and for any policy P that makes the system ergodic,  $J_{\mathbf{S}_0}(P)$  is the mean file transfer time in the system using Little's law ([41]).

We define the *blocking rate* as the ratio between the mean number of blocked users and the mean number of users accessing the system, once again assuming ergodicity. Given action a, let  $\beta(\mathbf{S}, a)$  the sum of transition intensities out of state  $\mathbf{S}$  and  $b(\mathbf{S}, a)$  the sum of the intensities of arrival or movements which would be blocked, then the reward is defined as  $\frac{b(\mathbf{S},a)}{\beta(\mathbf{S},a)}$ .

### Optimal control and parametrization

Given the previous description, we associate a DTMDP by *uniformization* as described in 2.4.4. We derive the optimal policy using *value iteration* as described in 2.4.1. It is noted that the complexity of finding the optimal policy is exponential in the number of relays, limiting the approach to small problems. In order to preserve scalability, we introduce a well-chosen family of policies. For commodity of notation we use the following indexing of  $\mathbf{S}$ :

$$(S_1, \cdots, S_k, \cdots, S_{(2N_R+1)N}) = ((S_{s,i})_{0 \le s \le N_R, 1 \le i \le N}, (S_{rel,s,i})_{1 \le s \le N_R, 1 \le i \le N}).$$
(5.30)

For  $\theta \in \mathbb{R}^{(2N_R+1)N}$  we write:

$$\langle \mathbf{S}, \theta \rangle = \sum_{k=1}^{(2N_R+1)N} \theta_k S_k.$$
(5.31)

To  $\theta$  we associate the deterministic weighted policy  $P_{d,\theta}$ :

$$P_{d,\theta}(\mathbf{S},1) = \begin{cases} 1 & , \langle \mathbf{S}, \theta \rangle \ge 0\\ 0 & , \langle \mathbf{S}, \theta \rangle < 0 \end{cases}$$
(5.32)

$$P_{d,\theta}(\mathbf{S},0) = 1 - P_{d,\theta}(\mathbf{S},1)$$
(5.33)

It is noted that a deterministic weighted policy is essentially an hyperplane separating the state space in two regions, each half-space corresponding to an action of  $\mathcal{A}$ .

It is also noted that the max-weight policy is a deterministic weighted policy. We then compare the performance of three policies: the optimal policy, the max-weight policy and the optimal deterministic weighted policy. The optimal deterministic weighted policy is well defined since the set of deterministic policies is finite. Figure 5.10 and 5.11 show the file transfer



Figure 5.10: File transfer time as a function of the traffic for different control strategies

time and the blocking rate for the three policies, for one relay, one traffic class



Figure 5.11: Block call rate as a function of the traffic for different control strategies

and a maximum of 10 users for all links. We can see that the max-weight policy is very close to the optimal policy when we are concerned with the block call rate, which is natural since it attempts to ensure stability. In the file transfer time case however, the optimal deterministic weighted policy is noticeably closer to the optimal policy than the max-weight. The fact that max-weight scheduling possibly incurs long delays has been reported in the literature. Hence based on those two results we can conclude that the set of deterministic weighted policies is rich enough to restrain the search to this set, since with a high number of relays and/or traffic classes, finding the optimal policy becomes prohibitively expensive.

# 5.4 Learning

We have demonstrated that the set of weighted policies is rich enough to represent a good trade-off between performance and search complexity. We move on to a *model-free* approach, and we assume no knowledge of the transition intensities and rewards. We are interested in learning the best weighted policy, simply by observing sample paths of the Partially Observable Markov Decision Process (POMDP)  $(\mathbf{S}(t), a(t), r(t))_{t \in \mathbb{N}}$ . The model can be partially observed for various reasons. For example if user arrivals are correlated in time, the evolution of the system after t depends on the user arrivals before t, and this information is not present in  $\mathbf{S}(t)$ . The method presented here is valid without assuming Poisson arrivals or exponentially distributed file sizes.

### 5.4.1 Policy gradient approach

We use the policy gradient approach described in 2.4.3. It is noted that such algorithms work with stochastic policies, for the cost to be differentiable with respect to the policy parameter. We introduce stochastic weighted policy  $P_{s,\theta}$ :

$$P_{s,\theta}(\mathbf{S},0) = 1 - f(\langle \mathbf{S}, \theta \rangle), \qquad (5.34)$$

$$P_{s,\theta}(\mathbf{S},1) = f(\langle \mathbf{S}, \theta \rangle), \qquad (5.35)$$

with  $f(x) = \frac{1}{1+e^{-x}}$ . We are interested in finding the  $\theta$  which minimizes the average cost  $J_{\mathbf{S}_0}(P_{s,\theta})$ . The link with the policies introduced in the previous section is that any deterministic weighted policy  $P_{d,\theta}$  can be approximated arbitrarily well by a stochastic weighted policy  $P_{s,K\frac{\theta}{\|\theta\|}}$ , with  $K \in \mathbb{R}^+$  arbitrarily large.

# 5.4.2 Convergence to a local optimum

We show how to converge to a local optimum of the average cost. We differentiate the action probabilities:

$$\frac{\partial \log(P_{s,\theta}(\mathbf{S},0))}{\partial \theta_k} = -f(\langle \mathbf{S}, \theta \rangle)S_k = -P_{s,\theta}(\mathbf{S},1)S_k$$
(5.36)

$$\frac{\partial \log(P_{s,\theta}(\mathbf{S},1))}{\partial \theta_k} = (1 - f(\langle \mathbf{S}, \theta \rangle))S_k = P_{s,\theta}(\mathbf{S},0)S_k$$
(5.37)

All *stochastic policies* guarantee ergodicity of the system if we are considering a MDP, as stated by the next result.

**Proposition 5.2.** If we are considering a MDP model (not a POMDP), for every  $\theta$ , the Markov chain  $\{\mathbf{S}(t)\}$  generated by policy  $P_{s,\theta}$  is ergodic, implying that  $J_{\mathbf{S}_0}(P_{s,\theta})$  is well-defined and does not depend on  $\mathbf{S}_0$ .

*Proof.* Consider an arbitrary state  $\mathbf{S}$  and the state  $\mathbf{0}$ . There exists a path with strictly positive probability between  $\mathbf{0}$  and  $\mathbf{S}$  since arrivals do not depend on the actions. There exists a path of strictly positive probability between

#### 5.4. LEARNING

**S** and **0** as well since in every state in which at least one user (packet) is present in the system, there is a transition corresponding to the departure of a user (or a packet) with strictly positive probability. It is the case because for any policy and any state there is a strictly positive probability for each action to be selected. This proves that the chain is irreducible.

Furthermore, the chain is aperiodic since there exists a transition from state  $\mathbf{0}$  to itself. This transition exists because we have applied *uniformization*.

Since the state space is finite, and the chain is both irreducible and aperiodic, this proves ergodicity of the chain for any policy.  $\hfill \Box$ 

Using the fact that  $0 < P_{s,\theta}(\mathbf{S}, a) < 1$ ,  $a \in \{0, 1\}, \mathbf{S} \in \mathcal{S}$  we have that:

$$-\max_{a \in \{0,1\}} \max_{\mathbf{S} \in \mathcal{S}} \left| \frac{\partial \log(P_{s,\theta}(\mathbf{S},0))}{\partial \theta_k} \right| < +\infty, \ 1 \le k \le (2N_R + 1)N$$

-  $\max_{a \in \{0,1\}} \max_{\mathbf{S} \in S} r(\mathbf{S}, a) < +\infty$ , with  $r(\mathbf{S}, a)$  the reward given state **S** and action a

Given  $\beta \in (0, 1)$ , and a sample path of the POMDP  $(\mathbf{S}(t), a(t), r(t))_{t \in \mathbb{N}}$ , we define the sequence of gradient estimates and the eligibility traces  $(\Delta(t), z(t))_{t \in \mathbb{N}}$  by the following recursive equation:

$$z(0) = 0 , \ \Delta(0) = 0 \tag{5.38}$$

$$z(t+1) = \beta z(t) + \nabla_{\theta} \log(P_{s,\theta}(\mathbf{S}(t), a(t)))$$
(5.39)

$$\Delta(t+1) = \Delta(t) + \frac{1}{t+1} [r(t)z(t) - \Delta(t)]$$
(5.40)

We denote by  $\Delta(t)(\theta)$  the value of the gradient estimate  $\Delta(t)$  when parameter  $\theta$  is used to highlight the dependence on  $\theta$ . Theorem 2.11 states that for  $\beta$  large enough:

$$\liminf_{t \to +\infty} \left\langle E\left[\Delta(t)(\theta)\right], \, \nabla_{\theta} J(\theta) \right\rangle > 0.$$
(5.41)

Namely,  $-\Delta(t)$  is a (noisy) descent direction for large t. We can use  $\Delta(t)(\theta)$  for optimizing  $J(\theta)$  using stochastic approximation, see section 2.3. We consider  $\Theta \subset \mathbb{R}^{(2N_R+1)N}$  a compact and convex set,  $[.]_{\Theta}^+$  the projection on  $\Theta$ ,  $(\epsilon_n)_{n\in\mathbb{N}}$  a sequence of positive step sizes satisfying the usual stochastic approximation conditions. We define  $\theta_n$  by:

$$\theta_0 \in \Theta \tag{5.42}$$

$$\theta_{n+1} = [\theta_n - \epsilon_n \Delta(t)(\theta_n)]_{\Theta}^+ \tag{5.43}$$

then  $\theta_n$  converges to a local minimum of J in  $\Theta$ . The convergence point is not necessarily unique if J or  $\Theta$  are not convex.

Furthermore, since  $-\Delta(t)(\theta)$  is a descent direction with high probability for large t, we have that the performance of the system improves almost monotonically, which is a very interesting property for system implementation. This is in sharp contrast with the traditional *learning phase* of learning algorithms such as *Q-learning* when the average reward changes rapidly. The learning method converges to a locally optimum policy. It is noted that convergence of the controller parameter  $\theta$  implies convergence of policies.

### 5.4.3 Implementation issues: traffic and scalability

The learning method is valid regardless of the statistical assumptions on traffic. Namely the validity of the policy gradient approach was shown by [10] even in the partially observable case.

It is noted that the algorithm is fully scalable (linear complexity) when the number of relays increase since all the components of the descent direction  $\Delta(t)(\theta_n)$  are estimated from the same sample path of the POMDP, incurring no additional costs when  $N_R$  or N increases. This is fundamental since some deployment scenarios include 30 RSs per BS.

### 5.4.4 Numerical experiments

We evaluate the performance of the learning algorithm in the same setting as Section 5.3. Figures 5.12 and 5.13 represent the evolution of the mean file transfer time and the controller parameters  $(\theta_1, \theta_2, \theta_3)$  respectively during the learning period. One update of  $\theta$  corresponds to  $10^3$  iterations of the underlying POMDP. As stated above, the mean file transfer time decreases in an almost monotonic fashion. The small variations are a numerical artefact due to the fact that the average reward is calculated on a finite number of iterations of the POMDP.

We run the learning process successively a 100 times from an initial condition randomly chosen in  $[-5, 5]^{(2N_R+1)N}$ , and we calculate the file transfer time at the value of  $\theta$  returned by the learning procedure. We calculate the global optimum by a global search (particle swarm optimization was used here). We then plot the c.d.f of the performance gap between the learning process and the global optimum on figure 5.14. In the worst case, the gap is of 25%, and the median performance gap is 11%. Hence despite its local nature and relatively low computational complexity, the learning procedure performs quite well when compared to a global search.

We compare between Poisson arrivals, and arrivals according to a *Markov-modulated Poisson* process with 2 states. Both states have equal stationary probability, the average time spent in a state is 1 minute and the arrival

#### 5.5. PROOFS

rate in state 2 the arrival rate in state 1 multiplied by 3. In each case we estimate the gradient of the cost and calculate the sign of it's dot product with the true gradient. If it is positive then the gradient estimate is a valid ascent direction, and the accuracy of the gradient is the probability of this dot product being positive. We plot the gradient accuracy as a function of the length of the simulation on figure 5.15. As expected, the accuracy is less for Markov modulated arrivals than for Poisson arrivals, since the arrivals tend to be more bursty, but the gap is not very large. This suggests that the learning procedure has good numerical performance even when the arrivals are correlated.



Figure 5.12: File transfer time during the learning process

# 5.5 Proofs

# 5.5.1 Proof of theorem 5.1

The process of arrivals and service requirements is  $\{T_k, \mathbf{1}_{\mathbb{A}_s}(r_k)\frac{\sigma_k}{R_s(r_k)}\}_{k\in\mathbb{Z}}$  for the link between users and station s, and  $\{T_k, \mathbf{1}_{\mathbb{A}_s}(r_k)\frac{\sigma_k}{R_{rel,s}}\}_{k\in\mathbb{Z}}$  for the link between the BS and RS s. Since the arrival process is stationary ergodic,



Figure 5.13: Controller parameters  $(\theta_1, \theta_2, \theta_3)$  during the learning process

Loynes theorem (see section 2.1) gives the stability conditions:

$$\lambda_0 |\mathbb{A}| E[\sigma] E_T^0 \left[ \frac{\mathbf{1}_{\mathbb{A}_s}(r_0)}{R_s(r_0)} \right] < (1-x), \tag{5.44}$$

$$\lambda_0 |\mathbb{A}| E[\sigma] \sum_{s=1}^{N_R} E_T^0 \left[ \frac{\mathbf{1}_{\mathbb{A}_s}(r_0)}{R_{rel,s}} \right] < x$$
(5.45)

with  $E_T^0$  the *Palm expectation* with respect to the arrival instants. The conditions are valid for a G/G/1/PS queue since Loynes theorem holds for all work-conserving service disciplines.

We write the capacity of the link between the BS and users:

$$C_0(x) = (1-x) \left( \int_{\mathbb{A}_0} \frac{1}{R_0(r)} dr \right)^{-1}$$
(5.46)

and the capacity of the link between the BS and RSs:

$$C_{rel}(x) = x \left(\sum_{s=1}^{N_R} \frac{|\mathbb{A}_s|}{R_{rel,s}}\right)^{-1}$$
(5.47)

Assume that the link between the BS and RSs is stable, its output process is stationary ergodic, and using a flow conservation argument it has the same



Figure 5.14: Comparison between local and global optima

intensity as the input. The capacity of the link between RS s and its users is then:

$$C_s(x) = (1-x) \left( \int_{\mathbb{A}_s} \frac{1}{R_s(r)} dr \right)^{-1}$$
(5.48)

The stability of the system is equivalent to the stability of all queues, hence:

$$C(x) = \min\left(C_{rel}(x), \min_{0 \le s \le N_R}(C_s(x))\right).$$
 (5.49)

Furthermore  $x \to C_{rel}(x)$  is strictly increasing and  $x \to \min_{0 \le s \le N_R} (C_s(x))$  is strictly decreasing, hence the unique optimal point  $x^*$  is:

$$x^{*} = \frac{\left(\max_{0 \le s \le N_{R}} \int_{\mathbb{A}_{s}} \frac{1}{R_{s}(r)} dr\right)^{-1}}{\left(\max_{0 \le s \le N_{R}} \int_{\mathbb{A}_{s}} \frac{1}{R_{s}(r)} dr\right)^{-1} + \left(\sum_{s=1}^{N_{R}} \frac{|\mathbb{A}_{s}|}{R_{rel,s}}\right)^{-1}}$$
(5.50)

Substitution of  $x^*$  in the capacity formula yields the maximal capacity which concludes the demonstration.

# 5.5.2 Traffic estimation

**Theorem 5.4.** Consider T > 0 a measurement time and  $f : \mathbb{A} \to \mathbb{R}$  - a function which is measurable, positive and bounded. We define the sequence



Figure 5.15: Impact of correlated arrivals on gradient estimation accuracy

 $\{F_n\}_{n\in\mathbb{Z}}$ :

$$F_n = \frac{1}{T} \sum_{k \in \mathbb{Z}} f(r_k) \mathbf{1}_{[nT,(n+1)T)}(T_k).$$
 (5.51)

We decompose  $F_n$  as a sum of its expectation, a martingale difference and a term due to the memory of the arrival process:

$$F_n = E[F_n] + M_n + G_n, (5.52)$$

$$M_n = F_n - E[F_n | \xi_{Tn}], \qquad (5.53)$$

$$G_n = E[F_n|\xi_{T_n}] - E[F_n].$$
 (5.54)

With assumptions 5.1:

$$E[F_n] = \int_{\mathbb{A}} \lambda(r) f(r) dr.$$
(5.55)

For assumptions 5.2, we further have that:

$$\sup_{n} E\left[F_n^2\right] < +\infty,\tag{5.56}$$

and for  $\gamma > \frac{1}{2}$ :

$$\frac{1}{N^{\gamma}} \sum_{n=1}^{N} M_n \xrightarrow[N \to +\infty]{a.s} 0.$$
(5.57)
#### 5.5. PROOFS

*Furthermore:* 

$$\frac{1}{N}\sum_{n=1}^{N}G_n \xrightarrow[N \to +\infty]{a.s} 0.$$
(5.58)

Finally, for assumptions 5.3,  $G_n \equiv 0$ .

We introduce another assumption on the mixing properties of the arrival process which is necessary for further results:

**Assumption 5.5.** There exists  $\gamma_0 < 1$  such that for any measurable positive and bounded function f, if  $\gamma_0 < \gamma \leq 1$ :

$$\frac{1}{N^{\gamma}} \sum_{n=1}^{N} G_n \xrightarrow[N \to +\infty]{a.s} 0.$$
(5.59)

It is noted that for Poisson arrivals (assumptions 5.3), assumptions 5.5 are not needed since  $G_n \equiv 0$ .

*Proof.* Applying the *Campbell formula* (see section 2.1) to  $(r,t) \rightarrow \frac{1}{T}f(r)\mathbf{1}_{[nT,(n+1)T)}(t)$  proves the first claim. We define  $||f||_{\infty} = \sup_{r \in \mathbb{A}} |f(r)|$ . The second claim is proven by:

$$\sup_{n} E\left[F_{n}^{2}\right] \leq \frac{\left\|f\right\|_{\infty}^{2}}{T^{2}} E\left[N([0,T)\times\mathbb{A})^{2}\right] < +\infty.$$
(5.60)

Define  $S_N = \sum_{n=1}^N M_n$ .  $S_n$  is a martingale and  $E[M_n^2] \leq 2\sup_n E[F_n^2] < +\infty$ . Applying the *law of large numbers for martingales* ([43, 44]) proves the third claim. Because we have assumed ergodicity of the arrival process,

$$\frac{1}{N}\sum_{n=1}^{N}F_{n} \xrightarrow[N \to +\infty]{} E\left[F_{n}\right], \qquad (5.61)$$

so that:

$$\frac{1}{N}\sum_{n=1}^{N}G_{n} \xrightarrow[N \to +\infty]{} 0, \qquad (5.62)$$

which proves the last claim.

### 5.5.3 Proof of theorem 5.2

### ODE

Since  $\rho$  does not change, we sometimes omit it for notation clarity. Consider the ODE  $\dot{x} = g(x)$ , and define the Lyapunov function :

$$U(x) = \frac{1}{2} \sum_{s=1}^{N_R} \frac{g_s(x)^2}{\rho_{rel,s}}.$$
 (5.63)

We calculate its gradient:

$$\frac{\partial U}{\partial x_s}(x) = -\frac{g_s(x)\rho_s}{\rho_{rel,s}} - \sum_{s'=1}^{N_R} g_{s'}(x).$$
(5.64)

Its derivative along solutions is:

$$\langle \nabla U, \dot{x} \rangle = -\sum_{s=1}^{N_R} \frac{g_s(x)^2 \rho_s}{\rho_{rel,s}} - \left(\sum_{s=1}^{N_R} g_s(x)\right)^2 < 0$$
 (5.65)

It is noted that U is indeed positive definite and radially unbounded. This proves that  $x^*$  is the unique equilibrium of the ODE and that it is globally asymptotically stable. Namely all solutions of the ODE converge to  $x^*$ , regardless of the initial condition.

#### Projected ODE

We have to take the constraint set H into account. Namely, since the iterates are projected on H, they follow the trajectory of the ODE *projected* on H. In the general case, we need to add a projection term to the ODE, that is:

$$\dot{x} \in g(x) + G(x). \tag{5.66}$$

G(x) is the minimal "force" which ensures that solutions remain in the constraint set H, and  $G(x) \neq \{0\}$  only if x belongs to the boundary of H. We prove here that solutions of the (non-projected) ODE starting in H remain in it, hence  $G(x) \equiv \{0\}$ . If  $x_s = 0$ , then:

$$\dot{x}_s = \rho_{rel,s} (1 - \sum_{s'=1}^{N_R} x_{s'}) \ge 0,$$
 (5.67)

and if  $\sum_{s=1}^{N_R} x_s = 1$  then:

$$\frac{d}{dt} (\sum_{s=1}^{N_R} x_s) = -\sum_{s=1}^{N_R} \rho_s x_s < 0.$$
(5.68)

This proves that H is an invariant set of the ODE without the need to add a projection term.

Stochastic approximation: decreasing step sizes

It is noted that  $x \to g(\rho, x)$  is affine hence smooth. We verify the necessary conditions for stochastic approximation theorems to be valid:

 $- \sup_{\substack{n \\ \text{rem 5.4,}}} E\left[g_s(\rho[n], x[n])^2\right] \le \sup_{n} E\left[(\rho_{rel,s}[n] + \rho_s[n])^2\right] < +\infty \text{ from theo-$ 

- $-x \rightarrow E[g(\rho[n], x)|\xi[n]]$  is continuous
- $\frac{1}{N^{\gamma}} \sum_{n=1}^{N} g(\rho[n], x[n]) E\left[g(\rho[n], x[n]) | \xi[n]\right] \xrightarrow[N \to +\infty]{a.s} 0, \text{ for } \frac{1}{2} < \gamma \leq 1$ from theorem 5.4
- $\frac{1}{N^{\gamma}} \sum_{n=1}^{N} E\left[g(\rho[n], x[n])\right] E\left[g(\rho[n], x[n])|\xi[n]\right] \xrightarrow[N \to +\infty]{a.s} 0, \text{ for } \gamma_0 < \gamma \leq 1$  from assumptions 5.5
- $-x \rightarrow E[g(\rho[n], x)|\xi[n]]$  is Lipschitz continuous uniformly in  $\xi[n]$ , because  $\sup_{\xi[n]\in\Xi} E[\rho_{rel,s}[n] + \rho_s[n]|\xi[n]] < +\infty.$

Applying theorem 2.9 proves that  $x[n] \xrightarrow[n \to +\infty]{a.s} x^*(\rho)$ .

Stochastic approximation: constant step sizes

For constant step sizes, the following properties are needed:

- $\Xi$  is a compact space, and  $\{\xi[n]\}_n$  does not depend on  $\{x[n]\}_n$  i.e the noise process is exogenous
- $\{g(\rho[n], x[n])\}_n$  is uniformly integrable since it is bounded in mean square
- $-x \to E[g(\rho[n], x)|\xi[n]]$  is continuous
- $\{E[g(\rho[n], x[n])|\xi[n]])\}_n$  and  $\{E[g(\rho[n], x)|\xi[n]]\}_n$  are uniformly integrable since they are bounded in mean square
- $\frac{1}{N} \sum_{n=1}^{N} \left( E\left[g(\rho[n], x[n])\right] E\left[g(\rho[n], x[n])|\xi[n]\right] \right) \xrightarrow[N \to +\infty]{a.s} 0 \text{ (actually the proof only requires convergence in probability)}$

Applying theorem 2.10 proves that the sequence  $\{x[n]\}_n$  converges to  $x^*(\rho)$  in distribution.

### 5.5.4 Proof of theorem 5.3

 $P \to \rho(P)$  is Lipschitz continuous and all its components are bounded away from 0 on  $\mathcal{P}$ , hence  $P \to x^*(\rho(P))$  is Lipschitz continuous as well. It is also noted that  $E[h(\rho[n], P)] = h(\rho(P[n]), P[n])$  by linearity.

Decreasing step sizes

We have that:

 $- \sup_{n} E[h_s(\rho[n], P[n])^2] \le \sup_{n} P_{max}^2 E[(\rho_s[n] + \rho_0[n])^2] < +\infty \text{ from theo-rem 5.4,}$ 

- $P \rightarrow E[h(\rho[n], P)|\xi[n]]$  is continuous
- $\frac{1}{N^{\gamma}} \sum_{n=1}^{N} h(\rho[n], P[n]) E[h(\rho[n], P[n]) | \xi[n]] \xrightarrow[N \to +\infty]{a.s} 0 \text{ for } \frac{1}{2} < \gamma \leq 1$ from theorem 5.4
- $\frac{1}{N^{\gamma}} \sum_{n=1}^{N} E\left[h(\rho[n], P[n])\right] E\left[h(\rho[n], P[n])|\xi[n]\right] \xrightarrow[N \to +\infty]{a.s} 0, \text{ for } \gamma_0 < \gamma \leq 1 \text{ from assumptions 5.5}$
- $-P \to E[h(\rho[n], P)|\xi[n]] \text{ is Lipschitz continuous uniformly in } \xi[n], \text{ be$  $cause } P \to |\mathbb{A}_s(P)| \text{ is Lipschitz continuous, and } \sup_{\xi \in \Xi} \sup_{r \in \mathbb{A}} \overline{\lambda}(r, [0, T), \xi_0) < +\infty.$

Combining theorem 4.4 and theorem 2.9 for  $P_{min}$  sufficiently small and  $P_{max}$  sufficiently large, proves that the sequence  $\{P[n]\}$  converges a.s to  $\mathcal{L}$ .

Following the same method as [16] [Lemma 1, Chapter 6, page 66], we can rewrite the update equations as:

$$x_s[n+1] = [x_s[n] + \epsilon_n g_s(\rho[n], x[n])]_H^+$$
(5.69)

$$P_s[n+1] = \left[ P_s[n] + \epsilon_n \frac{\delta_n}{\epsilon_n} h_s(\rho[n], P[n]) \right]_{\mathcal{P}}^{\top}.$$
 (5.70)

In particular:

$$\left| \frac{\delta_n}{\epsilon_n} E\left[ h_s(\rho[n], P[n]) \right] \right| \leq \frac{\delta_n}{\epsilon_n} \sup_n \sqrt{E\left[ h_s(\rho[n], P[n])^2 \right]}$$
  
$$\xrightarrow[n \to +\infty]{} 0$$
(5.71)

Applying theorem 2.9 once again, we have that  $\{x[n], P[n]\}_n$  converges a.s to the set  $\{(x^*(\rho(P)), P) : P \in \mathcal{P}\}$ , which an asymptotically stable set for the ODE:

$$\dot{x(t)} = g(x(t)), \dot{P}(t) = 0,$$
 (5.72)

projected on  $H \times \mathcal{P}$ . Hence  $\{(x[n], P[n])\}_n$  converges a.s a set on which all loads are equal for decreasing step sizes.

Constant step sizes

For the constant step sizes:

- $\Xi$  is a compact space, and the noise process is exogenous
- $\ \{h(\rho[n], P[n])\}_n$  is uniformly integrable since it is bounded in mean square
- $P \rightarrow E[h(\rho[n], P)|\xi[n]]$  is continuous

#### 5.5. PROOFS

- $\{E[h_s(\rho[n], P[n])|\xi[n]])\}_n$  and  $\{E[h(\rho[n], P)|\xi[n]]\}_n$  are uniformly integrable since it they are bounded in mean square
- $\frac{1}{N} \sum_{n=1}^{N} (E\left[h(\rho[n], P[n])\right] E\left[h(\rho[n], P[n])|\xi[n]\right]) \xrightarrow[N \to +\infty]{a.s} 0 \text{ (and in probability)}$

From theorem 4.4, and theorem 2.10, for  $P_{min}$  sufficiently small and  $P_{max}$  sufficiently large, this proves that  $\{P[n]\}_n$  converges in distribution to  $\mathcal{L}$  when  $\epsilon \to 0^+$ . Using the same technique as in the decreasing step size case, we write

$$x_s[n+1] = [x_s[n] + \epsilon g_s(\rho[n], x[n])]_H^+$$
(5.73)

$$P_s[n+1] = \left[ P_s[n] + \epsilon \frac{\delta(\epsilon)}{\epsilon} h_s(\rho[n], P[n]) \right]_{\mathcal{P}}^+, \qquad (5.74)$$

and:

$$\frac{\delta(\epsilon)}{\epsilon n''} \left| \sum_{n=n'}^{n'+n''} E\left[h_s(\rho[n], P[n])\right] \right| \\ \leq \frac{\delta(\epsilon)}{\epsilon} \sup_n \sqrt{E\left[h_s(\rho[n], P[n])^2\right]} \underset{\epsilon \to 0^+}{\to} 0$$
(5.75)

so theorem 2.10 proves that  $\{x[n], P[n]\}_n$  converges in distribution to  $\{(x^*(\rho(P)), P) : P \in \mathcal{P}\}$ . This justifies that  $\{(x[n], P[n])\}_n$  converges in distribution to a set on which all loads are equal when  $\epsilon \to 0$ .

### Chapter 6

### Conclusion and future work

In this thesis we have studied the design, modeling and performance evaluation of SON mechanisms in wireless networks. We have proposed SON algorithms for some important use cases. Flow-level dynamics where users arrive and depart dynamically have been taken into account using queuing models. The convergence of the proposed mechanisms has been proven using mathematical tools such as stochastic approximation and reinforcement learning. The proposed solutions fulfill the important requirements of: implementability in the control plane, stability, robustness to noise, low signaling overhead and tolerance to delays.

In this thesis, we have developed and studied SON mechanisms as standalone entities. This has allowed us to establish convergence of a SON mechanism when it is the only entity modifying network parameters. In practical deployments, several SON mechanisms will act simultaneously on the network parameters, on the same time scale. One can for example think of a load balancing mechanism adjusting cell sizes while another mechanism performs ICIC. Those two mechanisms have different objectives and there is no guarantee that their interaction will not cause instability. For the SON technology to be adopted by network operators, there is a need for a simple and robust *coordination mechanism*. We believe that the coordination problem is one of the most important open problems in SON research, and an enabler for large scale deployment of SON.

Another interesting perspective would be to consider more realistic queuing models and investigate whether the convergence of the proposed SON mechanisms still holds. The influence of user mobility and handovers for instance seems to be an interesting problem. The load balancing mechanism has been designed to *minimize the maximal load* and ensure *stability* whenever possible. This is roughly equivalent to minimizing the *blocking rate* of the network. It would be interesting to develop a load balancing mechanism which minimizes the average *file transfer time*.

### Chapter 7

# Appendices

### 7.1 Simulation methodology

We detail briefly the standard simulation methodology used for system simulations. Unless explicitly stated, all network simulation results contained in this thesis use this model.

The signal attenuation from location r to r' is the product of two terms:

$$L(r,r') = D(r,r')10^{\frac{S(r,r')}{10}},$$
(7.1)

with D and S path loss and shadowing respectively. *Path loss* represents losses due to distance on a coarse spatial scale and is chosen as a power law:

$$D(r,r') = \frac{A}{\|r - r'\|^{\eta}},$$
(7.2)

with  $\eta \geq 2$ .  $\eta = 2$  corresponds to free space propagation and  $\eta = 3.5$  to a dense urban environment. *Shadowing* represents variations of the signal attenuation on a finer spatial scale due to obstacles such as buildings. Shadowing is chosen as a stationary centered Gaussian process:

$$S(r, r') \equiv \mathcal{N}(0, \sigma^2), \tag{7.3}$$

with  $\sigma$  the shadowing standard deviation. A typical value for  $\sigma$  is between 0 and 10 dB. We do not assume  $r' \to S(r, r')$  to be a white process. Typically the law of  $r' \to S(r, r')$  involves a *correlation distance* d; i.e

$$E[S(r,r')S(r,r'')] \approx 0 \text{ if } ||r''-r'|| \ge d.$$
(7.4)

We assume that shadowing processes of two transmitters (e.g base stations) placed at different locations are independent. This is reasonable since we typically consider base stations which are several hundreds of meters apart.



Figure 7.1: 19 cells hexagonal network with wrap-around

Given  $N_{BS}$  base stations with locations  $\{r_s\}_{1 \le s \le N_{BS}}$  and transmitted powers  $\{P_s\}_{1 \le s \le N_{BS}}$ , the SINR at location r while being served by base station s is calculated by:

$$SINR(r,s) = \frac{P_s L(r_s, r)}{W N_0 + \sum_{s' \neq s} P_{s'} L(r_{s'}, r)}$$
(7.5)

with W the bandwidth and  $N_0$  the thermal noise spectral power density.

The basic network consists of 19 hexagonal cells. In order to avoid border effects, a wrap-around technique is used, and is equivalent to placing the stations on a torus. Figure 7.1 represents the network layout.

7.2. ACRONYMS 155	5
7.2 Acronyms	
<b>3GPP</b> 3rd Generation Partnership Project	!1
<b>AEP</b> Asymptotic Equi-repartition Property	0
<b>a.e</b> almost everywhere	3
<b>a.s</b> almost surely	5
<b>AWGN</b> Additive White Gaussian Noise	5
BS Base Station	9
c.c.d.f complementary cumulative distribution function108	18
c.d.f cumulative distribution function	\$5
<b>CDMA</b> Code Division Multiple Access	0
<b>CTMDP</b> Continuous Time Markov Decision Process	'0
<b>DTMDP</b> Discrete Time Markov Decision Process	'1
<b>FTP</b> File Transfer Protocol	5
<b>GPS</b> Global Positioning System	25
HetNet Heterogeneous Network	24
HSPA High Speed Packet Access	0

ICIC Inter-Cell Interference Coordination
i.i.d independent and identically distributed
iif if and only if
<b>ITU</b> International Telecommunication Union
<b>KKT</b> Karush-Kuhn-Tucker
<b>KPI</b> Key Performance Indicator
LTE Long Term Evolution
MAC Medium Access Control
MDP Markov Decision Process
MIMO Multiple Input Multiple Output
MMF Max-Min Fair
MTP Max Throughput
NGMN Next Generation Mobile Network
<b>ODE</b> Ordinary Differential Equation
<b>OFDMA</b> Orthogonal Frequency-Division Multiple Access

7.2. ACRONYMS	157
<b>OMC</b> Operation and Maintenance Center	19
<b>PF</b> Proportional Fair	73
<b>POMDP</b> Partially Observable Markov Decision Process	137
<b>PRB</b> Physical Resource Block	100
$\mathbf{p.d.f}$ probability density function	
<b>PS</b> Processor Sharing	
QoS Quality of Service	23
<b>REM</b> Radio Environment Map	25
<b>RR</b> Round Robin	
<b>RRM</b> Radio Resource Management	20
<b>RS</b> Relay Station	117
<b>SINR</b> Signal to Interference plus Noise Ratio	51
<b>SMDP</b> Semi-Markov Decision Process	69
<b>SNR</b> Signal to Noise Ratio	
SON Self-organizing networks	

158	CHAPTER 7. APPENDICES
<b>TDMA</b> Time Division Multiple Access.	
<b>V-BLAST</b> Vertical Bell Labs Space-Tin	ne52

# Index

#### 3GPP, 21

achievable rate, 41, 43 channel code, 42 achievable throughputs, 75 channel hardening, 87 admission control, 96, 126 AEP, 40 chip time, 50 alpha-fair utility, 76 coding function, 42 attractor, 55 coding theory, 45 autonomics, 20 communication channel, 42 continuous, 45 baseline, 67, 69 discrete, 42 basin of attraction, 55 memoryless, 42 Bellman equation, 62 configuration, 31 best-effort, 37 locally finite, 32 blocking rate, 36, 108, 130, 135 simple, 32 Campbell formula, 32, 145 Campbell measure, 34 control plane, 25 capacity, 38 convex optimization, 43 CDMA, 50coordination, 27 channel correlated arrivals, 120 AWGN, 45 correlation distance, 153 AWGN block fading, 48 AWGN in practice, 47 cyclic prefix, 49 band-limited AWGN, 46 block fading, 48 decoding function, 42 coherence time, 57discounted reward, 62 ergodic, 48 drive test, 25 fading, 48 drive test minimization, 25 MIMO Rayleigh-fading AWGN, 51, dropped call, 20, 97, 102 87, 91 multi-tap Rayleigh-fading AWGN, elastic traffic, 103, 107 49, 84, 90 energy savings, 24

parallel AWGN, 46 Rayleigh-fading AWGN, 82, 88 channel-aware scheduling, 73 constant data rate traffic, 107 coverage-capacity optimization, 93

entropy, 38 chain rule, 40 conditional, 38 differential, 45 ergodic transformation, 33 exploration probability, 64 fairness, 76 Fano's inequality, 39 filtration, 53, 119 finite difference, 53, 65 first-order measure, 32, 119 flow-level dynamics, 102, 117 frequency reuse, 23 full buffer, 74 generalization, 64 GPS, 25 gradient descent, 56 green networking, 24 Gronwall's lemma, 80 handover, 20, 102 margin, 24 heavy-tailed traffic, 120 heterogeneous network, 24, 117 hitchhiker's paradox, 34 hot-spot, 129 hot-spots, 23 ICIC, 23 information capacity, 43 insensitivity, 37 inter-cell interference, 117, 121 inter-code interference, 50, 84 inter-symbol interference, 50 invariant set, 54 joint typicality, 44 Kamke condition, 80 Kaufman-Roberts algorithm, 36

KKT conditions, 46

law of large numbers for martingales, 145learning phase, 140 likelihood ratio, 65 link-level curve, 47 load, 34, 35, 37 elastic traffic, 103 estimation, 102, 104, 126 streaming traffic, 107 load balancing, 23, 101, 106, 118, 124 loss system, 35 Loynes theorem, 34, 142 Lyapunov function, 55, 145 stability, 54 M/G/1/PS, 37, 103management plane, 26 Markov property, 60 Markov-modulated Poisson, 118, 120, 140marks, 32 max-weight, 131 MDP, 60, 118, 132 continuous time, 70 partially observable, 137 **SMDP**, 69 MIMO, 51 mixing time, 69 mobility robustness, 24 model-free, 63, 137 modified Shannon formula, 47 monotone dynamical system, 79 multi-rate Erlang model, 35, 107 multi-user detection, 40 mutual information, 39, 45 network capacity, 103 network management, 19 NGMN, 21 noisy channel coding theorem, 37

#### INDEX

Nyquist-Shannon sampling theorem, relay station, 117 47 OFDMA, 49 OMC, 26 on-line learning, 65 orthogonality factor, 51, 84 packet scheduling, 73 Palm distribution, 34 Palm expectation, 34, 142 particle swarm optimization, 140 path loss, 153 perfect channel knowledge, 74 Picard-Lindelof theorem, 79, 113 point process, 31, 118 ergodic, 33 intensity, 33 marked, 32 simple, 32stationary, 33 Poisson process, 33 policy, 133 deterministic, 67 history dependent, 60 Markov, 60 stochastic, 67, 138 policy gradient, 65, 118 infinite-horizon, 68 processor sharing, 120, 124 Q function, 63 Q-learning, 64, 140 QoS, 24queuing theory, 31 RAKE receiver, 51, 84 Rayleigh fading, 48, 122 reference station, 105 **REINFORCE**, 65 reinforcement learning, 59, 118 relay gain, 122

REM, 25 robustness to noise, 26 round-robin scheduler, 82, 120 **RRM**, 20 scalability, 65 scheduling gain, 82 scheduling policy, 74 scrambling code, 51 self-configuration, 20 self-healing, 20 self-optimization, 20 semi-dynamic network simulator, 97 shadowing, 153 shift operator, 33 signal attenuation, 153 sleep mode, 24 SNR, 46 SON, 19 source code, 41 source coding theorem, 37 spreading codes, 50 stability, 26, 53, 120 stochastic approximation, 52, 118, 128 stochastic gradient, 67, 76 sub-carriers, 49 symbol time, 50 time to trigger, 24 total reward, 61 tracking, 56 traffic intensity, 35, 103 training sequence, 48 transition probabilities, 61 typical set, 40, 43uniformization, 69, 71, 135, 139 user association, 23 V-BLAST, 52 value function, 62

INDEX

value iteration, 63, 118, 135 variance reduction, 67 void probabilities, 32

Walsh-Hadamard code, 51 water-filling, 46, 52 work-conserving, 35 workload, 35

X2 interface, 27

## Bibliography

- 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Study on minimization of drive-tests in next generation networks. TR 36.805, 3rd Generation Partnership Project (3GPP), January 2010.
- [2] 3GPP. Telecommunication management; Study on Energy Savings Management (ESM). TR 32.826, 3rd Generation Partnership Project (3GPP), April 2010.
- [3] 3GPP. User equipment (ue) radio transmission and reception (fdd) (release 8). TS 25.101, 3rd Generation Partnership Project (3GPP), March 2010.
- [4] 3GPP. Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions. TS 36.902, 3rd Generation Partnership Project (3GPP), April 2011.
- [5] 3GPP. Telecommunication management; Self-configuration of network elements; Concepts and requirements. TS 32.501, 3rd Generation Partnership Project (3GPP), April 2011.
- [6] 3GPP. Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements. TS 32.500, 3rd Generation Partnership Project (3GPP), December 2011.
- [7] 3GPP. Telecommunication management; Self-Organizing Networks (SON); Self-healing concepts and requirements. TS 32.541, 3rd Generation Partnership Project (3GPP), March 2011.
- [8] S. Akbarzadeh, R. Combes, and Z. Altman. Network capacity enhancement of OFDMA system using self-organized femtocell off-load. In *IEEE Wireless Communications and Networking Conference (WCNC 2012)*, april 2012.

- [9] F. Baccelli and P. Bremaud. *Elements of Queueing Theory. Palm Martingale Calculus and Stochastic Recurrences.* Springer, 2nd ed, 2003.
- [10] J. Baxter and P. L. Bartlett. Infinite-Horizon Policy-Gradient Estimation. Journal of Artificial Intelligence Research, 15:319–350, 2001.
- [11] F. Bergren and R. Jantti. Asymptotically fair transmission scheduling over fading channels. *IEEE transactions on wireless communications*, 3:326–336, January 2004.
- [12] David Blackwell. Discrete dynamic programming. Annals of Mathematical Statistics, 33(2):719–726, 1962.
- [13] B. Blaszcyszyn and M. Karray. Fading effect on the dynamic performance evaluation of OFDMA cellular networks. In 1st International Conference on Communications and Networking, 2009.
- [14] T. Bonald and A. Proutière. Wireless downlink data channels: User performance and cell dimensioning. In ACM Mobicom, 2003.
- [15] Thomas Bonald and Mathieu Feuillet. Network Performance Analysis. ISTE Ltd and John Wiley and Sons Inc, September 2011.
- [16] Vivek S. Borkar. Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press, 2008.
- [17] Sem Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Trans. Netw.*, 13:636–647, June 2005.
- [18] Gregory E. Bottomley, Tony Ottosson, and Yi-Pin Eric Wang. A generalized rake receiver for interference suppression. *IEEE Journal on Selected Areas in Communications*, 18(8):1536–1545, 2000.
- [19] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508– 2530, June 2006.
- [20] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, March 2004.
- [21] Schlegel Christian. Error probability calculation for multibeam rayleigh channels. *IEEE Transactions on Communications*, 44:290–293, 1996.

- [22] R. Combes, Z. Altman, and E. Altman. On the use of packet scheduling in self-optimization processes: Application to coverage-capacity optimization. In 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010, pages 98 –107, june 2010.
- [23] R. Combes, Z. Altman, and E. Altman. Scheduling gain for frequencyselective rayleigh-fading channels with application to self-organizing packet scheduling. *Performance Evaluation*, February 2011.
- [24] R. Combes, Z. Altman, and E. Altman. A self-optimization method for coverage-capacity optimization in OFDMA networks with MIMO. In 5th ICST International Conference on Performance Evaluation Methodologies and Tools (ValueTools), 2011, may 2011.
- [25] R. Combes, Z. Altman, and E. Altman. Self-organizing relays in LTE networks: Queuing analysis and algorithms. In 7th International Conference on Network and Service Management (CNSM), 2011, october 2011.
- [26] R. Combes, Z. Altman, and E. Altman. Self-organization in wireless networks: A flow-level perspective. In *The 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM* 2012), april 2012.
- [27] R. Combes, Z. Altman, and E. Altman. Self-organizing relays: Dimensioning, self-optimization and learning. *IEEE Transactions on Network and Service Management*, 2012.
- [28] R. Combes, S.E Elayoubi, and Z. Altman. Cross-layer analysis of scheduling gains: Application to lmmse receivers in frequency-selective rayleigh-fading channels. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011*, pages 133–139, may 2011.
- [29] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.
- [30] D. J. Dalay and D. Vere-Jones. An introduction to the theory of point processes. Springer, 2002.
- [31] Alan E. Gelfand, Peter Diggle, and Peter Guttorp. *Handbook of Spatial Statistics*. CRC Press, 2010.

- [32] B.M. Hochwald, T.L. Marzetta, and V. Tarokh. Multiple-antenna channel hardening and its implications for rate feedback and scheduling. *Information Theory, IEEE Transactions on*, 50(9):1893 – 1909, sep 2004.
- [33] ITU. Guidelines for evaluations of radio transmission technologies for imt-2000. ITU-R M.1225, International Telecommunication Union (ITU), 1997.
- [34] William C. Jakes. Microwave Mobile Communications. IEEE Press, 1974.
- [35] M. A. Kamath and B. L. Hughes. The asymptotic capacity of multipleantenna rayleigh-fading channels. *IEEE Transactions on Information Theory*, 51(12):4325–4333, 2005.
- [36] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. Annals of Mathematical Statistics, 23:462–466, September 1952.
- [37] Leonard Kleinrock. *Queuing Systems*. Wiley Interscience, 1975.
- [38] Harold J. Kushner and G. George Yin. Stochastic Approximation and Recursive Algorithms and Applications 2nd edition. Springer Stochastic Modeling and Applied Probability, 2003.
- [39] H.J. Kushner and P.A. Whiting. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE transactions on wireless communications*, 3:1250–1259, July 2004.
- [40] Tian Lan, D. Kao, Mung Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *INFOCOM*, 2010 *Proceedings IEEE*, pages 1 –9, mar. 2010.
- [41] John D. C. Little. A Proof for the Queuing Formula:  $L = \lambda$  W. Operations Research, 9(3):383–387, 1961.
- [42] L. Ljung. Analysis of recursive stochastic algorithms. Automatic Control, IEEE Transactions on, 22(4):551–575, 1977.
- [43] Michel Loève. *Probability Theory I.* Graduate Texts in Mathematics. Springer, 1977.
- [44] Michel Loève. Probability Theory II. Graduate Texts in Mathematics. Springer, 1978.

- [45] R. M. Loynes. The stability of a queue with non-independent interarrival and service times. *Mathematical Proceedings of the Cambridge Philosophical Society*, 58(03):497–520, 1962.
- [46] J. Mo and J. Walrand. Fair end-to-end window based congestion control. *IEEE transactions networking*, 8:556–566, October 2000.
- [47] P. Mogensen, Wei Na, I.Z. Kovacs, F. Frederiksen, A. Pokhariyal, K.I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela. LTE capacity compared to the shannon bound. In *Vehicular Technology Conference*, 2007. VTC2007-Spring. IEEE 65th, pages 1234 –1238, april 2007.
- [48] NGMN. Informative List of SON Use Cases. Technical report, The Next Generation Mobile Networks (NGMN) Alliance, April 2007.
- [49] NGMN. Use Cases related to Self Organising Network, Overall Description. Technical report, The Next Generation Mobile Networks (NGMN) Alliance, December 2008.
- [50] Martin L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley-Interscience, 2005.
- [51] H. Robbins and S. Monro. A stochastic approximation method. Annals of Mathematical Statistics, 22:400–407, September 1951.
- [52] Letian Rong, S.E. Elayoubi, and O.B. Haddada. Impact of relays on lte-advanced performance. In *Communications (ICC)*, 2010 IEEE International Conference on, pages 1–6, May 2010.
- [53] A. Samhat, Z. Altman, M. Francisco, and B.Fourestie. Semi-dynamic simulator for large scale heterogeneous wireless networks. *Interna*tional Journal on Mobile Network Design and Innovation (IJMNDI), 1(3-4):269–278, 2006.
- [54] C. E. Shannon. A mathematical theory of communication. *Bell system* technical journal, 27, 1948.
- [55] Hal L. Smith. Monotone Dynamical Systems: an Introduction to the Theory of Competitive and Cooperative Systems. American Mathematical Society, 1995.
- [56] R.S. Sutton and A.G. Barto. Reinforcement Learning, an Introduction. MIT Press, 1998.

- [57] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on*, 37(12):1936-1948, December 1992.
- [58] Emre Telatar. Capacity of multi-antenna gaussian channels. *European Transactions on Telecommunications*, 10:585–595, 1999.
- [59] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, June 2005.
- [60] R. Visoz and E. Bejjani. Matched filter bound for multichannel diversity over frequency-selective rayleigh-fading mobile channels. *Vehicular Technology, IEEE Transactions on*, 49(5):1832–1845, September 2000.
- [61] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. Machine Learning, 8:279–292, 1992. 10.1007/BF00992698.
- [62] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. 10.1007/BF00992696.