UNIVERSITÀ DI PISA
Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

The Spectrum Shortage Problem: Channel Assignment and Cognitive Networks

Vanessa Gardellin

Anno 2011

UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

The Spectrum Shortage Problem: Channel Assignment and Cognitive Networks

Autore:

Vanessa Gardellin *Firma*__________

Relatori:

Prof. Luciano Lenzini *Firma*__________

Prof. Giuseppe Anastasi *Firma*__________

Prof. Sajal K. Das *Firma*__________

Anno 2011
SSD ING-INF/03

Preface

I miei ringraziamenti vanno a tutti coloro con cui ho condiviso i ventidue anni di studio che mi hanno condotto fino al raggiungimento di questo traguardo.

Come da consuetudine e a pieno merito i miei primi grazie sono rivolti al mio babbo (concedetemi il toscanismo), Giancarlo, e alla mia mamma, Anna. I quali mi hanno sostenuta e incoraggiata in tutte le mie scelte in Italia, in Europa e, loro malgrado, oltre oceano. Nell'ambito familiare un immenso grazie va anche a Ghibli, il quale mi ha mostrato il significato di amore incondizionato, e a Brio che quotidianamente mi mostra come il calore di una casa piena di amore porti serenità dopo una lunga giornata di studio e lavoro.

Un grazie è anche rivolto ad amici, colleghi, compagni di studio e scorribande con cui ho iniziato e trascorso i miei studi, scambiato pensieri, idee e risate. In diversi modi hanno tutti contribuito nel mio percorso formativo, aiutandomi a credere in me stessa, suscitando in me nuovi interessi e la forza di andare avanti. Tra loro un particolare grazie è rivolto a Valentina.

Mi sembra ora giunto il momento di ringraziare coloro che hanno permesso la mia formazione finale. Un grazie va ai miei tutori: Luciano Lenzini, Giuseppe Anastasi, Enzo Mingozzi e Sajal K. Das. Nonchè ai colleghi del gruppo di ricerca: Claudio, Giovanni, Carlo e Daniele.

Un ultimo grazie in ordine cronologico ma non per questo meno importante va al mio compagno di vita. La sua ricerca è stata lunga quanto i miei studi ma come la saggezza popolare suggerisce "Per le cose belle ci vuol tempo", grazie Raffa.

Sommario

Recenti analisi hanno mostrato che il proliferare di applicazioni e servizi wireless, avvenuto nell'ultimo decennio, ha provocato il problema della scarsità delle frequenze. In questo lavoro è fornita una panoramica sul problema della scarsità delle frequenze considerando differenti tecnologie. Inizialmente verranno proposte soluzioni basate su reti mesh multi-antenna multi-canale per migliorare l'utilizzo dello spettro non licenziato. Successivamente verranno analizzate problematiche e soluzioni per l'utilizzo opportunistico delle risorse licenziate in reti cognitive.

Nelle reti mesh, il problema della scarsità dello spettro è trattato mediante l'uso di molteplici antenne settate su canali non interferenti sui dispositivi. Per questo scopo verrà proposto *G-PaMeLA*, un algoritmo che divide in sotto problemi locali l'allocazione di canali e la creazione del routing in reti mesh multi-antenna multi-canale. I risultati ottenuti mostrano che *G-PaMeLA* migliora significativamente le prestazioni della rete in termini di pacchetti persi e distribuzione delle risorse in confronto con algoritmi proposti in letteratura. Sfortunatamente, pur utilizzando canali non interferenti, il sovra-affollamento dello spettro delle frequenze non è risolto.

Per affrontare il problema del sovra-affollamento, attente analisi sono state condotte sullo spettro delle frequenze. Queste analisi hanno identificato l'opportunità di trasmettere su canali licenziati i quali sono sorprendentemente inutilizzati. Per risolvere il problema delle limitate risorse usando canali licenziati, sono state sviluppate le reti cognitive di accesso e mesh.

Nelle reti cognitive di accesso, il maggior problema è la self-coexistence, che è l'abilità di accedere a canali senza creare interferenze ad altri utenti sia licenziati sia non licenziati. In questo lavoro, saranno proposti due algoritmi basati sulla teoria dei giochi i quali si differenziano nel tipo di dispositivi presi in considerazione, non cooperativi (*NoRa*) e cooperativi (*HeCtor*), rispettivamente. I risultati mostrano che *HeCtor* migliora la capacità della rete ma con costi computazionali più elevati, il che porta a basse prestazioni quando l'occupazione dei canali varia rapidamente. Al contrario, *NoRa* ottiene la stessa capacità nella rete indipendentemente dall'occupazione dei canali, quindi i dispositivi si adattano rapidamente a questi cambiamenti.

Nelle reti cognitive mesh, la principale preoccupazione è come i dispositivi si coordinano tra loro in un ambiente che varia nel tempo e a seconda del luogo. A tal proposito sarà proposto *Connor*, un algoritmo di clustering utilizzato per risolvere il problema di coordinamento tra dispositivi il quale stabilisce canali di controllo a livello locale. *Connor*, al contrario degli algoritmi esistenti in letteratura, non richiede sincronizzazione e permette un veloce re-clustering quando si hanno cambiamenti nell'occupazione dei canali da parte di utenti licenziati. I risultati mostrano che *Connor* si comporta meglio di altri algoritmi esistenti in letteratura in termini di numero di canali usati per il controllo e di tempo richiesto per raggiungere e rimanere in una configurazione stabile.

Abstract

Recent studies have shown that the proliferation of wireless applications and services, experienced in the last decade, is leading to the challenging spectrum shortage problem. We provide a general overview regarding the spectrum shortage problem from the point of view of different technologies. First, we propose solutions based on multi-radio multi-channel wireless mesh networks in order to improve the usage of unlicensed wireless resources. Then, we move our focus on cognitive networks in order to analyze issues and solutions to opportunistically use licensed wireless resources.

In wireless mesh networks, the spectrum shortage problem is addressed equipping each device with multiple radios which are turned on different orthogonal channels. We propose *G-PaMeLA*, which splits in local sub-problems the joint channel assignment and routing problem in multi-radio multi-channel wireless mesh networks. Results demonstrate that *G-PaMeLA* significantly improves network performance, in terms of packet loss and throughput fairness compared to algorithms in the literature. Unfortunately, even if orthogonal channels are used, wireless mesh networks result in what is called *spectrum overcrowding*.

In order to address the spectrum overcrowding problem, careful analysis on spectrum frequencies has been conducted. These studies identified the possibility of transmitting on licensed channels, which are surprisingly underutilized. With the aim of addressing the resources problem using licensed channels, cognitive access and mesh networks have been developed.

In cognitive access networks, we identify as the major problem the self-coexistence, which is the ability to access channels on a non-interfering basis with respect to licensed and unlicensed wireless devices. We propose two game theoretic frameworks which differentiate in having non-cooperative (*NoRa*) and cooperative (*HeCtor*) cognitive devices, respectively. Results show that *HeCtor* achieves higher throughput than *NoRa* but at the cost of higher computational complexity, which leads to a smaller throughput in cases where rapid changes occur in channels' occupancy. In contrast, *NoRa* attains the same throughput independent of the variability in channels' occupancy, hence cognitive devices adapt faster to such changes.

In cognitive mesh networks, we analyze the coordination problem among cognitive devices because it is the major concern in implementing mesh networks in environments which change in time and space. We propose *Connor*, a clustering algorithm to address the coordination problem, which establishes common local control channels. *Connor*, in contrast with existing algorithms in the literature, does not require synchronization among cognitive mesh devices and allows a fast re-clustering when changes occur in channel's occupancy by licensed users. Results show that *Connor* performs better than existing algorithms in term of number of channels used for control purposes and time to reach and stay on stable configurations.

Contents

List of Figures

List of Algorithms

List of Tables

List of Acronyms

ACK	Acknowledgement Message
AG	Auction Game
AODV	Ad-hoc On-demand Distance Vector
BD_{conf}	Border Device Confirmation
BD_{inq}	Border Device Inquiry
BD_{req}	Border Device Request
BG	Bargain Game
BS	Base Station
CA	Channel Assignment
C-AN	Cognitive Access Network
CBD	Cluster Border Device
CBP	Coexistence Beacon Protocol
CBR	Constant Bit-Rate
CD	Cognitive Device
CFP	Cluster Formation Phase
CG	Coalitional Game
CHD	Cluster Head Device
CLICA	Connected Low Interference Channel Assignment
CBM	Cluster Beacon Message
CBD	Cluster Border Device
C-MD	Cognitive Mesh Device
C-MHN	Cognitive Multi-Hop Network
CN	Cognitive Network
COD	Cluster Ordinary Device
Connor	Control Channel Formation Protocol
CPE	Consumer Premises Equipment
CR	Cognitive Radio
C-WMN	Cognitive Wireless Mesh Networks
DRR	Deficit Round Robin
DTV	Digital Television
FCC	Federal Communications Commission
FCRA	Flow-based Channel and Rate Assignment
GT	Game Theory
HeCtor	Hedonic Coalitional Formation Game
IE	Information Element
ILP	Integer Linear Programming
IR	Inquiry Request
ISM	Industrial, Scientific and Medical
ISP	Internet Service Providers
JC	Join Confirmation
JCAR	Joint Channel Assignment and Routing
KA	Keep Alive Message
KAP	Keep Alive Phase
LACA	Load-Aware joint Channel Assignment and Routing Algorithm
MAC	Medium Access Control
MC-CDMA	Multi-Carrier Code Division Multiple Access

MD	Mesh Device
MG	Minority Games
MMGMS	Modified Minority Game with Mixed Strategies
MRMC-WMN	Multi-Radio Multi-Channel Wireless Mesh Network
NBS	Nash Bargain Solution
NC-AHN	Non-Cognitive Ad-Hoc Network
NCN	Non-Cognitive Network
NE	Nash Equilibrium
NIC	Network Interface Card
NoRa	Non-cooperative Repeated game
NTU	Non-Transferable Utility
OLSR	Optimized Link State Routing
OSPF	Open Shortest Path First
PaMeLA	Partitioned Mesh Channel Assignment
PG	Potential Game
PU	Primary Users
QoS	Quality of Service
SDSE	Strongly Dominant Strategy Equilibrium
SG	Stochastic Game
SU	Secondary User
TU	Transferable Utility
TVWS	TV White Space
U-NII	Unlicensed National Information Infrastructure
UP	Update Message
UWB	Ultra Wide Band
VoIP	Voice over IP
WMN	Wireless Mesh Network
WRAN	Wireless Regional Area Network

List of Symbols

$\Re$	Real Numbers
$\mathcal{T}(t_1, t_2)$	The larger set of devices transmitting in the time interval $[t_1, t_2]$
$\mathcal{W}$	Background noise power
$P_{i,j}$	Received power at device j when device i is transmitting
P^T	Transmitted power
P^{Max}	Maximum transmission power at the MAC layer
$\Delta_{i,j}$	Euclidean distance between devices i and j
η	Path loss exponent for propagation in free space
λ	Signal's wavelenght
γ_j	Capture threshold of device j
$G(\mathcal{V}, \mathcal{E})$	Physical topology in MRMC-WMN
$\mathcal{V}$	Set of devices in MRMC-WMN
V	Number of devices in MRMC-WMN
$\mathcal{E}$	Set of edges in MRMC-WMN
E	Number of edges in MRMC-WMN
$\mathcal{K}$	Set of channels
K	Number of channels
Θ	Set of gateways
χ_p	Number of NICs for the device $p \in \mathcal{V}$
Λ	Maximum link utilization
Γ	Maximum routing path length
Ψ_{pq}^k	Nominal data rate of link e_{pq} on channel k
$\mathscr{S}$	Set of sender-receiver flow pairs
$\gamma_{s,r}$	Traffic load from the sender device s to the receiver device r
β	Interference threshold
$\mathcal{I}_p$	Distance in number of hops between p and the furthest device to p in $G(\mathcal{V}, \mathcal{E})$
$\Omega_{d,p}$	Set of d-hop neighbors to the device $p \in \mathcal{V}$, with $d = \{0, 1, \ldots, \mathcal{I}_p\}$
$\|\Omega_{d,p}\|$	Number of d-hop neighbors to the device $p \in \mathcal{V}$
h_{sr}^G	Minimum number of hops between devices s and r in $G(\mathcal{V}, \mathcal{E})$
Ξ	Set of network crews
Z	Number of network crews
$\mathscr{C}$	Set of ways to split the network
$\|\mathscr{C}\|$	Number of ways to split the network
R	Set of ranking functions
$\|R\|$	Number of ranking functions
U	Set of criteria to assign channels to unused NICs
$\|U\|$	Number of criteria to assign channels to unused NICs
$\mathcal{P}$	Set of JCAR sub-problems
$\mathcal{P}_{c,r}$	JCAR sub-problem with $c \in \mathscr{C}$ and $r \in R$

$\mathcal{P}_{c,r}(d)$	ILP formulation of $\mathcal{P}_{c,r}$ with $d = \{0, 1, \dots, \mathcal{I}_p - 1\}$
σ	Set of solutions of the JCAR problem
$\bar{\sigma}$	Best solution of the JCAR problem
$\sigma^u_{c,r}$	Solution of the JCAR problem with $c \in \mathscr{C}$, $r \in R$ and $u \in U$
Θ	Set of gateways
$\lvert\Theta\rvert$	Number of gateways
x^k_{pq}	Logical topology, 1 if device p communicates with device q over the channel k and 0 otherwise
$G^*(G)$	Logical topology graph, i.e. links for which $x^k_{pq} = 1$
y^k_p	Interface assignment, 1 if $\exists p \in \mathcal{V}$ and $e_{pq} \in \mathcal{E}$ such that $x^k_{pq} = 1$ and 0 otherwise.
$y^k_{g,pq}$	Interference, 1 if $\exists g, p \in \mathcal{V}$ and $e_{pq} \in \mathcal{E}$ such that $y^k_g = x^k_{pq} = 1$ over the same channel k and 0 otherwise.
ψ^k_{pq}	Effective capacity of a logical link
$\rho^k_{pq,sr}$	Binary routing, 1 if the traffic from device s to device r is being routed via link e_{pq} over channel k and 0 otherwise
λ^k_{pq}	Aggregate traffic, i.e. sum of the traffic on link e_{pq} over channel k
μ_{sr}	Path existence, 1 if a path exists between devices s and r in $G^*(G)$ and 0 otherwise
$h^{G^*}_{pq}$	Path length, number of hops between devices s and r in $G^*(G)$
D_{tot}	Total interference weighted on the traffic load and the distance from the gateway
$\mathcal{G} = \{\mathcal{N}, \mathcal{S}, \mathcal{U}\}$	Game
$\mathcal{N}$	Set of devices (Players in game theory)
$\mathcal{L}$	Set of cognitive devices (BSs and CPEs)
$\mathcal{M}$	Set of CPEs
$\mathcal{M}_i$	Set of CPEs belonging to BS $i \in \mathcal{N}$
$\mathcal{S}$	Set of Strategies
$\mathcal{S}_i$	Set of strategies for player $i \in \mathcal{N}$
$\mathcal{U}$	Set of Utility Functions
$\mathcal{K}_i$	Set of available channels for device $i \in \mathcal{N}$
BW_i	Backoff window for player $i \in \mathcal{N}$
BC_i	Backoff counter for player $i \in \mathcal{N}$
n_i	Number of WRANs sensed by a player i
$\mathcal{C}$	Coalition structure
C	Number of coalitions
$\mathcal{C}_p$	A specific coalition, with $p \in \{1, \dots, C\}$
C_p	Cardinality of $\mathcal{C}_p$
$v(\mathcal{C}_p)$	Coalition Value of $\mathcal{C}_p$
maxC	Maximum coalition dimension size
maxΔ	Maximum distance among devices belonging to the same coalition

$\Delta_{\mathcal{C}_p,i}$	Distance between the furthest player in coalition $\mathcal{C}_p$ and player i
$\mathcal{C}_{Old}$	Current coalition
$\mathcal{C}_{New}$	Future coalition
g_i	Gain for player $i \in \mathcal{N}$
o_i	Complexity for player $i \in \mathcal{N}$
$\mathscr{I}$	Set of Internet Service Providers $\mathscr{I} = \{ISP_0$, ISP_1, $ISP_2\}$
$\mathcal{B}$	Set of clusters
B	Number of clusters
$\mathcal{B}_p$	A cluster with $p = \{1, \ldots, B\}$
B_p	Number of C-MDs in the cluster $\mathcal{B}_p$
H_p	Cluster head of $\mathcal{B}_p$
$\mathcal{K}_{\mathcal{B}_p}$	Set of available channels of cluster $\mathcal{B}_p$
$K_{\mathcal{B}_p}$	Number of channels in $\mathcal{K}_{\mathcal{B}_p}$
$MaxTxRx$	Maximum number of consecutive CBMs transmitted by a C-MD on the same channel
$max\Delta$	Maximum number of hops admitted into a cluster
$max\Delta_{\mathcal{B}_p}$	Maximum number of hops in the cluster $\mathcal{B}_p$
T_{IR}	Timeout of the IR message
T_{JC}	Timeout of the JC message
T_{UP}	Timeout of the UP message
w_i	Weight of C-MD i
μ_u	Number of hops from the two furthest C-MDs in $\mathcal{B}_u$
ρ_i	Number of hops between i and the furthest C-MD in its cluster
$TxRange$	C-MD transmission range

You gotta find what you like and let it kill you.
Kinky Friedman

1 Introduction

In recent years there has been a wide proliferation of wireless applications and services which has led to the fundamental and challenging spectrum shortage problem. In this work we analyze the spectrum shortage problem under different scenarios proposing solutions for *traditional* technologies, that is *multi-radios multi-channels wireless mesh networks* (MRMC-WMNs), and *emerging* technologies, that is *cognitive networks* (CNs) and *cognitive wireless mesh networks* (C-WMNs).

Part I analyzes how to improve the usage of unlicensed wireless resources studying the join channel assignment and routing problem in MRMC-WMNs, then Part II and Part III analyze issues and solutions to opportunistically use licensed resources in CNs and C-WMNs, respectively. Conclusions and considerations on future directions are drawn in Part IV.

Wireless mesh networks (WMNs) are an emerging and recently widely available technology providing high-bandwidth networks in industrial and residential settings. The opportunity to equip a single mesh device (MD) with multiple radios is seen as a key to improve network performance [108]. In fact, by setting the radios on orthogonal (*non-overlapping*) channels, multiple packets can be transmitted over-the-air simultaneously without colliding with one another. Hence, introducing the so called MRMC-WMNs. Although equipping a MD with multiple radios is not an issue from a financial point of view, a conclusive solution concerning how to assign different channels to these devices has so far not been found. In the literature the way to assign channels to radios, in order to improve the aggregate throughput of the network, is termed *channel assignment* (CA). There are two research issues that need to be addressed when channel assignment algorithms in MRMC-WMNs are applied: routing and limited number of radios for each MD. In fact, depending on how channels are assigned to radios, different paths with different characteristics could be found. For these reasons, the processes of routing and channel assignment are very much inter-related and hence are considered jointly in the literature under the name of *Joint Channel Assignment and Routing* (JCAR) problems. The JCAR problem is $\mathcal{NP}$-hard, hence several heuristic approaches have been proposed in the literature. After a general overview on channel assignment approaches for MRMC-WMNs in Chapter 2, we review several JCAR solutions in Chapter 3 and then we compare their performance in Chapter 4 through an extensive simulation study. Conclusions are presented in Chapter 5.

Unfortunately even if different channels are assigned to multiple radios per device, MRMC-WMNs result in what is called *spectrum overcrowding*. In order to address the spectrum overcrowding problem, careful analysis on spectrum frequencies has been conducted. These studies have led into the identification of unlicensed and licensed channels which are differently utilized. In particular, licensed channels result to be surprisingly underutilized [37] compared to unlicensed channels. For this reason, the networking community is studying and addressing the resources' problem through the creation of CNs which are seen as the answer to the spectrum overcrowding.

Cognitive networks [71] have been proposed to have easily maintainable networks that are continuously improved and upgraded by relying as little as possible on human intervention. CNs opportunistically operate in licensed channels allocated to the TV broadcasting service, supplying to the unlicensed spectrum scarcity, and are characterized by a high level of flexibility given by their ability in sensing the current environment, planning for the future, making decisions and acting accordingly. CNs have been first thought as access networks which consider point-to-multipoint communication paradigms where a base station supports multiple end-users and provides access to the Internet in rural and remote areas. Then the concept of CN has been extended to industrial and residential settings and hence mesh capabilities have been added. In Part II and Part III, we address the main problems that afflict cognitive access (C-AN) and mesh (C-WMN) networks, respectively.

Part II is dedicated to the major challenge in implementing *cognitive access networks*, which is the coexistence among network devices of the same type, known as *self-coexistence* and which can be addressed as a channel assignment problem. The major difference between how to address the channel assignment problem in traditional wireless networks and cognitive networks is the time and space variability of the set of accessible channels for each device. In Chapter 6, we introduce general concepts on CNs illustrating terminology and challenges. Then in Chapter 7, we address the self-coexistence in C-ANs as a channel assignment problem using game theoretic approaches. First, we propose *Nora* which takes into consideration non-cooperation among self-interested cognitive devices, then we propose *HeCtor*, which considers the possibility to have groups of cooperative cognitive devices. *Nora* and *HeCtor* are evaluated in Chapter 8 along with algorithms from the literature. Conclusions are drawn in Chapter 9.

Concluding in Part III, we analyze issues related to *cognitive wireless mesh networks* identifying as the most important, the coordination problem among cognitive mesh devices (CMDs). In fact, the several functionalities needed in order to manage a CN (spectrum sensing, spectrum decision, and spectrum sharing) all require exchange of information and hence a coordination mechanism among devices. The coordination problem can be addressed assuming the existence of a centralized control entity or implementing a control message exchange mechanism. The existence of a centralized control entity in C-WMNs is not guaranteed because CMDs form a multi-hop backbone tier and hence there are not guarantees on the existence of a connection from each CMD to the centralized control entity. The control messages exchange, instead, is a suitable solution for C-WMNs but it is not without challenges particularly since CMDs experience spectrum variability over time and location and therefore a fixed control channel suitable for every CMD could not exist. We propose *Connor*, a clustering algorithm based on local common control channels, which addresses the coordination problem among CMDs as an exchange control messages problem. *Connor* does not require synchronization among CMDs and allows a fast re-clustering when changes occur in channel's occupancy by licensed users. In Chapter 10, we introduce general concepts and terminology of C-WMNs, and in Chapter 11 we analyze the coordination problem along with the description of *Connor*. The performance of *Connor* are presented in Chapter 12 and conclusions are drawn in Chapter 13.

Summarizing, spectrum shortage and overcrowding problems have been addressed in this work under several conditions and assumptions with the objective of handling the growing demand of wireless resources in rural, remote as well as industrial and residential areas.

Part I

Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks

I have not failed. I've just found 10,000 ways that won't work.

Thomas A. Edison

2 Multi-Radio Multi-Channel Wireless Mesh Networks

2.1 Introduction

Wireless Mesh Networks (WMNs) are an emerging and recently widely available technology providing high-bandwidth networks in industrial and residential settings. WMNs have been originally developed for military applications but, thanks to decrement in size, cost, and power requirements of mesh devices (MDs), they are becoming every day more popular as access network paradigms.

A WMN consists of *backbone devices* and *end-users* as shown in Fig. 2.1 (See [5] for a survey). As backbone devices we identify *mesh routers* and *mesh gateways*, which are fixed and form multi-hop wireless links between end-users and the Internet. Mesh gateways are mesh routers with Internet connectivity. End-user devices, on the other hand, are typically mobile or nomadic mesh clients. Each mesh client is connected to a backbone device in order to have its packets forwarded from/to the Internet. We focus on the backbone tier alone with the aim of achieving a higher performance using different chunks of the frequency spectrum to connect backbone devices among them in a multi-hop fashion.

A chunk of the frequency spectrum is commonly called *channel* and the problem to assign channels to backbone devices in order to avoid co-channel interference among them is referred as *channel assignment* (CA) problem. In recent years, the possibility of equipping a single backbone device with multiple radios has became a reality. Hence, a new class of WMNs using multiple channels and multiple radios has been identified and referred under the name of *multi-radio multi-channel wireless mesh network* (MRMC-WMN). MRMC-WMNs are characterized by the ability of transmitting on different channels and by backbone devices equipped with more than one radio each. In MRMC-WMN, the CA problem is seen as the way to allocate channels to radios in order to improve the aggregate throughput of the network.

2.1.1 Outline of the Chapter

The remaining of the chapter is organized as follows. Section 2.2 analyzes the channel assignment problem in MRMC-WMNs, Section 2.3 describes the most common interference models used in the literature in order to analyze interactions among

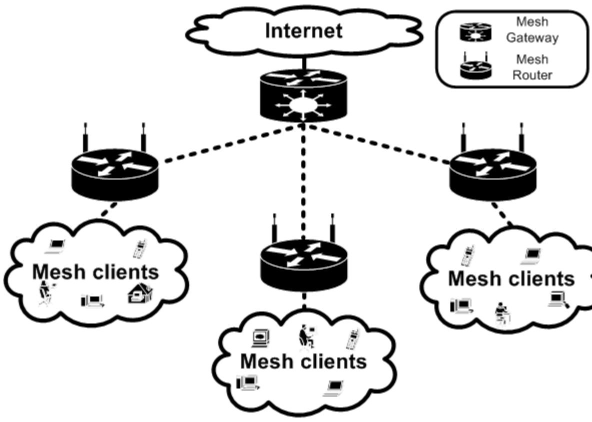

Figure 2.1: Wireless Mesh Network Architecture

wireless devices and Section 2.4 gives some terminology regarding network topologies.

2.2 The Channel Assignment Problem

The opportunity of equipping a single backbone device with multiple radios, called *Network Interface Card*s (NICs), is seen as a key to improve the network performance [108]. In fact, by setting the radios on orthogonal (*non-overlapping*) channels, multiple packets can be transmitted over-the-air simultaneously without colliding with one another. For example more frequency diversity can be allocated to those areas of the WMN that are expected to have a higher load. Although equipping a device with multiple NICs is not an issue from a financial point of view, a conclusive solution concerning how to assign different frequency bands to these devices has so far not been found. Moreover, to reduce contention and interference, randomly assigning channels to NICs or equipping nodes with a number of NICs equal to the number of allowable channels, are not efficient choices. A random assignment is not efficient because a node should minimize the number of neighbors sharing a common channel, while at the same time maintaining topological connectivity. Unfortunately, a random channel assignment cannot give these guarantees. A high number of radios, on the other hand, does not address the limited number of channels and routing problems. In fact, if the number of neighbors for a node is greater than the number of channels, then the node has to choose the appropriate neighbors that will share a common channel. In addition, equipping a node with a number of NICs equal to the number of channels does not solve the routing problem because it does not address how to choose a channel for a packet transmission.

A typical technology exploited to implement WMNs is the IEEE 802.11 standard where non-overlapping channels are assigned to NICs in order to enable transmissions. IEEE 802.11*b* and IEEE 802.11*g* standards [1] use the 2.4 GHz industrial, scientific and medical (ISM) band which is divided into 13 channels of which only 3 are orthogonal. Because of the frequency band choice, IEEE 802.11*b* and *g* equipment may occasionally suffer interference from microwave ovens, cordless telephones and Bluetooth devices with the result that they become crowded. Instead IEEE 802.11*a* standard [1] uses the 5 GHz unlicensed national information infrastructure (U-NII) band, which offers at least 12 non-overlapping channels rather than 3, thus providing a significant advantage. In practice the number of allowable channels is regulated by each country based on how the radio spectra are allocated to various services (See [1] for more details). To simplify our explanation, we consider 12 as a maximum number of non-overlapping channels. The IEEE 802.11 technology does not provide native support for multi-hop forwarding and channel assignment. In spite of this most WMNs are made up of off-the-shelf IEEE 802.11 devices, because of their widespread availability and very low cost. For this reason within the IEEE 802.11 working group, a task group *s* has been created in order to amend the standard and add the missing multi-hop functions. While the standardization process has not yet finished [2], the IEEE 802.11*s* draft is supported by a wide variety of industry leaders and is available as part of the 802.11 MAC layer in the most recent Linux kernels and FreeBSD.

2.2.1 Research Issues

There are two research issues that need to be addressed when CA algorithms are applied to MRMC-WMNs:

(i) Routing protocol, i.e. the process to select paths from sources to destinations.

(ii) Limited number of NICs for each router.

It is well-known that existing routing protocols for wired networks, such as the *Open Shortest Path First* (OSPF), are either inadequate or inefficient for WMNs [36]. Therefore, routing protocols from the domain of ad-hoc wireless networks, such as *Ad-hoc On-demand Distance Vector* (AODV) or *Optimized Link State Routing* (OLSR), are commonly adopted in WMNs with good results [36]. However when we change our focus from a WMN to MRMC-WMN more routing scalability and robustness are required [5]. In fact, depending on how channels are assigned to NICs, different paths with different characteristics could be found and the number of hops between two nodes can increase without any control. This behavior is caused by the fact that, in a multi-channel environment, two nodes can communicate only if they are in the transmission range of one another and they have at least one NIC tuned on a common frequency band. Additionally, channel assignment and routing both depend on the traffic load distribution. For these reasons, the processes of routing and channel assignment are very much inter-related and hence are considered jointly in MRMC-WMN under the name joint channel assignment and routing (JCAR) problem.

2.3 Interference Models

An interference model describes how devices influence each other, hence it is essential in order to model channel assignment algorithms among wireless devices. We identify two interference models: protocol and physical interference models.

The *protocol interference model* is a binary paradigm where interferences are considered on a pair basis and hence it provides a simple and easily tractable approach. However, the protocol interference model fails in properly capture the interference generated by the entire network. In fact, using a binary model it is possible associate at each device the exact number of overlapping competitors, but doing so, some solutions are ruled out. For example the protocol interference model do not consider solutions where two devices do not interfere, however if a third one is transmitting on the same channel, then the interference destroy the communication.

The *physical interference model* [110], instead, is a cumulative paradigm and hence capture how the entire network influence a single devices guaranteeing a more realistic model of the interactions among devices. Usually the physical interference model use the *Signal to Interference and Noise Ratio* (SINR) as in Eq. (2.1).

$$SINR_{i,j}(t_1,t_2) = \frac{P_{i,j}}{\sum\limits_{h \in \mathcal{T}(t_1,t_2) \setminus i} P_{h,j} + \mathcal{W}} \tag{2.1}$$

Here $\mathcal{T}(t_1,t_2)$ is the larger set of devices transmitting in the time interval $[t_1,t_2]$; $\mathcal{W}$ is the background noise power. The received power depends on the Euclidean distance $\Delta_{i,j}$, in meters, between devices i and j; the transmitted power P^T; the path loss exponent η; and, the signal's wavelength λ in meters:

$$P_{i,j} = \frac{P^T(\lambda/4\pi)^2}{\Delta_{i,j}^{\eta}}. \tag{2.2}$$

To effectively capture transmissions the SINR on the receiver device (j) has to be greater than or equal to its *capture threshold*, indicated by γ_j.

In an MRMC-WMN architecture, interference is often modeled with a *protocol interference model* to simply the JCAR problem. However, the protocol model does not correctly capture the cumulative nature of the interferences in a wireless environment, hence a physical interference model is preferred in order to consider the effect of cumulative interference from multiple devices transmitting at the same time on a single device.

Regarding to how devices can tolerate interference, a *protection* and a *pollution* viewpoints have been defined in the literature [98]. Protection means that a device can only operate in locations where it cannot generate any interference to other devices. In contrast, pollution allows interference under a given threshold to be non-disruptive. To establish these thresholds, a device needs to have knowledge regarding other devices, which may or may not be possible depending on its type and location.

2.4 Terminology of Network Topologies

Due to the nature of MRMC-WMNs, we distinguish between physical and logical topologies which consist of physical and logical links, respectively.

A *physical link* exists between any two devices if they are in the transmission range of one another. Two devices that share a physical link are called *one-hop neighbors* (or *neighbors* for short), while two devices that have a common neighbor, but are not neighbors themselves are called *two-hop neighbors*.

A *logical link* between two devices is characterized by the following properties: (i) there is a physical link between them, (ii) they have at least one NIC set to the same channel, and (iii) there is at least one traffic flow traversing them.

The set of physical links forms a *physical topology*, while logical links forms a *logical topology*.

Finally, we introduce the concept of connected network. A network is considered *connected* if at least one path exists between any pair of devices. The definition of *disconnected* network follows.

Do not wait for leaders; do it alone, person to person.

Mother Teresa

3 The Channel Assignment Problem

3.1 Introduction

The JCAR problem is seen as a key problem in the context of MRMC-WMNs, as also highlighted by the amount of work that has recently appeared in the literature.

Traditionally, the CA problem has been mapped on to the well-known graph coloring problem, i.e., to find the minimum number of colors assigned to devices in a graph such that two adjacent devices never have the same color [22]. While such an approach is important from a theoretical point of view, its applicability remains somewhat limited in practical terms, because of network-specific constraints and objective functions. For example, the number of colors that all the neighbors can have is limited by the number of NICs that a node has and it is difficult to capture a traffic load and/or cumulative interference models in a straightforward manner. Moreover the graph coloring problem as well as the JCAR problem are $\mathcal{NP}$-hard and there is no known algorithm to find the optimal solution in a reasonable amount of time (polynomial time with the network size) for non-trivial MRMC-WMNs.

In recent years, JCAR problems have been modeled following different approaches, which we can broadly classify into three categories.

(i) Optimization approaches: which are proved to be $\mathcal{NP}$-hard, then solved by relaxing constraints or using heuristic sub-optimal algorithms [78, 91].

(ii) Empirical approaches: whose effectiveness is typically verified through simulation [73, 93, 92, 12, 64, 107].

(iii) Mixed approaches: which are formulated as local optimization problems and then combined following empirical approaches [42].

3.1.1 Outline of the Chapter

The chapter is organized as follows. Section 3.2 introduces the system model, Section 3.3 illustrates mathematical optimization approaches proposed in the literature to address the JCAR problem, Section 3.4 describes several empirical algorithms and Section 3.5 proposes *G-PaMeLA*, a mixed approach.

3.2 System Model

System model and assumptions described in following are typical of JCAR problems presented in the literature, where has been assumed that the centralized entity running the CA algorithm knows the following configuration parameters.

(i) Physical topology (see Section 2.4). It is represented as a graph $G(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of devices and $\mathcal{E}$ is the set of unidirectional physical links. We indicate with V and E the cardinality of sets $\mathcal{V}$ and $\mathcal{E}$, respectively. Hereafter h_{sr}^{G} indicates the shortest path, in a number of hops, from device s to device r in $G(\mathcal{V}, \mathcal{E})$.

(ii) The number of NICs of device p, say χ_p. We assume $\chi_1 = \chi_2 = \cdots = \chi_V$ to simplify the notation but without loss of generality, and this number is indicated with χ. Thus, the set of allowable channels is $\mathcal{K} = \{1, 2, \cdots, K\}$. As previously mentioned, the number of allowable channels may be different in different countries. For sake of simplicity, we consider $K \leq 12$. In fact, we have to add to the constraint given by each country, environmental constraints where some channels could be used by neighboring networks or end-users. For these reasons we analyze the behavior of CA algorithms in configurations with different numbers of allowable channels. The limited number of channels implies that, with the exception of very small networks, some logical links must be assigned to the same channel, i.e. these links cannot be simultaneously active. We did not analyze the link scheduling problem in MRMC-WMNs [68] but we are considering to include this feature in a future study.

(iii) Received power, channel rates, and traffic loads. The concept of received power $P_{p,q}$ is defined in Eq.(2.2) and is closely related to the interference as explained in Section 2.3. The channel rate Ψ_{pq}^{k} represents the nominal data rate of the link e_{pq} on channel $k \in \mathcal{K}$. For instance, the IEEE 802.11a standard uses a 52-subcarrier orthogonal frequency-division multiplexing (OFDM) with a maximum raw data rate of 54 Mbps, which could be reduced to 48, 36, 24, 18, 12, 9 or 6 Mbps if required. With regard to both the received powers and the channel rates, WMN devices are static, hence channel conditions are quite stable [21]. An estimation of the physical layer status was investigated in [60] and it is not considered further in this study. Lastly, the set $\mathscr{S}$ contains all the sender-receiver flow pairs and its cardinality is the number of traffic flows through the network. Each traffic load from any sender device s to any receiver device r is $\gamma_{s,r}$ and is either available as *a priori* knowledge, based on historical data, or estimated while the WMN is operating.

We further define $\Omega_{d,p}$ as the set of devices at distance d, in a number of hops, from the device p taking into account $G(\mathcal{V}, \mathcal{E})$. If $|\Omega_{0,\theta}| = 1$ then the unique element $\omega_{0,\theta}$ in this set is the gateway, i.e. $\omega_{0,\theta} = \theta$, and each element $\omega_{1,\theta} \in \Omega_{1,\theta}$ is a one-hop neighbor to the gateway (*sub-gateway*). Follow that $\Omega_{1,\theta}$ is the set of sub-gateways. Notations used in the JCAR problems illustrated in this Part are shown in Table 3.1.

Table 3.1: JCAR Notation.

Symbol	Value
$\mathcal{V}$	Set of devices.
$\mathcal{E}$	Set of edges.
$G(\mathcal{V}, \mathcal{E})$	Physical topology.
$\mathcal{K}$	Set of channels.
Θ	Set of gateways.
χ_p	Number of NICs for the device $p \in \mathcal{V}$.
Λ	Maximum link utilization.
Γ	Maximum routing path length.
$P_{p,q}$	Received power at q when p is transmitting.
Ψ^k_{pq}	Nominal data rate of link e_{pq} on channel k.
$\mathscr{S}$	Set of sender-receiver flow pairs.
$\gamma_{s,r}$	Traffic load from the sender device s to the receiver device r.
β	Interference threshold.
$\mathcal{I}_p$	Distance in number of hops between p and the furthest device to p.
$\Omega_{d,p}$	Set of d-hop neighbors to the device $p \in \mathcal{V}$, with $d = \{0, 1, \ldots, \mathcal{I}_p\}$.
h^G_{sr}	Minimum number of hops between devices s and r in $G(\mathcal{V}, \mathcal{E})$.

3.3 Optimization Approaches

JCAR problems formulated as mathematical optimizations have the objective of finding the the best solution from a set of available alternatives. In this context we summarize two works extracted from the literature [78, 91].

Mohsenian et al. [78] formulates the JCAR problem as an ILP problem and as a heuristic algorithm but the results are only shown for the heuristic formulation due to the inherent complexity of the JCAR problem. Hence, in the heuristic algorithm they trade the accuracy of the solution for a (much) faster execution time. Their algorithm is based on randomly choosing the initial solution and then refining it within a limited number of configurable steps. Constraints and objective function of the problem in [78] are adapted to the local sub-problem summarized in Section 3.5.

Ramanathan et al. [91] proposed an unified framework to efficiently assign channels to devices or links in order to achieve the most efficient spatial reuse. Their formulation is based on a graph coloring problem, hence the execution time of their algorithms is high. For this reason they also proposed distributed versions.

3.4 Empirical Approaches

Empirical algorithms are formulated based on information gained by means of observations, experiences and experiments.

Marina et al. [73] proposed the *Connected Low Interference Channel Assignment* (CLICA), which is based on the use of a conflict graph and a protocol interference model. Kyasanur et al. in [64] proposed a JCAR algorithm suitable to be performed by a central server that periodically and dynamically collects channel interference information. However, [73] and [64] do not consider the traffic load on links.

Skalli et al. [107] formulated a fixed, rank based, polynomial time, greedy algorithm for centralized channel assignment where the rank of each device is computed based on its link traffic characteristics. Unfortunately this technique is difficult to apply if channel assignment and routing are considered jointly. In fact, the algorithm in [107] is based on a traffic matrix which assumes a priori knowledge of the routing algorithm. Clearly, when channel assignment and routing are done jointly, the traffic matrix is not known a priori.

The following solutions have the same objective as G-PaMeLA, hence they are evaluated for comparison purposes in the performance analysis in Chapter 4.

First, Raniwala et al. [93] proposed a centralized heuristic *Load-Aware joint Channel Assignment and routing algorithm* (LACA), which is specifically used for wireless Internet access applications. Given the set of initial link flow rates, LACA assigns channels in the attempt to have a proportional relation between flow rate and available bandwidth on each link. The available bandwidth values are estimated as a fraction of the link capacity and are used as inputs to the routing algorithm, which computes the shortest path for every flow. The resulting flow mapping on each link is used as a link flow rate for the next iteration, in which a new channel assignment is computed. The algorithm in [93] does not tie to any specific routing mechanism.

The same authors also proposed *Hyacinth* [92], which is constructed with a multiple spanning tree-based load balancing routing algorithm that can be adapted dynamically to a traffic load. The channel assignment problem is divided into two problems, i.e. neighbor-to-interface and an interface-to-channel binding problem. In this logical tree topology, each gateway is a root and each router uses an up-NIC to exclusively connect to its parent, and uses several down-NICs to connect to its children. Each parent router provides Internet connectivity to its children (routers), that is each wireless mesh router can access the Internet through the shortest available routing path. In Hyacinth, each router allocates the channels that are the least used by its neighboring routers to down-NICs. The channel assignment to devices positioned higher in the tree affects all devices lower in the tree hierarchy thus creating a non-robust CA due to *ripple-effects*.

Finally, in [12] the authors proposed the *Flow-based Channel and Rate Assignment* (FCRA) algorithm. FCRA is a centralized channel and rate assignment algorithm, which starts from a network mapped on an individual channel, which then improves the performance by adding different channels where possible. The FCRA algorithm does not require knowledge of the traffic demands and provides both a channel and a transmission rate for each link, taking into account the network-wide effect of such a choice.

3.5 Mixed Approach: G-PaMeLA

We propose our mixed approach to the JCAR problem, which is a *divide-and-conquer* scheme called *Generalized Partitioned Mesh network traffic and interference aware channeL Assignment* (G-PaMeLA).

The core of our scheme consists of solving a sequence of sub-problems in a given order and combining them using a post-processing procedure. Each sub-problem is much simpler than the global JCAR problem, because only the local constraints on interference are tested. A sub-problem is solved using a hybrid approach between a branch-and-bound and cutting plane, i.e. the *branch-and-cut* method [25]. The latter method is used to solve the ILP problems with the regular simplex algorithm, where some unknowns are restricted to integer values. In addition, we take into

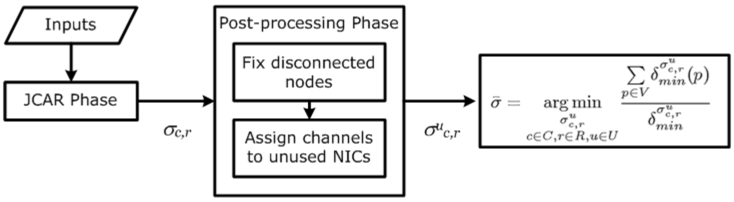

Figure 3.1: G-PaMeLA flow-chart.

Table 3.2: G-PaMeLA Notation.

Symbol	Value
Ξ	Set of network crews. Different crews can be defined considering topology and traffic profile.
$\mathscr{C}$	Set of ways to split the network. Each $c \in \mathscr{C}$ is defined using the network crews $\xi_z \in \Xi$. $z = \{0, 1, \ldots, \lvert\Omega_{1,\theta}\rvert\}$ and $\lvert\mathscr{C}\rvert = 1$ if the topology is unknown.
R	Set of ranking functions. $\lvert R\rvert = 1$ if the topology is unknown.
U	Set of criteria to assign channels to unused NICs.
$\mathcal{P}$	Set of JCAR sub-problems. $\mathcal{P}_{c,r} \in \mathcal{P}$ is a specific instance where JCAR, network crew c and ranking function r constraints are considered jointly.
σ	Set of solutions of the JCAR problem. $\sigma_{c,r} \in \sigma$ is the solution of the problem $\mathcal{P}_{c,r}$. A particular solution obtained after the post-processing phase is indicated by $\sigma^{u}_{c,r}$ with $u \in U$.

consideration the physical interference model (see Section 2.3) to address interference problem among devices.

We assume that a centralized entity, co-located with the gateway, exists and we dedicate the use of a NIC for control purpose. Hence, we assume the existence of a control channel common to all the backbone devices. Periodically, the gateway uses G-PaMeLA to assign channels and to determine the routing paths in the MRMC-WMN. In addition, the gateway is responsible for disseminating the updated channel assignment and routing to all devices using the control channel. The procedure is provided with input on the configuration parameters and the current status of the MRMC-WMN, which can be retrieved by means of a network management protocol running in the MRMC-WMN. The network functions that collect data from devices to the centralized entity and enforce channel assignments and routing that it produces, are outside the scope of our work, which focuses only on the algorithm that it runs. Thanks to the load balancing property of G-PaMeLA, we assume that the MRMC-WMN also acts appropriately if the traffic load changes, in fact in each link there is room free for additional flows.

We consider a set of gateways Θ to the Internet. Clearly we can trace the problem back to the case $\lvert\Theta\rvert = 1$ in fact if $\lvert\Theta\rvert > 1$ then the overall JCAR problem can be divided into $\lvert\Theta\rvert$ sub-JCAR problems where each takes into account the channel assignment made by the others. The devices are assigned to each sub-JCAR problem based on a combination of factors, including the distance from the gateway and the traffic load. Hereafter, to simplify the explanation and without loss of generality we consider $\lvert\Theta\rvert = 1$ and we indicate the gateway as $\theta \in \Theta$.

The divide-and-conquer strategy, on which G-PaMeLA is based on, consists of breaking one problem into smaller, more manageable sub-problems, and then taking control of these sub-problems one by one. This strategy is therefore a powerful tool for conceptually solving complex problems like the JCAR problem.

The technique used to break down the JCAR problem into sub-problems is managed from the first phase of G-PaMeLA, called JCAR phase. The second phase, called post-processing phase, instead combines the outcomes of the related sub-problems. The best solution $\bar{\sigma}$ is selected from all the solutions $\sigma_{c,r}^{u}$ according to a max-min fairness criterion combined with a load-aware interference objective. The G-PaMeLA flow chart is shown in Fig. 3.1 and the notations used throughout this Section are summarized in Tables 3.1 and 3.2. We now provide a formal description of the two phases separately.

3.5.1 JCAR phase

The JCAR phase uses an inner core procedure, which finds the optimal solution of a sequence of sub-problems which are formulated as an ILP problem and are solved following the order given by a *ranking function*. The output of each sub-problem defines the channel assignment and routing local to the devices associated with this sub-problem.

The divide-and-conquer approach entails splitting the overall problem into several sub-problems, thus in the JCAR phase we divide the devices into sets. Devices in the same set are characterized by a common property which could be, for example, the number of hops to the gateway. In this case the number of sets is equal to $\mathcal{I}_\theta$. As previously explained, we can then reduce the number of JCAR sub-problems to $\mathcal{I}_\theta - 1$. The time needed to solve the global ILP problem with G-PaMeLA is given by the sum of the time required to solve the $\mathcal{I}_\theta - 1$ sub-problems. No additional time is required to unify the sub-problems because each sub-problem is solved under the constraints given by the sub-problems already solved. To decide the set of devices associated with a sub-problem, the JCAR phase defines a property that each device satisfies. If the centralized entity performing G-PaMeLA is not aware of the physical topology pattern (*unknown topology*), then the property is defined as the number of hops between the device and the gateway and therefore the number of ranking functions is equal to 1. That is, we solve an ILP problem for each $\Omega_{d,\theta}$ where $d = 0, 1, \ldots, \mathcal{I}_\theta$. On the other hand, if the topology refers to well-known patterns such as grids or stars (*known topology*), the JCAR phase can define different properties and therefore several ranking functions. The details driving the choice of using a ranking function are described in the following.

Due to the limited number of constraints, the sub-problems are solved optimally in a shortened amount of time with standard solvers, e.g., using branch-and-cut techniques [70], making our JCAR solution feasible for an operational network. Moreover, multiple instances of the inner core procedure could be run, each one enforcing different routing constraints via so-called *network crews*. However, if the topology is unknown the number of combinations of network crews is $|\mathscr{C}| = 1$ because the construction of each crew is based only on the number of hops. The crews' construction and motivations are later detailed.

At the end of the JCAR phase we obtain a set σ of JCAR solutions, one for each ILP problem in the set $\mathcal{P}$. Let us indicate with $\sigma_{c,r} \in \sigma$ the solution of the JCAR problem $\mathcal{P}_{c,r} \in \mathcal{P}$ solved by following the ranking order $r \in R$ and under the constraints given by $c \in \mathscr{C}$.

We conclude this section by formulating the ILP problem and explaining working

Algorithm 1 G-PaMeLA JCAR Phase: pseudo-code.

1: **procedure** JCAR
2: let $\mathcal{P}_0$ be a JCAR problem with basic constraints and $\mathcal{P}_{c,r}(-1) = \emptyset$
3: compute $r \in R$ and $c \in \mathscr{C}$
4: **for** $c \in \mathscr{C}$ **do**
5: $\mathcal{P}_c = \mathcal{P}_0 +$ constraints due to c
6: **for** $r \in R$ **do**
7: **for all** $d = \{0, 1, \ldots, \mathcal{I}_\theta - 1\}$ ordered as r **do**
8: $\sigma_{c,r}(d) =$ solve $\{\mathcal{P}_{c,r}(d)\}$ under $\mathcal{P}_{c,r}(d-1)$
9: **end for**
10: **end for**
11: **end for**
12: $\sigma = \bigcup_{c \in \mathscr{C}, r \in R} \sigma_{c,r}(\mathcal{I}_\theta - 1)$
13: **end procedure**

Table 3.3: JCAR Problem Variables.

Symbol	Name	Definition
x^k_{pq}	Logical topology	It is 1 if device p communicates with device q over the channel k, 0 otherwise.
$G^*(G)$	Logical topology graph	It is derived from the graph $G(\mathcal{V}, \mathcal{E})$ and consists of all the links for which $x^k_{pq} = 1$.
y^k_p	Interface assignment	It is 1 if $\exists p \in \mathcal{V}$ and $e_{pq} \in \mathcal{E}$ such that $x^k_{pq} = 1$, 0 otherwise.
$y^k_{g,pq}$	Interference	It is 1 if $\exists g, p \in \mathcal{V}$ and $e_{pq} \in \mathcal{E}$ such that $y^k_g = x^k_{pq} = 1$ over the same channel k, 0 otherwise.
ψ^k_{pq}	Effective capacity of a logical link	It depends on the traffic that crosses the link $e_{pq} \in \mathcal{E}$ and on the number of NICs over channel k.
$\rho^k_{pq,sr}$	Binary routing	It is 1 if the traffic from device s to device r is being routed via link e_{pq} over channel k, 0 otherwise.
λ^k_{pq}	Aggregate traffic	Sum of the traffic on link e_{pq} over channel k.
μ_{sr}	Path existence	It is 1 if a path exists between devices s and r in $G^*(G)$, 0 otherwise.
$h^{G^*}_{pq}$	Path length	Number of hops between devices s and r in $G^*(G)$.

variables, tunable parameters and constraints. For the sake of brevity, we report the workflow as a pseudo-code in Algorithm 1 using mathematical notations.

An ILP instance of the problem $\mathcal{P}_{c,r}$, indicated as $\mathcal{P}_{c,r}(d)$ with $d = \{0, 1, \ldots, \mathcal{I}_\theta - 1\}$, is formulated including the routing constraints from the network crew c (line 5), and all the constraints on routing, capacity, and interference that affect the one-hop neighborhood of the $\Omega_{d,\theta}$ set. Additionally, $\mathcal{P}_{c,r}(d)$ needs to take into consideration the selected paths for sets that have already been solved (line 8).

The solution $\sigma_{c,r}$ of the ILP problem $\mathcal{P}_{c,r}$ is finally obtained by putting together all the channel assignment and routing choices in all the instances. The maximum number of JCAR solutions at the end of the JCAR phase, and passed to the post-processing phase, is $|\mathscr{C}| \times |R|$, which is 1 if the topology is unknown.

Table 3.3 summarizes the many working variables used by the ILP problem, whereas the tunable parameters are defined as follows:

- Λ is a value in the range $(0, 1]$ which represents an upper bound on the expected link utilization. In fact, in existing networks the channel rate cannot be used entirely due to the Medium Access Control (MAC) and the Physical layer's overhead, including headers, collisions, inter-frame spaces, preambles, and antenna switching gaps. With the help of Λ we force G-PaMeLA to only use a fraction of the nominal data rate Ψ^k_{pq}, $\forall e_{pq} \in \mathcal{E}$ and $\forall k \in \mathcal{K}$, by leaving room for any such overheads.

- Γ is an upper bound on the routing path length and its value is greater or equal to 1. With the help of Γ we force the inclusion of those routing paths that are longer, in terms of the number of hops, than the shortest ones computed in $G(\mathcal{V}, \mathcal{E})$. Let us take a binary variable μ_{sr} that is 1 if device s can reach device r in the logical topology, 0 otherwise. It is clear that μ_{sr} is always equal to 1 in the physical topology. Let $h^{G^*}_{sr}$ be the path length between devices s and r in $G^*(G)$. If $\Gamma = 1.5$ and $h^{G}_{sr} = 4$ then all the paths where $h^{G^*}_{sr} \leq 6$ are considered eligible. As a special case, if $\Gamma = 1$ only paths of the minimum length are allowed.

- β represents the resilience to interference, which in turn depends on path-loss and all the environment variables analyzed in Eq. (2.1). With the help of β, we decide how strict G-PaMeLA is with respect to the assumption that given a device p the devices in the set $\bigcup_{d=0,\ldots,\mathcal{I}_p} \Omega_{d,p}$ may or may not interfere with transmissions to p. Decreasing β means increasing the spatial re-use and the number of concurrent transmissions in the network and hence interference. It is unlikely that the non-predictable behavior forces us to make assumptions as in the ILP formulation in Fig. 3.2.

Λ and Γ have already been proposed in the literature (e.g., [78]), β instead, is a peculiarity of G-PaMeLA, in fact it is closely related to the physical interference model used.

Figure 3.2: G-PaMeLA: ILP formulation of $\mathcal{P}_{c,r}(d)$.

find

$$\max \delta_{min}(t) \quad \forall t \in \Omega_{d,\theta} \tag{3.1}$$

$$\delta_{min}(t) = \min_{\substack{e_{pq} \in \mathcal{E}_d; \\ k \in \mathcal{K};\ x_{pq}^k = 1}} (\Lambda \cdot \Psi_{pq}^k - \lambda_{pq}^k), \quad \forall e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.2}$$

Channel allocation constraints

$$x_{pq}^k = x_{qp}^k, \quad \forall e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.3}$$

$$\sum_{k \in \mathcal{K}} y_p^k \leq \chi, \quad \forall p \in \bigcup_{i=d,d+1,\ldots,\mathcal{I}_\theta - 1} \Omega_{i,\theta} \tag{3.4}$$

$$y_p^k \leq \sum_{e_{pq} \in \mathcal{E}_d} x_{pq}^k, \quad \forall p \in \bigcup_{i=d,d+1,\ldots,\mathcal{I}_\theta - 1} \Omega_{i,\theta};\ \forall k \in \mathcal{K} \tag{3.5}$$

$$x_{pq}^k \leq y_p^k, \quad \forall e_{pq} \in \mathcal{E} : p \in \bigcup_{i=d,d+1,\ldots,\mathcal{I}_\theta - 1} \Omega_{i,\theta};\ \forall k \in \mathcal{K} \tag{3.6}$$

$$\sum_{k \in \mathcal{K}} x_{pq}^k \leq 1, \quad \forall e_{pq} \in \mathcal{E}_d \tag{3.7}$$

Capacity constraints

$$\psi_{pq}^k \leq x_{pq}^k \cdot \Psi_{pq}^k, \quad \forall e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.8}$$

$$\sum_{e_{pq} \in \mathcal{E}_d} \frac{\psi_{pq}^k}{\Psi_{pq}^k} + \sum_{e_{qp} \in \mathcal{E}_d} \frac{\psi_{qp}^k}{\Psi_{qp}^k} \leq 1, \quad \forall k \in \mathcal{K} \tag{3.9}$$

Traffic constraints

$$\lambda_{pq}^k = \sum_{(s,r) \in \mathscr{S}} \rho_{pq,sr}^k \cdot \gamma_{s,r}, \quad \forall e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.10}$$

$$\lambda_{pq}^k \leq \Lambda \cdot \Psi_{pq}^k, \quad \forall e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.11}$$

Interference constraints

$$y_{g,pq}^k \leq \sum_{\substack{e_{gh} \in \mathcal{E}: \\ h \neq p, h \neq q}} x_{gh}^k, \quad \forall g \in \mathcal{V} : e_{gp} \in \mathcal{E} \wedge g \neq q;\ \forall e_{pq} \in \mathcal{E} : p \in \Omega_{d,\theta} \vee q \in \Omega_{d,\theta};\ \forall k \in \mathcal{K} \tag{3.12}$$

$$x_{gh}^k \leq y_{g,pq}^k, \quad \forall e_{gh} \in \mathcal{E} : h \backslash \{g,p,q\};\ \forall e_{pq} \in \mathcal{E}_d : e_{gp} \in \mathcal{E} \wedge g \neq p, p \in \Omega_{d,\theta} \vee q \in \Omega_{d,\theta};\ \forall k \in \mathcal{K} \tag{3.13}$$

$$\Psi_{pq} \geq \beta \cdot \sum_{\substack{e_{gp} \in \mathcal{E}: \\ g \neq p}} \Psi_{gq} \cdot (y_{g,pq}^k + x_{pq}^k - 1), \quad \forall p,q \in \mathcal{V};\ e_{pq} \in \mathcal{E}_d : p \in \Omega_{d,\theta} \vee q \in \Omega_{d,\theta};\ \forall k \in \mathcal{K} \tag{3.14}$$

Routing constraints

$$\sum_{k\in\mathcal{K}} \rho^k_{pq,sr} \leq 1, \quad \forall p,q \in \mathcal{V};\ \forall (s,r) \in \mathscr{S};\ e_{pq} \in \mathcal{E}_d : p \in \Omega_{d,\theta} \vee q \in \Omega_{d,\theta} \tag{3.15}$$

$$\rho^k_{pq,sr} = \rho^k_{qp,sr}, \quad \forall (s,r) \in \mathscr{S};\ e_{pq} \in \mathcal{E}_d;\ \forall k \in \mathcal{K} \tag{3.16}$$

$$\sum_{e_{pq}\in\mathcal{E}_d} \sum_{k\in\mathcal{K}} \rho^k_{pq,sr} \cdot \gamma_{s,r} - \sum_{e_{qp}\in\mathcal{E}_d} \sum_{k\in\mathcal{K}} \rho^k_{qp,sr} \cdot \gamma_{s,r} = = \begin{cases} \gamma_{s,r}, & ifs = p \\ -\gamma_{s,r}, & ifr = p \\ 0, & \text{otherwise} \end{cases} \qquad \forall p \in \bigcup_{\substack{i=d,d+1,\dots,\\ \mathcal{I}_\theta - 1}} \Omega_{i,\theta} \tag{3.17}$$

$$\sum_{e_{pq}\in\mathcal{E}_d} \sum_{k\in\mathcal{K}} \rho^k_{pq,sr} \leq \Gamma \cdot h^G_{sr}, \quad \forall (s,r) \in \mathscr{S} \tag{3.18}$$

Figure 3.2 shows the ILP formulation used in the JCAR phase of G-PaMeLA, where to split the global problem each ILP sub-problem acts on a subset of edges $\mathcal{E}_d \subseteq \mathcal{E}$ as defined in Eq. (3.21).

We now discuss the channel allocation and routing constraints in Fig. 3.2:

Eq. (3.3) expresses bi-directional links;

Eq. (3.4) limits the number of NICs per device;

Eq. (3.5) *and* Eq. (3.6) create the logical topology $G^*(G)$ [78];

Eq. (3.7) states that each device can only communicate with a one-hop neighbor via a single NIC. This limit is relaxed in the post-processing phase;

Eq. (3.8) limits the effective capacity, defined in Table 3.3, to the channel rate;

Eq. (3.9) limits the use, defined as the fraction of time that is spent for transmission, of any logical link [78];

Eq. (3.10) defines the aggregate traffic, as in Table 3.3;

Eq. (3.11) limits the aggregated traffic based on Λ;

Eq. (3.12) *and* Eq. (3.13) define the variable, $y^k_{g,pq}$ as in Table 3.3;

Eq. (3.14) limits the spatial re-use, according to the definition of β;

Eq. (3.15) *and* Eq. (3.17) define the routing. These are well-known in the literature [78, 58];

Eq. (3.16) forces bi-directional traffic to follow the same path. This constraint has been added because empirical evidence suggests that this leads fewer channels per device, thus yielding better solutions. This constraint is relaxed in the post-processing phase;

Eq. (3.18) limits the path length based on Γ;

Eq. (3.2) expresses the objective function that was first defined in [78], though in a slightly different formulation. This constraint represents an attempt to reach localized per-device max-min fairness, since it maximizes the minimum difference between the channel rate and the traffic load across all channels and all links around a device;

Eq. (3.1) is the overall objective function. δ_{min} is derived as the maximum over all $\delta_{min}(t)$, $\forall t \in \Omega_{d,\theta}$.

Ranking Functions

A ranking function $r \in R$ is a criterion to sort the JCAR sub-problems defined by G-PaMeLA. The first JCAR sub-problem to be solved is the one related to the gateway θ , i.e. $\Omega_{0,\theta}$, then the other sub-problems are solved in increasing order of the number of hops to the gateway. The rationale is that a device closer to the gateway is more critical than a peripheral one, since it relays more traffic. Therefore, such a device should be considered at an early stage of channel assignment and routing. Of course, different policies could be followed, e.g. an area with a higher load. Moreover, it is clear that it is not necessary to run the ILP problem for each $\Omega_{d,\theta}$ with $d = \{0, 1, \ldots, \mathcal{I}_\theta\}$ but $\mathcal{I}_\theta - 1$ are enough to cover all the edges. This consideration helps to speed up the process.

If the topology in unknown $|R| = 1$ but if the topology is, for example, a grid or a binary tree (i.e. it has a well-known pattern), it is possible to have more ranking functions customized per network topology and traffic load.

Network Crews

Depending on the network topology different network crews are definable. A network crew is a sub-set of $\mathcal{V}$ and c is the union of crews that adds the following constraint to the JCAR problem: all the devices in a crew can only reach the gateway through the respective sub-gateway $\omega_{1,\theta} \in \Omega_{1,\theta}$. The number of crews is equal to $|\Omega_{1,\theta}|$, the number of sub-gateways, if the topology is unknown but could be customized if the network's topology pattern is well-known. If the number of crews is customized the dimension of the set $\mathscr{C}$ also increases. $\mathscr{C}$ contains the combinations that satisfy:

$$\bigcap_z \xi_z = \emptyset \text{ and } \bigcup_z \xi_z \subseteq \mathcal{V}, \tag{3.19}$$

where ξ_z is a specific network crew and the number of network crews generated from the Algorithm 2 changes depending on the physical network topology pattern. The idea behind using $c \in \mathscr{C}$ is to balance traffic among the gateway's links, which are likely to become congested during the network's operation.

Figure 3.6 shows the crews and the resulting set $\mathscr{C}$ customized for a square-grid topology. In this example $z = \{1, \ldots, 6\}$ because it is possible to customize the number of crews by adding the devices that are at the same number of hops as more than one set, but in general $z = \{1, \ldots, |\Omega_{1,\theta}|\}$. Note that the gateway does not belong to any crew. Also, if there are devices besides the gateway, that do not belong to any crew, traffic flows originating from these devices can follow an arbitrary path.

The procedure describing the crew formation is in Algorithm 2 where: (i) the network crews are created and each one is associated with a sub-gateway (lines 17-19); (ii) a device $\omega_{d,\theta}$ in each set $\Omega_{d,\theta}$ is associated with a crew to obtain crews with equal cardinality (lines 20-31). The resulting set Ξ contains all the crews (line 33).

Algorithm 2 Network Crews: pseudo-code.

procedure NETWORK CREWS
// Z *represents the number of crews and could be customized.*
// $Z = |\Omega_{1,\theta}|$ *if the topology is unknown.*
  for all $z = \{0, \dots, Z\}$ **do** // *create network crews and assign a sub-gateway to each one.*
    $\xi_z = \xi_z \cup \omega_{1,\theta}$
  end for
  for $d = \{2, \dots, \mathcal{I}_\theta\}$ **do** // d *represents the number of hops to* θ
    for all $\omega_{d,\theta} \in \Omega_{d,\theta}$ considered all together **do**
// Ω_{1,ξ_z} *is the set of devices at distance* 1 *to at least one device in* ξ_z
      if $\omega_{d,\theta} \in \Omega_{1,\xi_z}$ for 1 device **then**
        $\xi_z = \xi_z \cup \omega_{d,\theta}$
      end if
      if $\omega_{d,\theta} \in \Omega_{1,\xi_z}$ for more than 1 device **then**
        if $|\xi_0| = \dots = |\xi_Z|$ **then**
          $\xi_z = \xi_z \cup \omega_{d,\theta}$
        end if
      end if
    end for
  end for
  Obtain: $\Xi = \{\xi_0, \dots, \xi_Z\}$ // Ξ *is the set contains all the crews.*
end procedure

Complexity of the Channel Assignment Problem

The relation V^K is well-known and expresses the complexity of the CA problem. The complexity of a channel assignment problem increases exponentially with the number of channels. However, estimating the complexity of our JCAR problem is trickier because it also depends on the number of NICs for each device and on the routing path length. The routing complexity can be formulated using the *Prim's algorithm* as $E \cdot \log V$. Eq. (3.20) expresses the complexity of a general JCAR problem subjected to the same G-PaMeLA conditions.

$$\mathcal{O}\,(V \cdot \chi)^K + \mathcal{O}\,(\;(E \cdot \chi) \cdot \log(V \cdot \chi)\;)\,, \tag{3.20}$$

where $V \cdot \chi$ expresses the complexity given by the number of NICs, and $E \cdot \chi$ is the number of edges in the network considering that each edge could be on χ channels.

As Eq. (3.20) shows, the time needed to solve a single sub-problem depends on the dimension of the set $\Omega_{d,\theta}$ and on $\mathcal{E}_d$ which is defined in Eq. (3.21) as the set of ingoing and outgoing physical links to/from $\Omega_{d,\theta}$.

$$\mathcal{E}_d = \left\{ \bigcup_{\substack{p \in \Omega_{d,\theta} \\ q \in \mathcal{V}: e_{pq} \in \mathcal{E}}} e_{pq} \right\} \cup \left\{ \bigcup_{\substack{p,q \in \bigcup_{i=d+1,\dots,\mathcal{I}_\theta} \Omega_{d,\theta}: \\ e_{pq} \in \mathcal{E} \wedge p,q \notin \Omega_{d,\theta}}} e_{pq} \right\}, \tag{3.21}$$

where the first part is the set of links ingoing or outgoing to/from the devices in the set $\Omega_{d,\theta}$, and the second part takes into consideration the physical links for which a channel assignment has not yet been made. The dimensions of the set of ingoing and outgoing edges in turn depend on the dimensions of the neighborhood of $\Omega_{d,\theta}$, indicated by $\Omega_{1,\Omega_{d,\theta}} = \Omega_{d-1,\theta} + \Omega_{d+1,\theta}$. Therefore, the routing complexity is reduced due to both network crews and previously solved sets. For these reasons we can express the complexity of our JCAR problem as in Eq. (3.22).

$$\mathcal{O}\left(\sum_{d=\{0,\ldots,\mathcal{I}_\theta-1\}}(|\Omega_{d,\theta}|\cdot\chi)^K\right)+\mathcal{O}\left(\sum_{d=\{0,\ldots,\mathcal{I}_\theta-1\}}(E_d\cdot\chi)\cdot\log(|\Omega_{d,\theta}|\cdot\chi)\right) \tag{3.22}$$

It is straightforward that by reducing the number of devices and adding constraints, the complexity is also appreciably reduced, as also proved by the divide-and-conquer approach.

3.5.2 Post-processing phase

The goal of the post-processing phase is to fix the following two issues in the solutions obtained after the JCAR phase:

(i) The logical topology could be disconnected.

(ii) Some NICs could be not set on any channel.

Hence, the post-processing phase ensures that the resulting logical topology, let us say $G^*(G)$, is connected and that there are no unused NICs. These issues are solved in two subsequent steps, which are described below and illustrated by means of the pseudo-code in Algorithm 3. Fixing these issues would not be necessary if the solutions were found by solving an overall optimization problem, such us the one proposed in [78], which takes into account all the constraints at the same time.

In this phase we introduce a new metric, called D_{tot}, which represents the total interference weighted on the traffic load and the distance from the gateway.

$$D_{tot}=\sum_{k\in\mathcal{K}}\left[\sum_{\substack{p\in\mathcal{V}\\ e_{pg}\in\mathcal{E}}}\left(\lambda_{pg}^k\cdot 2^{\mathcal{I}_\theta-h_{p\theta}^G}+\sum_{\substack{q\in I_p\\ e_{qg}\in\mathcal{E}}}\lambda_{qg}^k\cdot 2^{\mathcal{I}_\theta-h_{q\theta}^G}\right)\right], \tag{3.23}$$

Where I_p represents the set of devices whose transmissions would interfere with those from device p if they occur in overlapping time intervals. Note that $\lambda_{pg}^k=0$ if $e_{pg}\notin G^*(G)$. D_{tot} is used by G-PaMeLA to decide which pairs $\langle q,k\rangle$ $(q\in\mathcal{V},\ k\in\mathcal{K})$, if added to the solution $\sigma_{c,r}$, produce the least overall interference.

Algorithm 3 details the two procedures that create the post-processing phase. The first procedure forces $G^*(G)$ to be connected by adding logical links to any solution $\sigma_{c,r}$ (lines 36-50). This is achieved by creating the logical links that incur the least interference, according to D_{tot}. Instead, the second procedure increases the connectivity of $G^*(G)$. Due to Eq. (3.4) some NICs may not have been assigned to any channel (lines 38-48). This is fixed by exploiting the unused NIC in order to improve the quality of the solution. To obtain this improvement, three different criteria are defined. We indicate the set of criteria as U and a particular instance as $u\in U$. The criteria were defined as described below.
Given a device t with unused NICs:

1. if there is a neighbor q also with an unused NIC, choose the channel k to set the edge e_{tq} such that D_{tot} is minimized and q has no other edges on k (lines 55-64);

Algorithm 3 Post-processing Phase: pseudo-code.

```
procedure Post-processing
    procedure Fix disconnected devices
        for all $\sigma_{c,r}$ with $c \in \mathscr{C}$, $r \in R$ do
            for all $t \in \mathcal{V}$ do
                if $\mu_{t\theta} = 1$ then continue // If $t$ can reach the gateway
                else
                    for all $e_{tq} \in \mathcal{E}$, $k \in \mathcal{K}$ do
                        Find the pair $\langle q, k \rangle$ such that: $\mu_{t\theta} = 1$ and $\min D_{tot}$
                    end for
                end if
                if $\nexists \langle q, k \rangle$ then continue
                else Update $\sigma_{c,r}$ with $x_{tq}^k = x_{qt}^k = 1$ and $y_t^k = y_q^k = 1$ // Create a new logical link
                end if
            end for
        end for
    end procedure
    procedure Assign channels to unused NICs
        for all $c \in \mathscr{C}$, $r \in R$, $u \in U$ do
            $\sigma_{c,r}^u = \sigma_{c,r}$ // Create the new solution to update
        end for
        for all $t \in \mathcal{V}$ do // Criterion 1
            while $\sum_{k \in \mathcal{K}} y_t^k < \chi$ do
                for all $e_{tq} \in \mathcal{E}$, $k \in \mathcal{K}$ do
                    Find the pair $\langle q, k \rangle$ such that: $\sum_{k \in \mathcal{K}} y_q^k < \chi$ and $y_q^k = 0$ and $\min D_{tot}$
                end for
            end while
            if $\nexists \langle q, k \rangle$ then continue
            else Update $\sigma_{c,r}^1$ with $x_{tq}^k = x_{qt}^k$ and $y_t^k = y_q^k = 1$ // Create a new logical link
            end if
        end for
        for all $t \in \mathcal{V}$ do // Criterion 2
            while $\sum_{k \in \mathcal{K}} y_t^k < \chi$ do
                for all $e_{tq} \in \mathcal{E}$, $k \in \mathcal{K}$ do
                    Find the pair $\langle q, k \rangle$ such that: $\sum_{k \in \mathcal{K}} y_q^k = \chi$ and $\delta_{min}^k(q) \neq \delta_{min}$ and $\min D_{tot}$
                end for
            end while
            if $\nexists \langle q, k \rangle$ then continue
            else Update $\sigma_{c,r}^2$ with $x_{tq}^k = x_{qt}^k$ and $y_t^k = 1$ // Create a new logical link
            end if
        end for
        for all $t \in \mathcal{V}$ do // Criterion 3
            while $\sum_{k \in \mathcal{K}} y_t^k < \chi$ do
                for all $e_{tq} \in \mathcal{E}$, $k \in \mathcal{K}$ do
                    Find the pair $\langle q, k \rangle$ such that: $\min D_{tot}$
                end for
            end while
            if $\nexists \langle q, k \rangle$ then continue
            else Update $\sigma_{c,r}^3$ with $x_{tq}^k = x_{qt}^k$ and $y_t^k = 1$ // Create a new logical link
            end if
        end for
    end procedure
end procedure
```

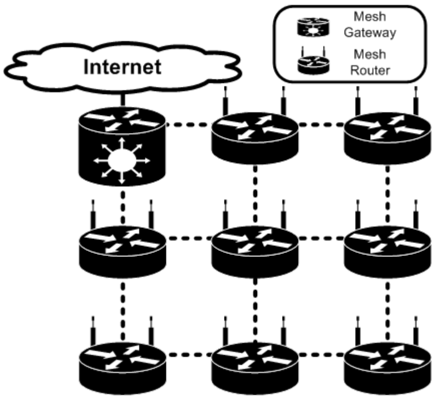

Figure 3.3: Square-grid topology.

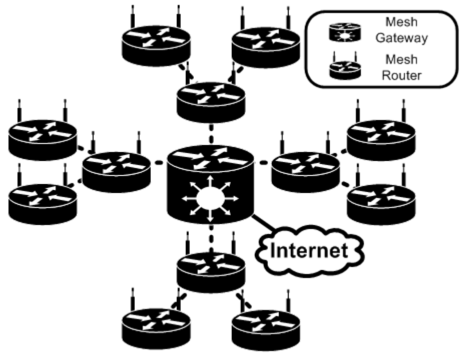

Figure 3.4: Hierarchical topology.

2. if all the neighbors have all the NICs tuned onto a channel then choose a neighbor q and a channel k such that q is not a bottleneck device on channel k, i.e. if $\delta^k_{min}(q)$ is greater than the overall δ_{min}, and set the unused NIC to the common channel k (lines 65-74);

3. re-use the channel k to add a logical link with q such that D_{tot} is minimized (lines 75-84).

Each of the above three criteria is less restrictive than the previous one in terms of overall interference. In all cases, channel k is selected so as to minimize D_{tot}. Thus, for every solution $\sigma_{c,r}$ obtained from the first procedure, the output of the second procedure is a set of three solutions $\sigma^1_{c,r}$, $\sigma^2_{c,r}$ and $\sigma^3_{c,r}$. The exact procedure is reported in Algorithm 3.

After all the possible solutions are found, whose number is smaller than or equal to $|\mathscr{C}| \times |R| \times |U|$, the final solution is given by:

$$\bar{\sigma} = \underset{\substack{\sigma^u_{c,r} \\ c\in\mathscr{C}, r\in R, u\in U}}{\arg\min} \frac{\sum_{p\in\mathcal{V}} \delta^{\sigma^u_{c,r}}_{min}(p)}{\delta^{\sigma^u_{c,r}}_{min}}. \tag{3.24}$$

The so called *best solution* ($\bar{\sigma}$) is the one that provides the best compromise between the max-min fairness and the load-aware interference objectives.

3.5.3 Customized versions of G-PaMeLA

In the following section we customize G-PaMeLA identifying different ranking functions and enforcing different routing constraints via several combinations of network crews. In fact, several network topologies are identified as well-known physical patterns for WMNs. In these cases, G-PaMeLA customizes network crews and ranking functions computations *ad-hoc* for the physical topology graph. We analyze two types of topologies: hierarchical and grid [32], which are of practical interest in application scenarios where WMN is deployed from scratch by a network operator, e.g.

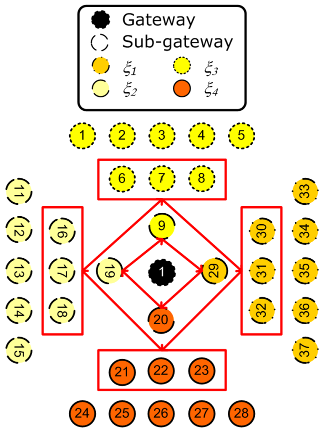

Figure 3.5: Ranking function and Crews for a 37 devices extended star.

municipal wireless, and where multi homing and overlaid access networks need to be set up (Fig. 3.3-3.4).

In terms of *hierarchical topology* we can address stars and trees, in fact, the tree topology is a collection of stars arranged in a hierarchy. An example of *extended star* topology is shown in Fig. 3.4. The *root device*, which is the only one that has no other device above it in the hierarchy, is chosen as a mesh gateway. The crews are automatically defined because each device reaches the gateway only through a single sub-gateway. Instead the ranking functions need to be defined. We choose the gateway as the device with the highest rank, then we split each $\Omega_{d,\theta}$, with $d = \{1, \dots, \theta\}$, into sets of devices depending on the different sub-gateways that they use to reach the gateway. The order in which sets run the ILP problem at the same distance from the gateway is not relevant because the network's topology is symmetric. Figure 3.5 shows ranking function and network crews.

The main problem in the hierarchical topology is that an individual device may be isolated from the network and thus from the gateway too. In fact, if a *leaf device* does not have a routing path to the gateway, that leaf is isolated, but if a *non-leaf device* has no routing path to the gateway, an entire section of the network becomes isolated from the rest. Due to failure issues the hierarchical topology is rarely applied in a real WMN, and is thus not focused on here where instead, we prefer focus on grid topologies.

Grid topology networks are widely used in both theoretical and experimental studies on WMNs [93, 92, 34, 7]. In addition, in [95] via a numerical analysis it has also been shown that square-grids have several beneficial properties for networks where devices are uniformly distributed. Therefore, we assume that the WMN devices are located in a square-grid physical topology graph. To simplify our explanation, we consider $|\Theta| = 1$ and so the gateway θ is placed in the corner of the square-grid, as shown in Fig. 3.6 and Fig. 3.7.

In the JCAR phase, we identify V sub-problems, one for each device. Therefore, the ILP problem in Fig. 3.2 is unchanged if we consider $\Omega_{d,\theta}$ as a set containing only one device, i.e. $d = \{0, 1, \dots, V-1\}$, and the ranking function as a criterion to sort the devices in $\mathcal{V}$. As previously explained, we can run $V-2$ ILP sub-problems,

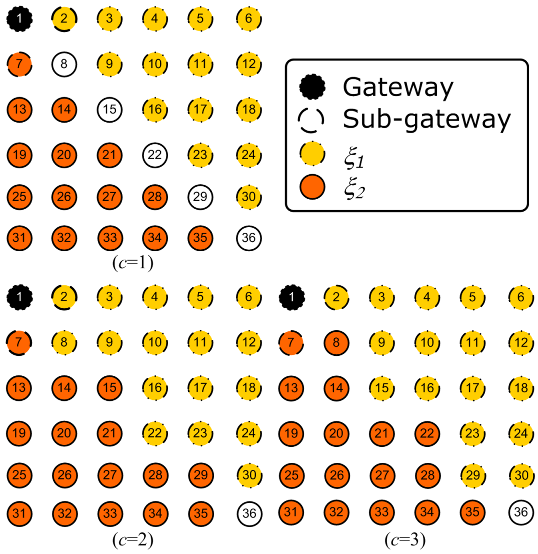

Figure 3.6: Network crews for a 6×6 grid.

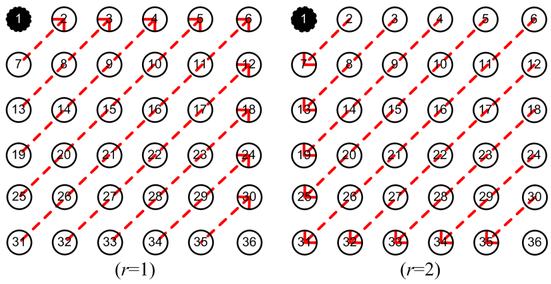

Figure 3.7: Ranking functions for a 6×6 grid.

in fact the last device cannot choose its channel assignment because it has already been chosen from its neighborhood. The time needed to solve an individual JCAR sub-problem in a square-grid is less than in G-PaMeLA due to there being fewer NICs and edges to assign, whereas the number of sub-problems is greater.

We found that using any of the three network crews illustrated in the example in Fig. 3.6 gives good results in all the networks and traffic configurations tested. For the same scenario, we consider a number of ranking functions equal to the number of sub-gateways ($|\Omega_{1,\theta}|$). The two ranking functions tested are illustrated in the example in Fig. 3.7, where devices closer to the gateway are considered at an early stage of channel assignment and path selection, as explained in Section 3.5.1. This means that, every $\mathcal{P}_{c,r}$ is solved by visiting each device one by one in the order defined by r.

The maximum number of solutions $\sigma_{c,r}$ at the end of the JCAR phase, and passed to the post-processing phase, is $|\mathscr{C}| \times |R|$.

The post-processing phase as well as the best solution computation are equal to G-PaMeLA. For the sake of brevity in Chapter 4 we call the customized algorithm for the JCAR problem *PaMeLA*, as in [32].

All life is an experiment. The more experiments you make the better.
Ralph Waldo Emerson

4 Performance Evaluation

4.1 Introduction

In this section we compare different JCAR solutions proposed in the literature: Hyacinth [92], FCRA with the rate adaptation disabled [12], LACA [93] and *G-PaMeLA* in its generalized and customized versions [32, 40]. Additionally, a random channel assignment is considered as a reference.

4.1.1 Outline of the Chapter

The Chapter is organized as follows. Section 4.2 reports the execution time of several scenarios in order to show that a mixed approach can be run at the time scale of provisioning with non-specialized hardware. The rest of the analysis is carried out via a detailed packet-level simulation, whose settings are described in Section 4.3. The results for several topologies are discussed in Sections 4.4, 4.5 and 4.6.

4.2 Execution time analysis

In this section we analyze the amount of time required by our mixed approach (*G-PaMeLA*) to produce a channel assignment and routing for each node. First, we briefly describe the tools used and then, we report the results in several different conditions.

G-PaMeLA was implemented using the ILOG tools AMPL [51] and CPLEX [52] for describing and solving the ILP instances in the JCAR phase. Input preparation and post-processing were implemented in C++. The results were obtained with a dedicated Linux 2.6 workstation equipped with an Intel Core 2 CPU at 2.13 GHz and 2 GB of main memory. The C++ code was compiled with the GNU gcc compiler version 3.4.6, with architecture-specific optimizations.

Figure 4.1 shows the amount of time required by *G-PaMeLA* to solve the JCAR problem when the number of nodes increases and the number of NICs and allowable channels change. Note that *G-PaMeLA* is easily scalable because the maximum complexity is given by the time needed to solve an individual neighborhood, which in turn depends on the number of neighbors of each device (see Section 3.5.1 for more details). The time generally increases with the number of nodes and when the number of channels increases. In contrast to the general formulation of a JCAR

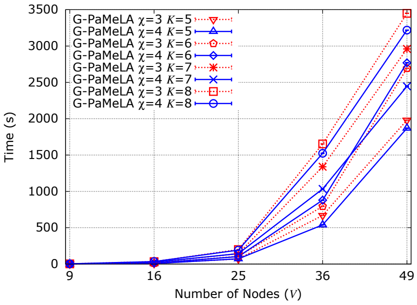

Figure 4.1: *G-PaMeLA* execution time against the number of nodes with different K and χ.

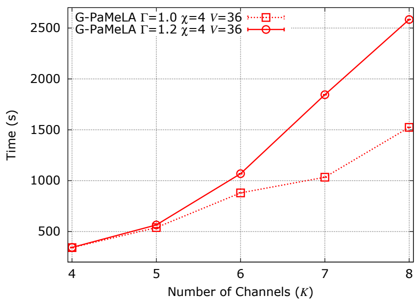

Figure 4.2: *G-PaMeLA* execution time against the number of channels with $V = 36$, $\chi = 4$ and $\Gamma = 1$ or $\Gamma = 1.2$.

problem in Eq. (3.20), less time is required if each node is equipped with more NICs. In fact, *G-PaMeLA* is a greedy algorithm, thus by equipping nodes with more NICs, helps a set $\Omega_{d,\theta}$ to be *less greedy* with respect to the other nodes. Consequently, sets of nodes with fewer ranks can find a channel assignment and a routing path solution more easily.

Figure 4.2, on the other hand, shows how the routing influences the execution time. Given a number of NICs and allowable channels, when the authorized path length increases, then so too does the time needed to find a solution. From Fig. 4.2, it is also possible to see how the complexity increases when the number of channels increases.

In conclusion, the execution time of *G-PaMeLA* depends on the number of allowable channels and on the cardinality of the sets ($\Omega_{d,\theta}$) with a higher rank. In Fig. 4.1, we consider sets that are always bigger than the dimension on the network and thus the execution time increases. Therefore the complexity of *G-PaMeLA* does not actually increase with the total number of nodes, rather it depends on the cardinality of each set and in particular the dimension of the sets with higher ranks, i.e. sets of nodes under fewer constraints. This unfortunate situation is solvable by defining a different property to create the sets of nodes.

The extended star topology helps to clarify this concept. As can be seen in Fig. 4.3, by splitting the sets of nodes at the same distance in the number of hops to the gateway, the execution time of our divide-and-conquer approach is substantially reduced. Therefore, in this context knowledge of the topology does help.

As further example is the grid topology. Figure 4.4 shows that *PaMeLA* is advantageous in finding a JCAR solution in terms of execution time. For example, let us consider a non-peripheral node. In a grid topology, each non-peripheral node has a neighborhood of 9 nodes. However we also need to consider that some of these nodes have a bigger value in the ranking order and for this reason they have already solved their JCAR sub-problem. As an example let us consider node 15 and $r = 1$ in Fig. 3.7. The neighborhood is $neigh(15) = \{8, 9, 14, 16, 20, 21, 22\}$ but nodes $\{8, 9, 14, 20\}$ have already been considered and therefore the ILP sub-problem of node 15 solves a smaller neighborhood. Therefore, in a grid topology the maximum neighborhood to be solved is in the order of 4 nodes. This analysis highlights that

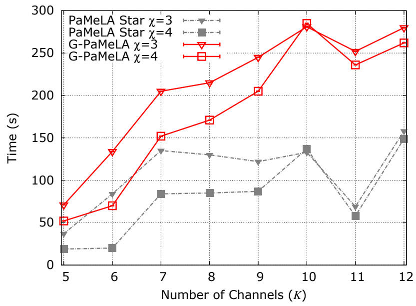

Figure 4.3: Comparison between *G-PaMeLA* and extended star execution time against the number of channels (K).

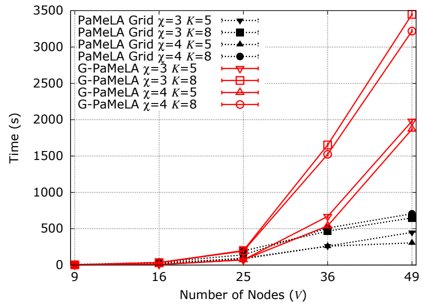

Figure 4.4: Comparison between *G-PaMeLA* and *PaMeLA* execution time against the number of nodes (V) with several combinations of K and χ.

using a customized version of *G-PaMeLA* based on the topology knowledge speeds up the JCAR phase.

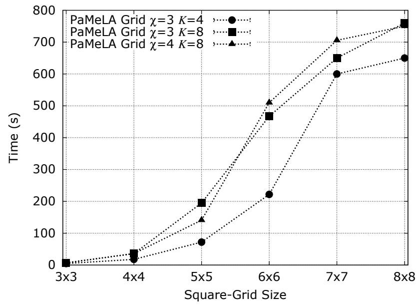

Figure 4.5: *PaMeLA* execution time against the square-grid size with several combinations of K and χ.

To analyze the behavior of a grid topology in more detail, we show its execution time in three different conditions: $\chi = 3$ and $K = 4$, $\chi = 3$ and $K = 8$, $\chi = 4$ and $K = 8$. Figure 4.5 shows the behavior when the grid sizes change. To better explain our results we choose a square-grid but all the conclusions can be applied to a grid with an arbitrary dimension. As can be seen, even with rather large WMNs, consisting of more than 36 nodes, and considering different configurations, the execution time is relatively small.

In conclusion, knowledge of the topology guarantees the possibility of applying several simplifications, however the more general *G-PaMeLA* approach is also less expensive than an optimal algorithm thanks to the divide-and-conquer technique. Solving the global JCAR problem with the same hardware and software used for *G-PaMeLA* is impractical, since the execution time is in the order of many days, starting from 4×4 WMNs. Moreover, in the following we show that our mixed approach is also better than the empirical approaches described in Section 3.4.

Table 4.1: Physical Layer Parameter Values.

Minimum SINR for each PHY								
PHY rate (Mbps)	6	9	12	18	24	36	48	54
SINR (dB)	9	10	11	13	17	20	25	27
P^T	17 dBm							
Band	5.15 GHz							
$\mathcal{W}$	-95 dBm							

4.3 Simulation environment

The simulation study was carried out with ns-2 [53], which was modified to enable a multi-radio multi-channel feature and to include the SINR-based physical interference model in Eq. (2.1). Using the physical interference model, we can obtain a realistic measurement of the interference resulting from the JCAR solution. Without loss of generality we assume that P^T in Eq. (2.2) is equal for all the nodes.

The MAC frame is assumed to be correctly received if the SINR is greater than or equal to the minimum value required by the IEEE 802.11 standard. The parameters used in the simulations, which refer to the IEEE 802.11a OFDM-based physical layer, are reported in Table 4.1. All experiments were conducted with the RTS/CTS mechanism disabled.

We take into consideration different topologies and we assume that the distance between one-hop neighbors is 140 m. The channel rate for a physical link is set to 6 Mbps. As far as traffic is concerned, each node (except the gateway) is assumed to have exactly one bidirectional Constant Bit-Rate (CBR) traffic flow towards the gateway. The packet size is kept constant at 1024 bytes. The following transmission rates were considered: 26, 40, 52, 70, 85 or 104 Kbps. The duration of each simulation was 500 s, which was verified to be enough for the simulated system to reach a steady-state. Samples were not collected during the first 100 s to remove the initialization bias. Several independent replications for each scenario were run, according to the independent replications method [65]. Mean values were then estimated along with 95% confidence intervals, which are not reported in the figures whenever negligible.

4.3.1 Performance Metrics

To assess the performance, the following performance metrics were used.

(i) Collision probability: defined as the probability that a packet experienced a collision along the path from sender to receiver nodes.

(ii) Throughput: defined as the number of bits received by the receiver nodes in the unit of time.

(iii) Normalized throughput: defined as the throughput obtained with a given JCAR schemes divided by the random channel assignment scheme throughput.

(iv) Fainess index: defines in Eq. (4.1) as a *Jain's fairness index*.

$$fairness = \frac{(\sum x_i)^2}{n \cdot \sum x_i^2}. \tag{4.1}$$

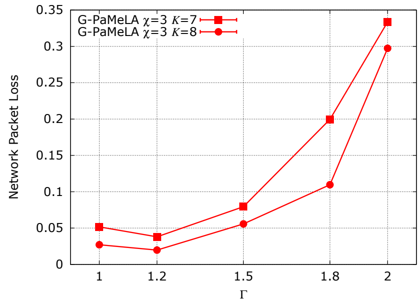

Figure 4.6: Network packet loss vs. Γ with $\gamma_{sr} = 24$ Kbps and $V = 36$.

Where x is the throughput for each flow and n is the number of flows. The result ranges from $1/n$ (worst case) to 1 (best case).

(v) Packet loss: defined as the ratio between the number of packets sent by the sender node and the number of packets correctly received by the receiver node. There are two reasons why a packet may be dropped: (a) the maximum number of retransmissions at the MAC layer is exceeded (we set this number to 7, which is the default in most devices); (b) the MAC buffer has overflowed (we set the buffer size to 100 packets for each node except for the gateway where the size is proportional to the number of flows ingoing and outgoing).

All metrics were collected for per traffic flow, per node, per link, per channel and as network aggregates. Note that to increase readability, we sort flow IDs in the increasing value of the performance metric.

4.4 *G-PaMeLA* analysis

In this section we analyze the overall *G-PaMeLA* packet loss when the routing parameter (Γ) changes and its fairness as defined in Eq. (4.1). The following results apply to a uniform topology with 36 or 25 mesh nodes.

Figure 4.6 shows the overall packet loss when Γ ranges from 1 to 2 and 36 nodes are considered. $\Gamma = 2$ allows a routing path with a double length, in the number of hops, compared to the shortest path represented by $\Gamma = 1$. Values of Γ close to 1 only affect a long path, in contrast Γ close to 2 influences all the network. Increasing Γ after a certain value produces more interference, in fact a longer path produces packet collisions on more links. However by setting Γ properly, the overall packet loss decreases.

The throughput fairness was defined earlier and consists of evaluating whether each flow is receiving a fair share of system resources. Figure 4.7 shows the fairness when the flow rate for each flow increases. Immediately we can see that *G-PaMeLA* guarantees the fairness between flows from different nodes, i.e. flows that are one-hop to the gateway or flows at a distance of I_θ are also treated in the same way when the network is overloaded. In conclusion, the use of *G-PaMeLA* guarantees a better fairness in terms of throughput compared to the state-of-the-art JCAR solutions taken into consideration.

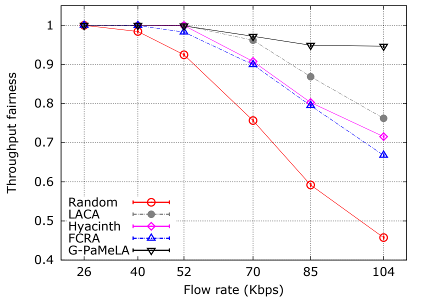

Figure 4.7: Throughput fairness vs. individual Flow rate with $\chi = 3$, $K = 8$ and $V = 36$.

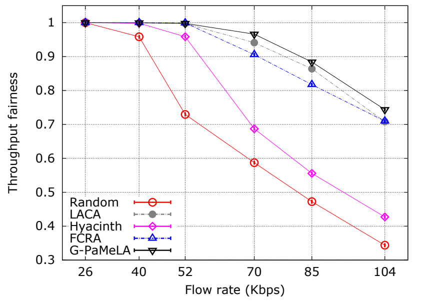

Figure 4.8: Throughput fairness vs. individual Flow rate with $\chi = 3$, $K = 4$ and $V = 25$.

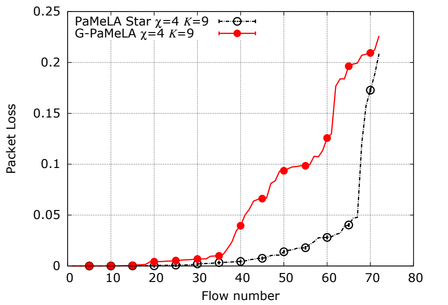

Figure 4.9: Comparison between *G-PaMeLA* and Extended Star in terms of Packet loss vs. Flow ID with $\chi = 3$, $K = 9$ and $\gamma = 104$Kbps.

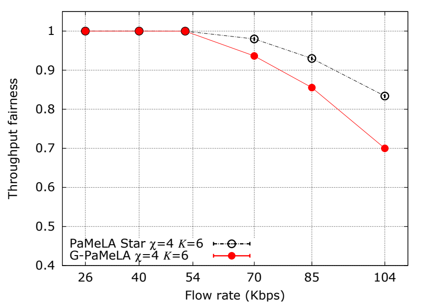

Figure 4.10: Comparison between *G-PaMeLA* and Extended Star in terms of Throughput Fairness vs. individual Flow rate with $\chi = 4$ and $K = 6$.

For the sake of completeness in Fig. 4.8 we show the throughput fairness when the flow rate for each flow increases for a WMN with 25 nodes. Each node is equipped with 3 NICs and 4 allowable channels are considered. We notice how the results and conclusions are similar to the case with 36 nodes presented in Fig 4.7.

4.5 Extended star analysis

As previously mentioned, the extended star topology suffers from failure issues due to a low connectivity. However, this topology is easy to implement and for this reason we show in Fig. 4.9 and Fig. 4.10 the performance in terms of packet loss and throughput fairness, respectively. The extended star topology is compared with *G-PaMeLA* to highlight how knowledge of the topology is an advantage in the JCAR problem.

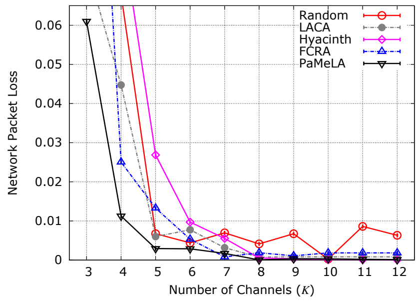

Figure 4.11: Network-wide packet loss vs. number of channels with $\chi = 3$ and $\gamma_{sr} = 26$ Kbps.

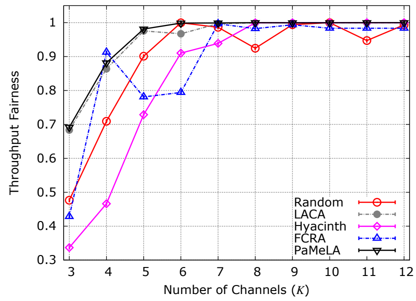

Figure 4.12: Throughput fairness vs. number of channels with $\chi = 3$ and $\gamma_{sr} = 52$ Kbps.

4.6 Grid analysis

In order to analyze the behavior of a customized version of *G-PaMeLA*, we consider 36 nodes arranged in a 6×6 square grid topology as in Figs. 3.6 and 3.7. Depending on the scenario, each node in the WMN is equipped with 3 or 4 NICs (χ), while the total number of allowable channels (K) ranges from 3 to 12.

PaMeLA was configured with $\Lambda = 0.8$, $\Gamma = 1$, and $\beta = 3.5$. The channel assignment and routing output with all the combinations of χ, K, and γ_{sr} were tested through simulation experiments.

We begin by showing in Fig. 4.11, the impact of decreasing the number of allowable channels on the physical wireless network technology. Each node is equipped with 3 NICs and each flow rate is $\gamma_{sr} = 26$ Kbps. This experiment demonstrates that the channel assignment algorithm can adapt itself when the number of allowable channels decreases. In fact *PaMeLA* uses all the channels to split the collision domain and thus increases the cross-section throughput.

In the previous section we considered the fairness given by *G-PaMeLA* when the flow rate increases, instead in this section, we set the flow rate to $\gamma_{sr} = 52$ Kbps and change the number of allowable channels. The results when each node is equipped with 3 NICs are shown in Fig. 4.12 where it is possible see that *PaMeLA* guarantees a high degree of fairness no matter how many channels are used.

To complete the analysis, we show in Fig. 4.13 the outcome of increasing the total number of NICs on each node. We can see that with 3 NICs the network performs worse than with 4 NICs per node, but this improvement is not significant.

To sum up, equipping each node with 4 NICs does not help as much as increasing the total number of channels in the network, as also shown in several papers in the literature, e.g. [93]. However, as shown in Section 4.2 could help in terms of execution time.

We conclude our performance analysis by considering the effect of *PaMeLA* on each individual flow. First we show the throughput, then packet loss and collision probability.

Figure 4.14 shows the normalized throughput with respect to the random channel assignment. We show the case with 3 radios, 4 channels and $\gamma_{sr} = 26$ Kbps. Some studies [27] have been carried out to estimate the maximum throughput in arbitrary

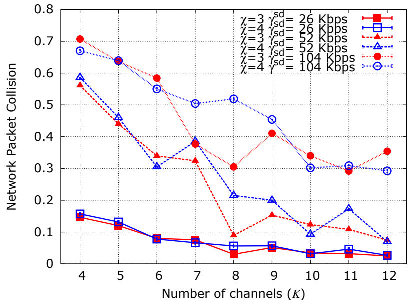

Figure 4.13: Network-wide collision probability with $\chi = 3$ and $\chi = 4$.

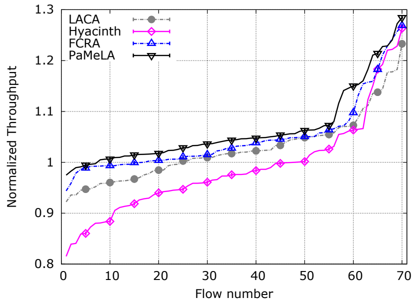

Figure 4.14: Normalized throughput per traffic flow, with $\chi = 3$, $K = 4$, $\gamma_{sr} = 26 Kbps$.

wireless networks with SINR but the results still apply to a simplified SINR model. Our analysis indicates that a random channel assignment is not enough and only a suitable JCAR algorithm can use the advantage provided by a MRMC-WMN. *PaMeLA* performs better than the other algorithms considered in this study. In fact, Fig. 4.14 shows that it also guarantees a better fairness between all the flows if the path length that they have to follow varies from a minimum of one hop to a maximum of $\Gamma \cdot h_{sr}^{G}$ hops (10 in the case considered in this Section).

The packet loss and the packet collision probability with 3 NICs per node and 8 allowable channels are shown from Fig. 4.15 to Fig. 4.20 where we consider three different traffic loads $\gamma_{sr} = 26$ Kbps, $\gamma_{sr} = 52$ Kbps and $\gamma_{sr} = 104$ Kbps, respectively. It is clear how packet loss and packet collision probability decrease in the same way. Comparing *PaMeLA* with LACA, it is possible to see how, even though they are comparable in terms of throughput fairness as seen in Fig. 4.7, *PaMeLA* achieves the same fairness by ensuring less packet loss per flow.

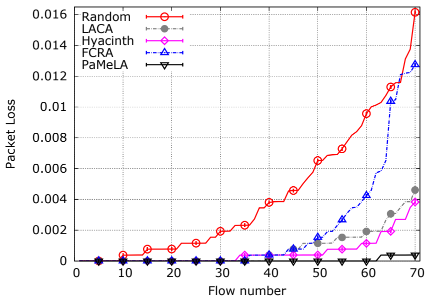

Figure 4.15: Packet loss per traffic flow with $\gamma_{sr} = 26$ Kbps, $\chi = 3$ and $K = 8$.

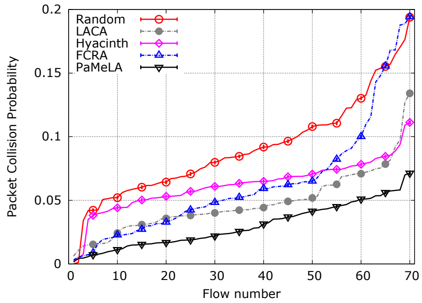

Figure 4.16: Packet collision probability per traffic flow with $\gamma_{sr} = 26$ Kbps, $\chi = 3$ and $K = 8$.

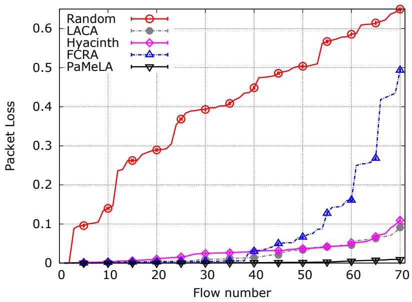

Figure 4.17: Packet loss per traffic flow with $\gamma_{sr} = 52$ Kbps, $\chi = 3$ and $K = 8$.

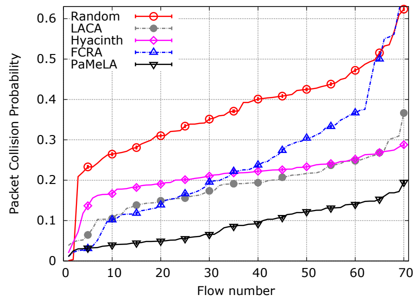

Figure 4.18: Packet collision probability per traffic flow with $\gamma_{sr} = 52$ Kbps, $\chi = 3$ and $K = 8$.

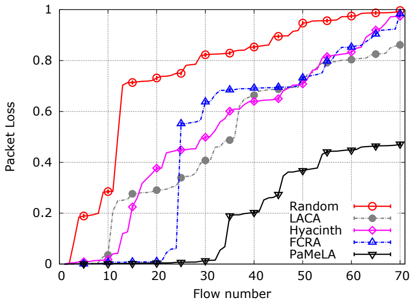

Figure 4.19: Packet loss per traffic flow with $\gamma_{sr} = 104$ Kbps, $\chi = 3$ and $K = 8$.

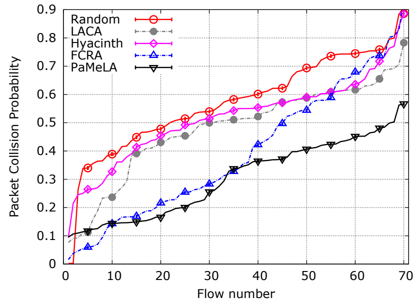

Figure 4.20: Packet collision probability per traffic flow with $\gamma_{sr} = 104$ Kbps, $\chi = 3$ and $K = 8$.

Things should be made as simple as possible, but no simpler.
Albert Einstein

5
Conclusions

In this part we addressed the spectrum shortage problem in MRMC-WMNs. We analyzed the join routing and channel assignment problem and proposed several approaches presented in the literature. The techniques proposed are distinguished in optimization approaches and empirical approaches. The former have to deal with too high execution times, while the latter produce solutions with poor performance results. In order to overcome these limitations we proposed a mixed approach, *G-PaMeLA*, which is a divide-and-conquer approach to splitting in local sub-problems the joint channel assignment and routing algorithm for MRMC-WMNs.

In order to evaluate *G-PaMeLA* we compared different JCAR solutions presented in the literature. The performance was evaluated through a detailed packet-level simulation with several combinations of allowable channels and NICs per node. This comparison shows that the execution time of *G-PaMeLA* is relatively low, which makes it feasible for real MRMC-WMNs with non-specialized hardware, even for large mesh networks with tens of nodes. Moreover, the results demonstrate that our scheme significantly improves network performance, in terms of the packet loss of all traffic flows and throughput fairness.

Part II

Channel Assignment in Cognitive Access Networks

All would live long, but none would be old.

Benjamin Franklin

6 Cognitive Access Networks

6.1 Introduction

The goal of our work is to address the spectrum shortage problem in wireless networks. In the previous part, we analyzed solutions for multi-radio multi-channel wireless mesh networks act to enable the use of multiple unlicensed channels. Our analysis showed that channel assignment techniques can effectively increase network performance in particular when they are jointly used with routing algorithms.

However, joint channel assignment and routing algorithms are still affected by spectrum overcrowding problems. With the aim of addressing these problems, careful analysis on unlicensed and licensed spectrum frequencies has been conducted. Surprisingly, the spectrum frequencies analysis showed that licensed channels are underutilized compared to unlicensed channels [37]. Hence, the networking community has moved his focus on licensed channels seeing in them an enormous potential in addressing the spectrum shortage problem.

Wireless networks, which are able to opportunistically use licensed channels, fall under the name of *cognitive networks*. In this and in the following parts, we extensively describe and propose solutions for cognitive networks because we see in them the most attractive wireless network architecture to solve the spectrum shortage problem.

6.2 Cognitive Networks

Recent studies [37] have shown an abundance of underutilized radio spectrum resources, which led to the creation of new classes of wireless networking technologies. One such is the family of *Cognitive Networks* (CNs), which opportunistically operates mainly in the *TV White Space* (TVWS) and in general in any unused frequency spectrum. The TVWS consists of chunks of spectrum allocated to the TV broadcasting service but not all used locally.

CNs offer an enormous potential because are easily maintainable networks, which are continuously improved and upgraded in a way that is completely integrated with surrounding environments [71]. However, they are not without significant challenges, due to the variability of available resources in time and space.

A typical CN consists of Primary Users (PUs) and Secondary Users (SUs).

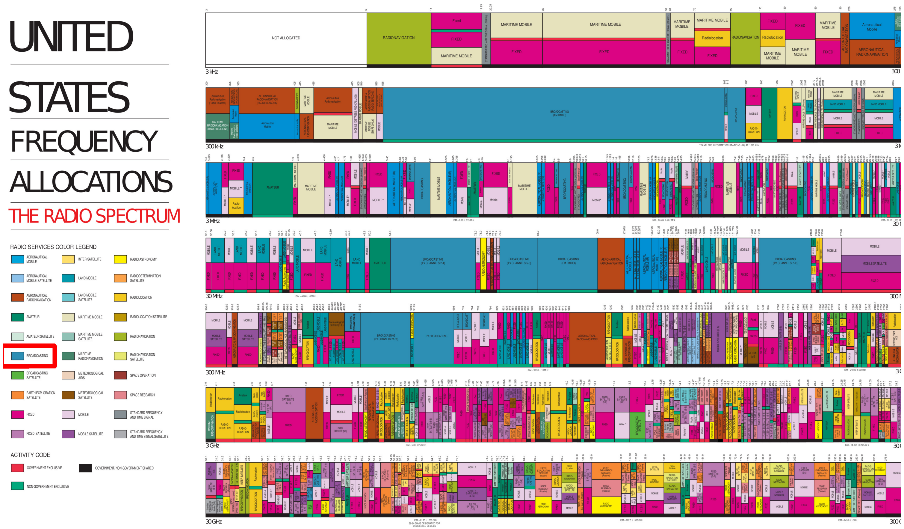

Figure 6.1: Spectrum utilization in the United States [54]. The light blue chunks (circled in red on the left) are TV channels.

(i) PUs are licensed users. Example PUs include Digital TV transmitters (DTVs), microphones, and generally any device that requires a license and payment to use chunks of the spectrum.

(ii) SUs are unlicensed cognitive devices (CDs) which want to access the licensed spectrum without paying any fee. Hence, SUs opportunistically grab the unused frequencies without causing any harmful interference to PUs. Example SUs are base stations and end-users, which provide services in rural and remote areas, CDs used in order to create emergency networks in disaster areas, and in general any device that wants to operate into the licensed spectrum but does not have any right.

All the complexity of the spectrum sharing is borne by the SUs, who deploy rules to opportunistically operate in the licensed spectrum and are essentially invisible to PUs. Therefore the PUs do not require any change in their spectrum management.

In the United States, the *Federal Communications Commission* (FCC), who has the regulatory responsibility for the non-Federal radio spectrum (i.e., private internal business, and personal use), provides advices on technical and policy issues pertaining to spectrum allocation. These rules dictate that when an SU senses a PU by listen to its beacon or decoding an overheard primary message, it needs to vacate the channel and switch to another within a *channel move time* of two seconds [37]. Figure 6.1 shows how the frequency spectrum from 3 kHz to 300 GHz is assigned by the FCC to different services [54, 55]. The figure highlights how every chunk of the frequency spectrum has been assigned to a specific use and that the same frequency can be shared between different services (vertical division of frequencies). Hence, in order to deal with the increasing demands for spectrum, the FCC proposed the adoption of a new regulatory spectrum policy, which allows the use of licensed spectra by unlicensed users.

Although design CRs is challenging from an electronic point of view, they are becoming a reality due to recent "Moore's law" advances in programmable integrated circuits that have created the opportunity to develop radios that can adapt to a wide variety of interference conditions and multiple protocol standards. Resulting in a collaboration between otherwise incompatible systems.

Two standards, namely IEEE 802.19 and IEEE 802.22, have been proposed in order to exploit CNs. The former specifies radio technology methods to enable the family of IEEE 802 wireless standards to use TVWS most effectively by providing standard coexistence methods among unlicensed devices. On the other hand, IEEE 802.22 [56] is based on *Cognitive Radios* (CRs) and targets wireless broadband access in rural and remote areas using TVWS in very high frequency (VHF) and ultra high frequency (UHF) bands, while avoiding interference with the PUs. The IEEE 802.11 group is also working on the new IEEE 802.11*af* standard for TVWS with the initial draft planned by the end of 2010.

6.2.1 Origins of Cognitive Radio

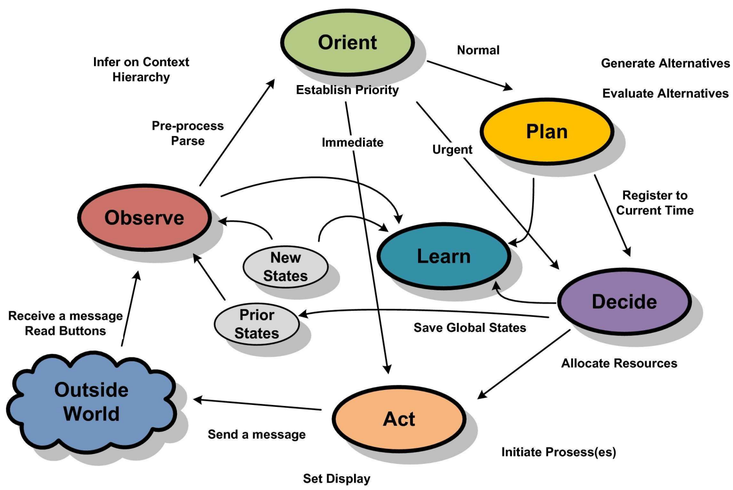

Figure 6.2: Cognition cycle.

The cognitive radio (CR) terminology was coined by Mitola who, for the first time, discussed in [77] the potential contribution of CR to spectrum sharing.

CRs work following a cognition cycle, as shown in Fig. 6.2. The *outside world* provides stimuli based on which SUs have to behave. In the *observe stage*, SUs parse information and identify the communication context. This parsed information is used by the *orient stage* in order to decide the priority of the communication: normal, urgent or immediate. To each one of these priorities correspond a stage: plan, decide and act. In the *plan stage*, SUs generate and evaluate alternatives. In the *decide stage*, resources are reserved for the communication. In the *act stage*, SUs manage the communication. To sum up, the cognition cycle promotes the adaptability of SUs to the real-time conditions of the surrounding environment.

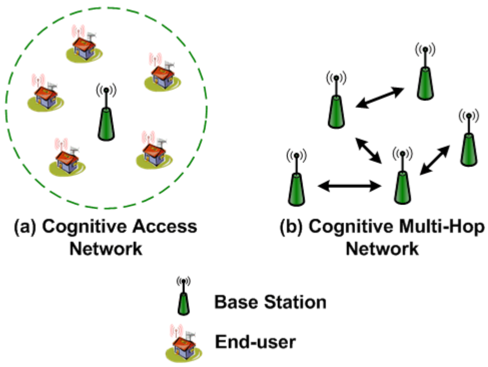

Figure 6.3: Network architectures. (a) Cognitive access network. (b) Cognitive multi-hop network.

6.2.2 Cognitive Network Architectures

CNs can be broadly divided in *cognitive access networks* (C-ANs) and *cognitive multi-hop networks* (C-MHNs). This division is made based on the communication paradigm used, point-to-multipoint in the former and point-to-point in the latter. Figure 6.3 (a) and (b) show a C-AN and a C-WMN, respectively.

Cognitive Access Networks

In C-ANs, a base station (BS) supports multiple end-users and provides access to the Internet. BS and end-users are assumed to be a single entity, called *cell*. Each cell could belong to a different Internet Service Provider (ISP) and hence communication between cells may or may not exist. However, Internet connectivity is ensured to each cell and coordination mechanisms could be implemented if different ISPs agree to coordinate over wired connections. Access networks are commonly used for last-mile broadband accesses. Examples are BSs in cellular networks or access points in wireless local area networks.

Cognitive Multi-Hop Networks

C-MHNs include *ad-hoc* and *mesh* communication paradigms. They consider source-destination pairs where connections are established directly between network devices and hence a decentralized model is considered. That is, a preexisting infrastructure, such as access points in managed wireless networks does not exist. The basic difference with access networks is that in multi-hop networks no direct access to Internet is provided, in contrast each device can obtain an Internet connectivity through multiple hops. Thanks to the recent technological improvement, multi-hop networks are seen as networks where devices are self-organizing and do not need infrastructure support. The major challenge in C-MHNs is the coordination among devices, in fact, no Internet connectivity is guaranteed and the coordination needs to be set up on wireless links.

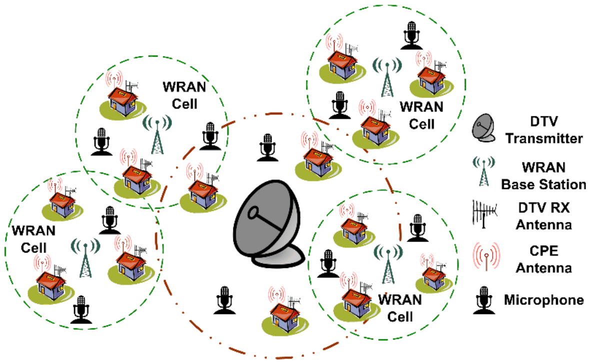

Figure 6.4: A Cognitive Access Network Architecture.

6.2.3 Outline of the Chapter

In this Part we treat and propose solutions related to C-ANs, meanwhile in Part III we deal with C-MHNs.

The remaining of the chapter is organized as follow. Section 6.3 introduces C-ANs and their commonly used terminologies. Section 6.4 analyzes the main issue in C-ANs, the *self-coexistence problem*.

6.3 Network Model and Terminology

C-ANs consider infrastructure oriented CNs where a *base station* (BS) supports multiple end-users, called *consumer premises equipment*s (CPEs). BS and CPEs are assumed to be a single entity, called *Wireless Regional Area Network* (WRAN). Each WRAN could belong to a different ISP and hence communication between WRANs may or may not exist. Internet connectivity is ensured by each BS to its CPEs and coordination mechanisms could be implemented if different ISPs agree to coordinate over wired connections.

Both BSs and CPEs have the ability to sense the frequency spectrum. The BS has global knowledge of the surroundings and thus instructs the CPEs to perform distributed sensing measurements, which may be different from CPE to CPE in order to make the sensing operations lighter. Based on the measurements and feedback received from the CPEs, the BS decides which operations to take on.

Figure 6.4 shows a typical C-AN consisting of one DTV transmitter, several microphones, and four WRANs. Network model and terminology used in this Part are as defined by the IEEE 802.22 standard.

6.4 Self-coexistence Problem

CNs are obtaining a significant success thank to their adaptability to various scenarios and their large range of applicability. However, CNs are not without challenges. One of the major challenges in CNs is the notion of *self-coexistence* [69, 35]. Self-coexistence is defined as the ability to guarantee coexistence among network devices that may or may not follow the same set of rules but want to operate in the same network without causing any disruptive interference, that is, self-coexistence is the ability to access chunks of spectrum on a non-interfering basis with respect to PUs and SUs. Hence, it can be seen as the ability to guarantee that there is no degradation in PU services and no interference between SUs.

The self-coexistence problem has to be addressed because it afflicts each type and size of CN. Even small scale emergency networks are afflicted by self-coexistence problems. An example was the fireworks depot explosion in the 2000 in Netherlands [88] where fire brigades, police and relief workers of the medical teams experienced a great deal of communication breakdown, both internally and with each another due to lacks of common standards for each disaster relief group and overloaded emergency frequency bands.

Self-coexistence can be addressed using *centralized* or *distributed* approaches depending on the nature of the network. If a centralized approach is used, common signaling among SUs or coordination entities are required. In contrast, if a distributed approach is used, SUs require self-organization and self-management abilities. In the following we separately describe the two approaches in order to understand their implications and then we move our focus on how to treat the self-coexistence problem as a channel assignment problem.

6.4.1 Centralized Approaches

Centralized approaches offer easy tractability because they allow coordination among CDs and hence agreements on the chunks of the spectrum used by each CD.

Coordination mechanisms are difficult to guarantee in CNs because these networks could house CDs that respond to different standards (IEEE 802.11, IEEE 802.22, etc.) and/or belong to different ISPs. Hence, devices may not have a common communication protocol. Moreover, devices are built by different manufactures, who may have an incentive to develop products with a selfish behavior, so that they perform better than products developed by other manufacturers.

6.4.2 Distributed Approaches

Distributed approaches are characterized by SUs that access chunks of the spectrum in a *distributed* manner by acting selfishly. A selfish behavior implies that devices are worried only about their own outcome instead of the overall network performance.

In order to obtain self-coexistence using distributed approaches, CDs need protocols and algorithms under which they are able to self-organize and self-manage their own resources taking into account the existence of other CDs using the same resources.

6.4.3 Self-coexistence as Channel assignment

In wireless environments and therefore in CNs, the most important resource is the frequency spectrum. The frequency spectrum is divided into chunks, called *channels*, which do not cause interference with each other if they are orthogonal. Hence, devices operating in orthogonal channels do not interfere. Therefore, self-coexistence can be seen as the problem of assigning channels to devices for communication purposes.

In the literature, the ability to assign channels to devices avoiding co-channel interference is referred as the *channel assignment problem*. The key concept behind an efficient channel assignment is to find appropriate channels in such a manner that devices can coexist without causing any harmful interference and network objectives are met. Usually objectives include QoS satisfaction, high spectrum utilization, and traffic throughput.

The self-coexistence problem in C-ANs can be regarded as a channel assignment problem where channels are spectrum opportunities identified by CDs and used for communication purposes on a non-interfering basis. In fact in both self-coexistence and channel assignment problems, a device must communicate without causing interference to any other device, licensed or unlicensed.

In the literature, several studies on channel assignment and TVWS utilization have been proposed [23, 90]. These works mostly treat self-coexistence as a channel assignment problem, considering that the assumptions valid for WMNs are not applicable to CNs. This is because the set of accessible channels in CRs is time-variant and also differ from device to device. Since spectrum variability adds another dimension of complexity to the channel assignment problem, new approaches and mechanisms need to be designed to support self-coexistence among CDs.

Alternatively, the self-coexistence problem can be addressed by modifying coexistence mechanisms defined by standards. Considering the IEEE 802.22 standard, a new coexistence mechanism could replace the existing one [19] or could be integrated into it [6]. Unfortunately, these mechanisms require some modifications to the IEEE 802.22 medium access control (MAC) layer that are also not desirable.

You never fail until you stop trying.

Albert Einstein

7 NoRa and HeCtor

7.1 Introduction

We propose two game theoretic frameworks, called *No*n-cooperative *R*epe*a*ted game (*NoRa*) [40] and *He*donic *C*oali*t*ional F*or*mation game (*HeCtor*) [41], in order to address the self-coexistence problem in C-ANs [39].

NoRa is a multi-player non-cooperative repeated game, which considers selfish users. It belongs to the class of *potential games* and regards self-coexistence as a distributed channel assignment problem.

HeCtor has the potentiality of form coalitions in which CDs cooperate to improve their performance. It is formulated as a hedonic coalitional formation game which allows cooperation among *disjoint* sub-groups of CDs.

In Section 7.2 we present basic concepts of game theory in order to understand terminology and approaches and in Section 7.3 we distinguish between non-cooperative and cooperative communication paradigms describing the most popular families of games used to describe distributed resource sharing among CDs in order to address the self-coexistence problem. In Section 7.4 we present game's models and assumptions used to describe our game theoretic frameworks. Finally, *NoRa* is presented in Sections 7.5 and *HeCtor* in Sections 7.6.

7.2 Game Theory

Game Theory (GT) is a powerful mathematical tool developed for the purpose of analyzing interactions in decision processes [35]. GT has been extensively applied in microeconomics but recently has received attention as a useful tool to design and analyze distributed resource allocation algorithms [44, 75].

In the following we present components that characterize a game and we introduce some terminology.

7.2.1 Game Definition

We can mathematically define a game in its normal form as $\mathcal{G} = \{\mathcal{N}, \mathcal{S}, \mathcal{U}\}$ where $\mathcal{G}$ is a game, $\mathcal{N}$ is the finite set of *players*; $\mathcal{S}$ is the non-empty set of *strategies*; and $\mathcal{U}$ is the set of *utility functions* or payoffs. To understand how these components describe a game let us singularly illustrate them.

7.2.2 Players

Players are entities participating in the game. The set $\mathcal{N}$ represents players that compete for shared resources and we indicate with N the cardinality of this set.

Several classifications are given to characterize a player.

(i) *Rational*, that is, each player always selects the strategy that yields it the greatest payoff.

(ii) *Myopic* or *Foresighted*, in terms of its impact on other players. Myopic players always act to maximize their immediate achievable reward. They ignore the impact of their competitors' reactions over their own performance, and determine their responses to gain the maximal immediate rewards. Foresighted players, instead, behave by taking into account the long-term impacts of their actions on their rewards. They anticipate how the other users will react, and maximize their performance by considering the responses of the other players. As consequence, foresighted players require additional knowledge about the other players to assist their decision process.

(iii) *Selfish* or *self-interested*, that is, players make their own decision independently in order to maximize their own payoff without necessarily respecting the overall system objective.

Moreover, we distinguish in Section 7.2.2 between cooperative and non-cooperative coordination mechanisms among players, while in Section 7.2.2 we describe single stage games and repeated games, which are two way how players can engage into a game.

Cooperative vs. Non-Cooperative

The type of players reflects in the coordination mechanism that can be *cooperative* [82] or *non-cooperative* [14]. They both are defined as centralized or distributed. If a centralized paradigm is used, the scalability is an issue because players require a centralized coordination entity and/or a common signaling paradigm.

Cooperative and non-cooperative terminologies can be deceptive because they may suggest that there is not space for cooperation in the former and no conflict or competition, in the latter. Instead, the difference is in the way how behaviors are imposed. In non-cooperative games, players *self-enforce* their behavior, while in cooperative games there is an *external* entity that impose the way to act. From a purely game theoretic prospective, a non-cooperative game specifies strategies that are available to players while cooperative games describe the resulting outcomes when players engage the game together in different combinations. However, cooperative players do not have global interests, but players behave cooperatively to obtain their own maximum out of the game. The major concern about cooperative games is the necessity of a centralized coordination entity or a common signaling, which however are afflicted by scalability issues. To address these issues *coalitional games* have been defined.

Coalitional games are an emerging class of games which address the scalability issue defining cooperative subgroups, i.e. *coalitions*, of the original set of players.

Thus reducing the original centralized problem in distributed sub-problems where players belonging to the same coalition cooperate but there is not cooperation between coalitions. This means that players cooperating in a coalition do not have a centralized coordination entity but they use distributed transmissions. Coalitional games can be divided based on the payoff distributions into *Transferable Utility* (TU) [84] and *Non-Transferable Utility* (NTU) [82]. In TU coalitional games, player belonging to the same coalition divide the total payoff among them. In NTU coalitional games, instead, each player has a different payoff based on its advantage of belonging to a coalition and cannot be shared with other players.

Usually, cooperative games are used where there is the need of fairly share a scarce resource among competing players. Concepts such as *bargaining games* embody specific notions of fairness and take into account the strategic interests of competing users. However, due to scalability issues non-cooperative games are preferred.

A non-cooperative game is characterized by players that make decisions independently and can easily deviate from the network protocol to seek for more benefit for themselves. This can lead to a solution that is not social efficient, because players can increase their performance degrading the others players' payoff. One of the most used technique to provide incentives for selfish players to behave cooperatively is the *payment method* [57]. The payment method introduces a way to influence players' behaviors. Players assume that there is some kind of virtual currency in the system and that each player has to pay some virtual money to the central entity based on payoffs. However, the payment method is hardly scalable due to the necessity of a centralized entity. For this reason a distributed approach is desirable also in non-cooperative games.

Single Stage Game vs. Repeated Game

Another important factor that a player has to decide is how to engage into the game. The way how a game can be played distinguish single stage games and repeated games [72]. Given the base game $\mathcal{G}$, called *stage game*, a single stage game is a game where each player engages only one time into it. On contrary, a repeated game is characterized by finite or infinite repetitions of the same stage game. Both, single and repeated games, can be simultaneous or sequential which means that players can play all in the same time instant or one after another, respectively.

Repeated games are a simplification of a bigger family of games called *stochastic games* (SGs) [106]. SGs are repeated games with probabilistic/stochastic transitions, i.e., the game moves to a new state with a certain probability. The new state depends on the previous state and the actions chosen by players. In SGs each player knows its own state and strategies, but it does not know states and strategies taken by other players. The set of strategies distinguish repeated and stochastic games. In SGs, the set of strategies depends on the current state, while in repeated games is identical at each stage.

Repeated and stochastic games have the same work flow. At the begin of each stage, the game is in a certain state. Players select their strategies and receives a reward that depends on both current state and selected strategies. We distinguish internal and external strategies. Given a player $i \in \mathcal{N}$, external strategies are strategies of the other $\mathcal{N} - i$ players, while the internal strategy is the strategy chosen by the player i. Therefore, the state transition of each player is directly impacted by its own internal actions and indirectly impacted by the external actions of all players through the resource competition.

In [114], the authors proposed a stochastic game where the decisions that need to be taken are based on the players' incomplete and asymmetric information about the environment and other players' strategies. Based on their information, each player can develop *beliefs* about the current state of the evolution of the network over the time. Based on these beliefs players can pro-actively select the optimal policy for interacting with other devices such that they maximize their utilities.

In Section 7.3 we describe several families of games more in details and we explain how different types of players can be used to model CDs.

7.2.3 Strategies

Strategies are the choices that a player can take. We assume the existence of a strategy set S_i for each player $i \in \mathcal{N}$ thus we have $\mathcal{S} = \{S_1, S_2, \dots, S_N\}$ for the set of players $\mathcal{N} = \{1, 2, \dots, N\}$. The *strategy profile* of the game [38], instead, is given by $\mathbf{s} = \{s_1, s_2, \dots, s_N\}$ where player 1 chooses strategy $s_1 \in S_1$, player 2 chooses strategy $s_2 \in S_2$ and so on. For every different combination of individual strategies, we have a different strategy profile $\mathbf{s}$ and the set of all such strategy profiles is $\mathbf{S} = S_1 \times S_2 \times \cdots \times S_N$.

7.2.4 Utility Functions

An utility function and the resulting payoff decides how good a strategy profile is. At each player $i \in \mathcal{N}$ is associated a strategy set $S_i \in \mathcal{S}$ and a payoff set $U_i \in \mathcal{U}$. Results that to a strategy $s_i \in S_i$ correspond a payoff $u_i \in U_i$, $\forall i \in \mathcal{N}$.

To properly choose the utility function several factors have to be taken into consideration. The utility function has to reflect system characteristics as well as physical properties of the environment. In addiction, the utility function has to satisfy mathematical properties with the objective to guarantee equilibrium convergence. To determine if a convergence point exist, the *Nash Equilibrium* point (NE) has been defined [86, 87].

Equilibrium Points

The NE point is an important concept in GT and in resource allocation problems. It defines a condition in which no player can benefit from changing its own strategy unilaterally, while the other players keep their strategies unchanged. A NE point correspond to a strategy profile where no player has interest to deviate. Given a strategy profile $\mathbf{s}^* = (s_i^*, s_{-i}^*) \in \mathbf{S}$, a NE point is defined as in Eq. (7.1) for each player $i \in \mathcal{N}$.

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \quad \forall s_i \in S_i \tag{7.1}$$

Where $u_i(s_i, s_{-i})$ is the utility function of player i when it uses the strategy s_i and the other players use strategies s_{-i}. By carefully designing utility function and strategies, the game can be balanced at a unique socially optimal NE, where the summation of all payoffs is maximized.

Stronger notions of equilibrium also exist. A commonly used concept is the *Strongly Dominant Strategy Equilibrium* (SDSE). In SDSE, the convergence is to a

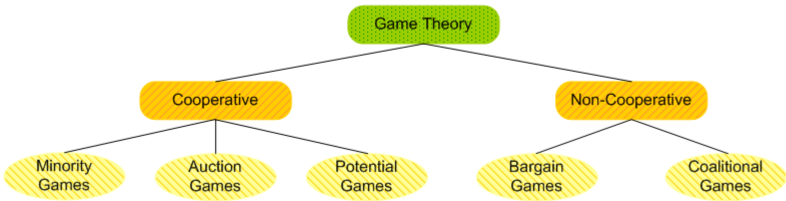

Figure 7.1: Game theoretic approaches used to address self-coexistence in CNs.

overall system optimum. Given a strategy profile $\mathbf{s}^* \in \mathbf{S}$, a SDSE is defined as in Eq. (7.2).

$$\begin{cases} \forall s_{-i} \in S_i, \forall s_i \neq s_i^*, & u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \\ \exists s_{-i} \in S_i, \forall s_i \neq s_i^*, & u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}) \end{cases} \tag{7.2}$$

This means that at least for one player the strategy has to be strictly dominant.

7.3 Families of Games

Many researchers are currently engaged in designing efficient protocols for CNs. These studies cover a wide range of issues including channel assignment, power control, call admission control and interference avoidance. A tutorial survey describing how game theory has been applied to CNs is proposed in [117] where the authors do not focus on a specific problem, but instead describe game concepts and paradigms in detail. In [35], instead, we focused on game theoretic approaches in order to address the self-coexistence problem.

We focus on game theoretic studies and methodologies related to the self-coexistence problem in CNs, and we describe network architectures and characteristics of CDs. In order to model interactions among players, several families of games with different objectives have been used. We distinguish between non-cooperative games, in Section 7.3.1 and cooperative games, in Section 7.3.2. Figure 7.1 shows an overview on game theoretic approaches used to address the self-coexistence problem in CNs.

7.3.1 Non-Cooperative Games

Non-Cooperative games are characterized by players that self-enforce to themselves a behavior, i.e. the game does not explicitly say the payoff for a group of players if they play a specific strategy profile but, instead, each player knows only its own payoff when its strategy changes. This means that the game is under *incomplete information* because each player has not information about the other players strategies.

In this section we present three families of non-cooperative games extensively used in the literature to model the cognitive device ability of reconfigure transmission parameters. We present *minority games* [102], *auction games* [104, 45, 30] and *potential games* [85, 40] giving a general introduction of the game along with examples from the literature.

The reason behind the application of different families of games is that, depending on characteristics and objectives of players, a type of game can better model a system respect to another.

Minority Games

Minority games (MGs) are characterized by a group of $\mathcal{N}$ players that have to independently decide between a binary set of strategies. Originally MGs have been proposed by Challet and Zhang [28], is a branch of GT for studying competition and self-imposed cooperation in a non-cooperative environment with limited resources. Players in a MG do not interact or negotiate with each other directly, i.e., the game is under incomplete information. The goal of MGs is to help all the players to make better decisions even without direct knowledge of other players' strategies.

The most famous MG is the *El Farol game*. In this game players have to decide if to go or not to go to the El Farol bar on Friday night. Going to the bar is enjoyable only if the bar is not too crowded and at the same time an empty bar is not a desirable condition. For this reason, if all the $\mathcal{N}$ players decide to go, thinking that the bar will be empty, then the bar will be overcrowded. In contrast, if they all decide not to go, the bar will be empty. In the same way, if all the devices in a wireless network decide to transmit on the same channel the situation is not enjoyable because no transmission is successful due to interference and if no device transmits on a channel, the resource is wasted.

In [102] the authors proposed a *Modified Minority Game with Mixed Strategies* (MMGMS) to model self-coexistence among non-cooperative players. They identify players with WRANs, which act under a non-cooperative paradigm in a distributed manner, i.e., without centralized authority or common signaling. The authors modeled the spectrum band switching game as an infinitely repeated game where WRANs try to minimize their cost in finding a clear channel, i.e. minimize the number of repetitions where transmissions fail. They propose a mixed strategy where the competing WRANs must adhere to in order to achieve a NE point. As interference model the authors used the protocol interference model, which associate to each WRAN the exact number of overlapping competitors. Their choice of interference model is given by a better tractability of the problem in spite of a more realistic model that can be obtained using the physical interference model as explained in Section 2.3. Hence, the MMGMS approach in [102] fails in terms of interference model.

Auction Games

Auction games (AGs) study how players interact in auction markets, i.e. where their strategies are a set of bids. There are many possible sets of rules for an auction and usually the objectives are efficiency and equilibrium of the bidding strategies. Several are also the type of auctions which depend on how players engage the game and the amount of "money" payed by winning players. The term *money* in AGs is used to refer actual money transaction, virtual credits or virtual currency to indicate a quantity of a good that players have to pay in order to use a resource.

Among AGs we recall First-price sealed-bid auctions, Vickrey auctions, English auctions, and Dutch auctions. Unfortunately these single unit auctions do not properly model situations where multiple winners emerge as in resource sharing problems.

Therefore, to model a channel assignment different types of auction games have to be used. One of these is the knapsack auction mechanism.

The *knapsack auction mechanism* is characterized by players that want to place objects in a knapsack. Here, the knapsack represents the available channels and the objects are the amount of data to transmit. Each player evaluates the placement of an object in the knapsack and its bid is related to the amount of money that players want to pay to use a resource. In [104], the authors modeled the coexistence of licensed and unlicensed users as a sealed-bid knapsack auction, which dynamically allocates channels to devices based on their bids. Their objective is to maximize the spectrum utilization for every player. The authors took into consideration service providers and end-users. However, the similarity with BSs and CPEs is straightforward. The disadvantage of this approach is the need of a centralized entity, which knows bids and amount of data that any device wants to send, and a common signaling to communicate with the centralized entity. Due to the necessity of a centralized entity, the sealed-bid knapsack auction in [104] is hardly applicable to CNs, where in fact the existence of a centralized entity is not guaranteed and often not desirable. A centralized entity is not guaranteed because CDs could belong to different IEEE standards, i.e., they do not follows the same paradigms but want to self-coexist. The need of a common signaling, instead, is not desirable due to the overhead that brings into the system.

Another AG used to model the spectrum allocation is the *Anglo-Dutch auction game*. In [45], the authors proposed a two rounds mechanism as follows. In the first round, N players compete for K resources using an English auction increasing their bids until $K+1$ players remain. In the second round, each remaining player submits a sealed bid at or above the bid at which the first round had stopped. Players with the K highest bids win the auction and pay either their respective bids or the highest bid. However, the Anglo-Dutch auction in [45] is inefficient to model the channel assignment problem because does not consider that more than one player can be on the same resource (channel).

AGs can be used also to model source-destination cognitive pairs, as in [30] where the authors proposed a *non-cooperative multiple-PU multiple-SU auction game*. In this game, SUs share the available spectrum of licensed PUs subject to the interference temperature constraint at each PU, i.e., they use a pollution model as described in Section 2.3. The authors proposed a distributed algorithm in which each SU updates its strategy based on local informations to converge to an equilibrium point. Their algorithm required the exchange of a small amount of information among nearby SUs.

In [30], the *ping-pong effect* is also studied. This means that the authors took into consideration configurations where free channels exist and players could infinitely jump from a free channel to another, without realizing that both the channel assignments yield the same outcome, or they could jump all together on the same channel that hence become overcrowded. In this work, the authors addressed the ping-pong effect using the no-regret learning [48], which converge to the same equilibrium given by the auction game. The basic idea under the no-regret learning is that the probability of choosing a strategy is proportional to the "regret" for not having chosen other strategies. This approach requires that SUs have knowledges regarding PUs and other SUs, hence is not suitable in every type of CN.

Another interesting game is the more competitive *Stackelberg game* [118] in which PUs choose their prices to maximize their revenue, i.e., PUs sell excessive spectrum to SUs for monetary return. However, this is in contrast with the assumption for which SUs are transparent to PUs and hence can be applied only to network where PUs and SUs are coordinated.

In summary, AGs can be used to model channel assignment problems but they also require the existence of a centralized entity, which however is not guaranteed and desirable in CNs.

Potential Games

Potential games (PGs) are characterized by a finite set of players that can engage in a finite set of strategies and the incentives of all players are mapped into one function, called *potential function*. A potential function is used to analyze NE points. In [80] has been proved that PGs guarantee the convergence to an NE point. To obtain a potential game the following requirements have to be satisfied.

(i) Players have to choose from a finite set of strategies.

(ii) Players' payoffs have to depend on the number of other players choosing the same strategy.

(iii) A potential function $Pot : \mathcal{S} \rightarrow \mathfrak{R}$ mapping strategies into real numbers has to be defined. A potential function is an increasing function that exactly reflects any unilateral change in players' payoffs, i.e. the potential function has to increase at each stage game.

(iv) Players have to engage into the game one at time in a sequential order in order to guarantee the convergence to an NE point.

PGs can be divided based on how changes in players' payoffs reflect in the potential function. We identify exact, weighted, and ordinal potential games where the variation in the potential function is respectively equal, proportional or has the same sign respect to the player's payoff improvement. An overview on potential games and their application to interference, power and waveforms game formulations for CDs is given in [83].

PGs have been used to model the channel assignment problem in CNs where the set of players ($\mathcal{N}$) is the set of CDs competing for a finite set of TVWS. Hence, the set of strategies of each player (S_i) is the TVWS chosen to transmit, i.e., the *channel*.

In [85], the point-to-point interaction between source-destination cognitive pairs is described. The authors proposed a potential game which describes a distributed and dynamic channel assignment scheme for selfish players under cooperative and non-cooperative scenarios. However, their main results are for cooperative devices only. Hence, no further explanations and proprieties are brought out for non-cooperative potential games.

The point-to-multipoint communication paradigm was modeled in [49] as a $\mathcal{N}$-player game, where the authors had as objective the maximization of the number of CPEs for each WRAN.

Conclusions on Non-Cooperative Games

Non-cooperative game theoretic approaches are used to efficiently model distributed environments where players have only local knowledge because centralized entities or global signal paradigms do not exist.

The absence of centralized or common coordination mechanisms means that non-cooperative games adapt quickly to the surrounding environment. However, they can lead to a bad equilibrium and hence poor performance due to the lack of mechanisms that soften the selfish behavior of CDs. In order to address the bad equilibrium problem in Section 7.3.2 we analyze cooperative games.

7.3.2 Cooperative Games

Cooperative games define the outcome for each player if a group of players follows a specific strategy profile. These games analyze situations where the players' objectives are partially cooperative and partially conflicting, i.e. players have an interest in cooperating in order to achieve the greatest possible total payoff but at the same time they have conflicting goals in sharing the resources obtained.

Generally the game theoretic approaches used to model cooperative games are *bargaining games* [49, 24, 45] and *coalitional games* [18]. The former uses a centralized approach, meanwhile the latter can be centralized or distributed.

In the following we describe bargain and coalitional games focusing on their application to solve spectrum management problems.

Bargain Games

Bargain games (BGs) are situations where players want to reach an agreement regarding how to distribute resources. Each player prefers the agreement that maximize its own payoff. In BGs, players analyze their own rewards if the agreement is reached or if players fail to do so. Hence, the only analysis of the single player is not sufficient to model the entire bargain process.

A BG is defined using a set of possible agreements (set of strategies) and a disagreement point. Players ask for portions of the resource under consideration. If the sum of the requested portions of the resource is less than the total resource, then all players obtain the resource otherwise they obtain nothing. The amount of requested resource is the strategy of each player, while results of the bargaining process and disagreement point are payoffs. Therefore, a BG forces cooperation among players regarding on how to divide a resource. In self-coexistence problems players are CDs, while the resources to share are channels.

A common concept used to model bargaining interactions is the *Nash bargain solution* (NBS) [81]. A NBS provides a way to divide a resource and defines four axioms that a solution has to satisfy to guarantee the convergence to an optimal equilibrium point.

(i) Linearity.

(ii) Independence of irrelevant alternatives.

(iii) Symmetry.

(iv) Pareto optimality.

Let us consider a two players game to explain how a bargain game works. Player a makes an offer, say $x(a)$, to player b, which can either accept or reject the offer. If player b accepts the offer, this offer takes place. Otherwise, player b applies a discount on its utility and makes an offer, say $x(b)$, to player a. Note that $x(a)$ and $x(b)$ are in the range $[d, z]$, where z is the total resource and d is the disagreement point, i.e. the minimum obtainable utility by both players. Moreover, we have that the offers of the two players have to satisfy $x(a) + x(b) \leq z$. The bargain game continues until player a or player b accepts the offer, that is, maximize $x(b)$ and $x(a)$.

In [49], the authors proposed a cooperative scheme using the NBS. Given a set of channels and a set of CPEs, they assign channels to WRANs so that the number of CPEs managed by each WRAN is maximized. They show that using the NBS, the number of CPEs into each WRAN increases compared to a game where players do not cooperate. This means that each WRAN uses resources more efficiently. Unfortunately the game presented in [49] does not scale because its complexity grows when the number of channels and/or the number of CPEs increases. In fact, the authors used a centralized approach where each WRAN has to find all the possible channel assignments.

To reduce the complexity of a centralized approach, the authors in [24] proposed local bargaining groups. They addressed the spectrum management problem in CNs considering that players self-organize the network into bargaining groups. They shown how their approach significantly reduces the algorithm complexity compared to graph-coloring solutions. The problem with this approach is the implicit willingness of collaboration among devices which is not realistic in practical systems.

A different variation of BGs used to model a distributed implementation of dynamic spectrum allocation is the *Rubinstein-Stahl bargaining model* [97]. This bargaining approach consists of a game where players want to reach an agreement on how to share a resource and at each negotiation step where they fail in reach an agreement, part of the resource is wasted for all players. This approach has been used in [45] where the authors considered a decentralized game to reflect the decentralized nature of the network, and hence they modeled a bargain game under incomplete information. This game theoretic model can lead to a waste of resources and hence is not a desirable approach.

In conclusion, BGs are mainly centralized approaches which require communication among CDs, but how previously explained common communication paradigms are not guaranteed in CNs. Moreover, BGs could produce situations where a non-negligible amount of resources is wasted. In fact, until players do not reach an agreement point, the resources are not utilized. Hence, properly modeled BGs can be effective in configurations where the time is not important, however the channel assignment problem is not among these.

We now analyze coalitional games aimed at reducing the computational complexity of bargaining centralized approaches. Moreover, the problem of reaching an agreement point in coalitional games does not take place.

Coalitional Games

Coalitional games (CGs) have been extensively used to model distributed cooperation among players sharing resources.

CGs are characterized by the formation of cooperative *sub-groups* of devices, referred to as *coalitions* [82]. The formation of coalitions includes two activities: *structure generation* and *optimization*.

(i) Structure generation is the formation of coalitions such that players, within each coalition, coordinate their activities, but players do not coordinate between coalitions. Therefore, the number of coalitions and the cardinality of each subgroup need to be defined.

(ii) The optimization problem that has to be solved is the maximization of the difference of gain given by the cooperation minus the cooperation cost.

In a coalitional game, we call: $\mathcal{C}$ the coalition structure; C the number of coalitions; $\mathcal{C}_p$ one of the coalitions with $p \in \{1, \dots, C\}$; and C_p the size of coalition $\mathcal{C}_p$.

A coalition structure can be characterized by overlapped or disjoint coalitions, and partial or exhaustive involvement of players in the coalition structure. When coalitions are disjoint and exhaustive, they are called *partitions*. This means that given two partitions, $\mathcal{C}_p$ and $\mathcal{C}_q$, where $p, q \in \{1, \dots, C\}$ and $p \neq q$, we have $\mathcal{C}_p \cap \mathcal{C}_q = \emptyset$ and $\bigcup_{p=1}^{C} \mathcal{C}_p = \mathcal{N}$. Where $\mathcal{N}$ is the set of all the players.

To clarify the difference between coalitions and coalition structure [101], let us consider $\mathcal{N} = \{1, 2, 3\}$, disjoint coalitions and exhaustive involvement of players. In this case we have $2^N - 1 = 7$ possible coalitions: $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{1, 2, 3\}$ and 5 coalition structures: $\mathcal{C} = \{\ \{\{1\}, \{2\}, \{3\}\},\ \ \{\{3\}, \{1, 2\}\},\ \{\{2\}, \{1, 3\}\},\ \ \{\{1\}, \{2, 3\}\},\ \ \{\{1, 2, 3\}\}\ \}$.

Moreover, every coalition $\mathcal{C}_p$ is characterized by a *coalition value*, denoted by $v(\mathcal{C}_p)$, which quantifies the coalition's payoff in a game. The coalition value is particularly important because it determines form and type of the game.

Let us use the previous example to illustrate what a coalition value is. We consider, for sake of simplicity and without loss of generality, a game where coalition values are an a-priori knowledge and are given as follows.

(i) Coalitions of single players: $v(\{1\}) = 0$; $v(\{2\}) = 0$; $v(\{3\}) = 0$.

(ii) Coalition of players' pairs: $v(\{1, 2\}) = 150$; $v(\{1, 3\}) = 150$; $v(\{2, 3\}) = 150$.

(iii) Grand coalition: $v(\{1, 2, 3\}) = 120$.

This means that players on their own have a null payoff, any pair of players has a payoff equal to 150 and the coalition containing all the players, called *grand coalition*, has an overall payoff equal to 120. Note that the coalition value has to be divided among the members of the coalition. Assuming that players divide the coalition value in equal parts among them, the individual payoff for the members of the grand coalition is lesser than the payoffs of the members of coalitions with two players. This condition is due to cooperation costs, therefore players have the interest to cooperate but they have to consider also their costs in cooperation.

The *grand coalition* concept is strictly related to CGs and has a considerable importance. Often the grand coalition is seen as the optimal solution because no coalition has a value greater than the sum of all the players' payoffs. However, in games where a coalition brings gains to its members, but gains are limited by the cost in forming the coalition, the grand coalition is seldom the optimal structure. In fact, large coalitions increase the complexity of the game due to high synchronization and communication costs.

Coalitional games can be divided based on the payoff distributions into *Transferable Utility* (TU) [84] and *Non-Transferable Utility* (NTU) [82]. The TU property implies that the total payoff can be divided in any manner among the coalition members. In contrast, in an NTU game each player has a different payoff based on its advantage of belonging to a coalition and cannot be shared with other players in the same coalition. Therefore, the value of a coalition in an NTU game is a vector of payoffs in the set of real numbers, $v(\mathcal{C}_p) \subseteq \mathfrak{R}^{C_p}$, where each element $v_i(\mathcal{C}_p) \in v(\mathcal{C}_p)$ represents the payoff that player $i \in \mathcal{C}_p$ can obtain within the coalition.

In [18], the authors proposed a repeated coalitional game to model the competitive behavior between independent wireless networks in allocating a common shared channel. Players are wireless networks, which play repeatedly in a resource sharing games without direct coordination or information exchange. Strategies determine whether competing networks cooperate or ignore the presence of other networks. Players have to make decisions about when and how often to attempt to access the wireless medium in order to maximize their observed utility. The authors did not consider channels, but they proposed a distributed coordination algorithm to schedule transmissions on the wireless medium.

CGs are also used to address the self-coexistence problem from a power allocation prospective [33] or using a joint power/rate control and channel assignment solution [113].

In conclusion, CGs are a valuable instrument to model channel assignment algorithms and hence to guarantee self-coexistence among CDs. However, CG complexity increases when changes occur in the surrounding environment due to a slower adaptability compared to non-cooperative games. They should therefore be applied in scenarios where PUs do not rapidly change transmission channels and where slower adaptability is not an issue.

Conclusions on Cooperative Games

Cooperative games perform better than non-cooperative games and avoid bad equilibrium conditions which non-cooperative games suffer from. However, they have higher computational complexity than non-cooperative games. This complexity can be kept low adopting a distributed cooperative approach which at the same time soften the selfish behavior of cognitive devices. However, distributed cooperative approaches suffer when rapid changes occur in the network surrounding environment.

7.4 Game Models and Assumptions

A game is defined as $\mathcal{G} = \{\mathcal{N}, \mathcal{S}, \mathcal{U}\}$, hence in order to present our game theoretic frameworks in Section 7.5 and 7.6, we define the game components and its work flow.

We consider a set $\mathcal{L} = \{1, 2, \dots, L\} = \mathcal{N} \cup \mathcal{M}$ representing the total number of CDs, where $\mathcal{N}$ is the set of BSs and $\mathcal{M}$ is the set of CPEs. For each BS $i \in \mathcal{N}$, we define $\mathcal{M}_i \subseteq \mathcal{M}$ as the subset of CPEs connected to it where $\bigcup_{i\in\mathcal{N}} \mathcal{M}_i = \mathcal{M}$ and $\mathcal{M}_p \cap \mathcal{M}_q = \emptyset \; \forall p, q \in \mathcal{N}$ and $p \neq q$. We consider WRANs as the players of our games and, by considering that each BS manages an entire WRAN, we reduce the set of players to $\mathcal{N}$. Note that hereafter we use the terms BS and WRAN interchangeably to refer to players.

Strategies are the spectrum opportunities, i.e. *channels* or TVWS, identified in the CN. We refer the set of channels as $\mathcal{K} = \{1, \dots, K\}$ and we define for each WRAN i the subset of $\mathcal{K}$, called the *candidate channel set* and indicated by $\mathcal{K}_i$. The candidate channel set $\mathcal{K}_i$ contains all the channels where the WRAN i does not sense PUs, i.e. it is the set of suitable channels. Hence, each WRAN chooses its transmission channel from a different candidate channel set which is *time-variant* and depends on the surroundings of the specific WRAN. The strategy for each player i is expressed as binary variables s_i^k defined as in Eq. (7.3).

$$s_i^k = \begin{cases} 1 & \text{if WRAN } i \text{ chooses channel } k \\ 0 & \text{otherwise} \end{cases} , \tag{7.3}$$

$\forall k \in \mathcal{K}_i$.

The utility function is chosen in order to represent the quality of a channel and hence is captured as a measurement of the radio environment using the SINR model as presented in Section 2.3. A CD senses any other device as a competitor, which if tuned on the same channel creates interference. Hence, WRANs compete for TVWS with neighboring WRANs in order to avoid co-channel interference and hence to guarantee self-coexistence among BSs and CPEs belonging to different WRANs. Unfortunately, in our model we cannot identify which device is transmitting in the interval $[t_1, t_2]$ of Eq. (2.1). Hence, we dinamically consider the worst case where the device causing the largest interference is the one that is transmitting. In order to extend the interference formulation considering the SINR for each channel $k \in \mathcal{K}$, we obtain Eq. (7.4).

$$SINR_{i,j}^k(t) = \frac{P_{i,j} \cdot x_i^k(t)}{\sum\limits_{h\in\mathcal{T}(t)\setminus i} P_{h,j} \cdot x_h^k(t) + \mathcal{W}} \tag{7.4}$$

$\forall i \in \mathcal{N}$, $\forall j \in \mathcal{M}_i$, $\forall k \in \mathcal{K}$; where the binary variable $x_l^k(t)$ is defined as in Eq. (7.5).

$$x_l^k(t) = \begin{cases} 1 & \text{if } l \text{ is transmitting on channel } k \\ & \text{in the time interval } t \\ 0 & \text{otherwise} \end{cases} , \tag{7.5}$$

$\forall l \in \mathcal{L}$.
Where for the sake of brevity we write t instead of $[t_1, t_2]$. Moreover, we indicate with γ_l the capture threshold for every CD, i.e., $\forall l \in \mathcal{L}$. In practice, we want to achieve a certain bit error rate performance at each CD.

Notice that we distinguish between Eq. (7.5) and Eq. (7.3) because each WRAN i assigns a value to x_l^k $\forall l \in \mathcal{L} \setminus i$ and $\forall k \in \mathcal{K}_i$ in agreement with the sensed environment, whereas s_i^k is the strategy that each WRAN chooses. Note that the relation $\sum_{k\in\mathcal{K}} s_i^k = 1$ holds for each player i because it is supposed to choose just one channel for its transmissions. We have also that if $s_i^k = 1$, then $s_i^k = x_j^k \; \forall j \in \mathcal{M}_i$ because each CPE needs to be on the same channel with respect to the BS.

The performance of the channel assignment algorithm depends significantly on the chosen utility function $u_i \in U_i$ because it characterizes the preference of a player for a particular channel. As shown in [40], spatial reuse maximization is a better choice for utility in CNs compared to the commonly used interference minimization. In fact, a CN is characterized by a high resource variability in time and space and the spatial reuse maximization can converge faster than the interference minimization and can soften the players' selfish behavior. Moreover, the *ping-pong effect* underlined in [30] is avoided because CDs look for channels that respect the capture threshold instead of channels where the interferences are minimized. In other words, a channel free of interference is chosen only if there are no channels with the SINR greater than the capture threshold.

Equation (7.6) shows the utility function used in Section 7.5 and 7.6.

$$u_i = \min_{k\in\mathcal{K}_{i,\alpha_i}} \left[\sum_{j\in\mathcal{M}_i} SINR_{i,j}^k + \sum_{j\in\mathcal{M}_i} SINR_{j,i}^k \right], \tag{7.6}$$

$\forall i \in \mathcal{N}; \alpha_i \in (0, 1]$.
Where $\mathcal{K}_{i,\alpha_i}$ denotes the set of channels in $\mathcal{K}_i$ satisfying $SINR_{i,j}^k \cdot \alpha_i \;\geq\; \gamma_j$ and $SINR_{j,i}^k \cdot \alpha_i \;\geq\; \gamma_i$, $\forall j \in \mathcal{M}_i$. Here α_i is a tunable parameter that represents the amount of tolerable interference suffered by a WRAN and is equal to $\alpha_i = \frac{K_{i,\alpha_i}}{K_i}$ subject to $K_{i,\alpha_i} \geq \frac{K_i}{2}$ where K_{i,α_i} and K_i are the cardinalities of $\mathcal{K}_{i,\alpha_i}$ and $\mathcal{K}_i$, respectively. This means that each WRAN i chooses α_i according to the sensed interference in its surroundings. A WRAN that senses a high level of interference chooses α_i approaching 1. In contrast, α_i approaching 0 is chosen if interferences are low.

The overall payoff of the game is given by $u = \sum_{i\in\mathcal{N}} u_i$ and it depends on the power and channel chosen by each player as well as those of others. Several techniques of power control have been proposed in the literature. In [105], the authors proposed a power control scheme for non-cooperative CDs where communication between transmitter-receiver pairs is assumed in order to know the expected SINR of receivers. A different approach used in the literature is the pricing scheme which requires a centralized entity in order to decide rewards and prices. However if a centralized entity or a common information exchange mechanism among WRANs are not defined, power control techniques and pricing schemes cannot be applied. In fact as proved in [96], if WRANs cannot exchange control messages, there is no way of limiting their transmission power and as consequence, each WRAN transmits at the maximum admissible power (P^{Max}) defined by the MAC layer. Moreover, the same authors proved that each WRAN transmits at P^{Max} at the NE point. Consequently, the SINR does not depend on the transmission power but it depends only on the transmission channel selected.

The *work flow* for ours game theoretic frameworks for CNs is as follows.

(i) At the beginning of the game, each WRAN senses the environment and dynamically chooses the channel that minimizes the interference in the set of candidate channels.

(ii) During the game, the WRANs sense the environment and make decisions based on the game model.

(iii) The game ends when all the WRANs are successful in capturing channels where the interference is under the capture threshold, and is re-initiated when channels occupancy changes are sensed in the surrounding environment.

7.5 NoRa: Non-cooperative Repeated game

NoRa is a multi-player non-cooperative repeated game which models the self-coexistence problem in CNs. It is characterized by a set of homogeneous players, $i \in \mathcal{N}$, that have to choose from a finite set of strategies $\mathcal{S}$ (the transmission channels), and their payoffs depend on the number of other players that choose the same channel. In this description we identify the characteristic of a potential game as described in Section 7.3.1.

Model *NoRa* as a *potential game* means define the *potential function* which exactly reflects any unilateral change in the utility function of each player. We define the potential function as in Eq. (7.7) where the sum reflects the increase in the overall utility function covering all the users at each stage of the repeated game.

$$Pot = \sum_{i \in \mathcal{N}} u_i \tag{7.7}$$

How players engage in the game is a crucial point in a non-cooperative environment. In order to address this problem we define a novel *backoff mechanism* as described in Section 7.5.1.

Considering a non-cooperative environment, where stages are not synchronized, and adding to it the innovative backoff mechanism, we assume that WRANs play the game in a sequential order. It is worth pointing out that classifying our game as a potential game guarantees the convergence to an NE point [80].

7.5.1 Backoff Mechanism

In CNs, no synchronization or central entity is assumed to coordinate the playing order and players choose a strategy based only on the surrounding environment. We adopt a backoff mechanism so that the neighboring players play in a sequential order and at each stage the potential function increases. An increase in the potential function is derived from an increment in the utility function, which due to mathematical properties corresponds to a decrement in the interference. The backoff mechanism introduced is characterized by a backoff window (BW_i), defined in Eq. (7.8), and a backoff counter (BC_i), randomly chosen in the backoff window and decreased by one at each stage. Thus,

$$BW_i = [F_i,\ F_i + n_i],\ \text{where} \tag{7.8}$$

$$F_i = \begin{cases} \min\left[\min\limits_{j\in\mathcal{M}_i}\frac{P_{i,j}}{\gamma_j}, \min\limits_{j\in\mathcal{M}_i}\frac{P_{j,i}}{\gamma_i}\right] & \text{if} \sum\limits_{h\in\mathcal{T}\setminus i} P_{h,j}+\mathcal{W}<1 \\ \min\left[\min\limits_{j\in\mathcal{M}_i}\frac{SINR^k_{i,j}}{\gamma_j}, \min\limits_{j\in\mathcal{M}_i}\frac{SINR^k_{j,i}}{\gamma_i}\right] & \text{otherwise.} \end{cases} \tag{7.9}$$

Where n_i is the number of WRANs that a player i senses.

Each WRAN $i \in \mathcal{N}$ computes its backoff window, BW_i reflecting the amount of interference on the channel used for transmission that depends on the number of WRANs in its surroundings. Thus, the higher the interference, the more frequently the WRANs try to change their channels. When BC_i reaches zero, the WRAN senses the environment and re-allocates its radio resource. After the reallocation, BW_i and BC_i are computed as in Eq. (7.8).

7.6 HeCtor: Hedonic Coalitional Formation Game

It is well known that cooperation can improve performance in wireless networks from the physical layer [74] to the networking layer [47]. However, obtaining cooperation among devices in CNs is not trivial because, in a distributed environment, a device can easily deviate from seeking more benefits for itself. In order to provide *incentives* to selfish players to behave cooperatively, several game theoretic models have been proposed in the literature. We choose to use the coalitional game model because it describes the trade off between the advantages due to cooperation and the disadvantages due to cooperation costs.

A coalitional game is characterized by payoff distributions, coalition characteristics, and coalition value properties, as described in Section 7.3.2. In *HeCtor*, we quantify the payoff with the help of the interference suffered by each player, which clearly cannot be divided among players. Hence our game is formulated as an NTU game. Moreover, each WRAN plays the game and belongs to only one coalition in order to limit the complexity of the channel assignment. Hence, in our coalitional game, coalitions are disjoint and exhaustive. Lastly, we identify the cost in terms of the channel assignment complexity and consider the communication costs among BSs as negligible, because the BSs communicate through a wired transmission. In the same way, the time required by a WRAN to change the transmission channel is not considered. This is because in CNs, the switching of channels is necessary due to the existence of PUs that can appear on a channel at any moment. Thus our algorithm does not increase this complexity.

Based on the previous considerations and following the game theory terminology, we classify *HeCtor* as a hedonic coalitional formation game and in Section 7.6.1 we describe the mechanism adopted to generate the coalition structure. In Section 7.6.2, instead, the game implementation of *HeCtor* is described.

7.6.1 Coalition Formation

To generate a coalition structure, a centralized approach can be used; however, such an approach is $\mathcal{NP}$-hard [101] because it entails iterating over all the partitions of the player set $\mathcal{N}$. In fact, the number of coalition structures of a set $\mathcal{N}$ grows exponentially with the number of players as explained in Section 7.3.2. Therefore, a distributed coalition formation is desirable. The approaches used for distributed coalition formations vary, such as heuristics [101], Markov chains [94] and bargaining theories [10]. However, these studies consider only TU games and are not applicable to an NTU game such as *HeCtor*.

In this work we follow the general rules proposed in [9] to implement a coalitional game.

(i) Define a suitable *order* for comparing coalition structures.

(ii) Define two simple rules to *merge* or *split* coalitions.

(iii) Define a notion of partition *stability*, i.e., equilibrium.

To compare coalition structures, each player builds preferences over all possible coalitions that can be formed in order to decide which coalition it prefers in terms of received payoffs by a coalition or weights that each player gives to other players. Several well-known sort criteria have been defined to hold different properties [9] such as utilitarian, Nash, egalitarian for TU games, majority and Pareto for NTU games. *HeCtor* considers selfish players that makes autonomous decision for their own benefit, i.e., preferences are based on the individual player's payoff rather than the coalition value as in the sort criteria for TU games that cannot be applied.

Equation (7.10) defines the *majority order* which however cannot correctly capture the coalition value in our game because players are selfish and have as objective the maximization of their own payoff, without taking care of the overall game payoff.

$$(h_1, \ldots, h_N) \succ_{Majority} (l_1, \ldots, l_N) \tag{7.10}$$
$$\text{iff } \forall i \in \{1, \ldots, N\} \; |\{i : h_i > l_i\}| > |\{i : l_i > h_i\}|$$

In contrast the *Pareto order* defined in Eq. (7.11) correctly captures the properties of our game.

$$(h_1, \ldots, h_N) \succ_{Pareto} (l_1, \ldots, l_N) \tag{7.11}$$
$$\text{iff } \forall i \in \{1, \ldots, N\} \; h_i \geq l_i \text{ and } \exists i \in \{1, \ldots, N\} : h_i > l_i$$

In fact, given two coalitions $\mathcal{C}_p$ and $\mathcal{C}_q$ to the same players, $\mathcal{C}_p$ is preferred over $\mathcal{C}_q$ by Pareto order if *no player is hur*t and at least for one player $v_i(\mathcal{C}_p) > v_i(\mathcal{C}_q)$.

Once an order has been defined, a player has to decide whether to join or leave a coalition using the merge and split rules. The *merge rule* states that two coalitions decide to merge into a single one if this new coalition is preferred by the players over the Pareto order. Similarly, the *split rule* asserts that a coalition splits into smaller coalitions if at least one player prefers the new coalitions to the larger one, i.e. a selfish player decides to move from its current coalition to a new one, regardless of the effect that this move on the players belonging to the old coalition.

Finally, the notion of equilibrium is given by [9] which shows that, if the players play in a sequential order, the final coalition structure is independent of the sequence of merges and splits. To obtain this, we adapt the same backoff mechanism described by Eq. (7.8).

7.6.2 Coalitional Game Implementation

We now illustrate the three steps needed to implement *HeCtor*.

(i) In the first step, a player decides whether to join or leave a specific coalition.

(ii) In the second step the coalition that a player wants to join decides whether or not the new member decreases its utility function.

(iii) Finally, the new channel allocation is determined.

To start the game we consider two ingredients: (i) geographical and hence SINR constraints; and (ii) commercial constraints. By a geographical constraint we mean the geographical location of each player (i.e., BS or WRAN). We start from the assumption that the BSs closer in space create a mutual interference that is negatively reflected in the SINR and hence are interested in acting as a single unit, i.e., belonging to the same coalition. By commercial constraints, we mean that the BSs belonging to the same ISP are interested in cooperating in order to reduce the interference, i.e., improve their performance.

We start with an initial coalition structure $\mathcal{C}$ where the BSs belonging to the same ISP and located in the same geographical area form a coalition. Algorithm 4 proposes a criterion to establish the initial coalition structure. However, the criterion chosen does not influence our game. In fact, using the merge and split rules, the initial coalition structure is only a starting point and the final structure does not depend on it [9].

Algorithm 4 Initial Coalition Formation

procedure INITIAL COALITION FORMATION($\mathcal{N}$)
    $\mathcal{C} = \emptyset$ // *Coalition structure.*
    $\mathcal{B} = \mathcal{N}$ // *Set of players not belonging to any coalition.*
    maxC // *Maximum coalition dimension size.*
    maxΔ // *Maximum distance among devices belonging to the same coalition.*
    $\Delta_{\mathcal{C}_p,i}$ // *Distance between the furthest player in coalition $\mathcal{C}_p$ and player i.*
    for all $i \in \mathcal{B}$ **do** // *Add the player i to an existing coalition.*
        for all $p \in \{1, \ldots, C\}$ **do** // *For all the existing coalitions.*
            if $C_p <$ *maxC* **and** $\Delta_{\mathcal{C}_p,i} <$ *max*Δ **then**
                $\mathcal{C}_p = \mathcal{C}_p \cup i$; $\mathcal{B} = \mathcal{B} - i$
            end if
        end for
        if $\mathcal{B} \cap i \neq \emptyset$ **then** // *If i cannot be added to any existing coalition.*
            $\mathcal{C}_{C+1} = \{i\}$ // *Create a new coalition.*
            $\mathcal{C} = \mathcal{C} \cup \mathcal{C}_{C+1}$ // *Add the coalition to the coalition structure.*
        end if
    end for
end procedure

Given the initial coalition structure, each BS i computes BW_i and BC_i as specified in Eq. (7.8). When $BC_i = 0$, the BS i follows the merge and split rule to decide whether to:

(i) Leave the current coalition and form a new coalition on its own.

(ii) Leave the current coalition and join another existing coalition.

(iii) Stay with the current coalition.

Hereafter the current coalition is referred to as $\mathcal{C}_{Old}$ and the future coalition as $\mathcal{C}_{New}$.

A player makes a decision based on its worth. Players should evaluate their involvement in each possible coalition. However, in practice, this exhaustive search results in a waste of resources because players are not interested in belonging to a coalition where the members are geographically located further away. In fact, if players are outside a certain range, their cooperation has no effect on the mutual interference. To clarify, let us take two BSs p and q in a range of 100 km, the mutual received power is $P_{p,q} = \frac{P^T(\lambda/4\pi)^2}{100^\kappa}$. Hence, the received power is about $\frac{1}{10^5}$ of the original power and BSs do not interfere with each other. Therefore, a player evaluates its worth considering only the coalitions that have members in its neighborhood.

For each of these coalitions, a BS i computes its gain (g_i) as given in Eq. (7.12) and its complexity (o_i) as in Eq. (7.13).

$$g_i = \min_{i \in \mathcal{C}_{New}} (\min(SINR^k_{i,j}, SINR^k_{j,i})) - \min_{i \in \mathcal{C}_{Old}} (\min(SINR^k_{i,j}, SINR^k_{j,i})) \quad (7.12)$$

$\forall i \in \mathcal{N}, j \in \mathcal{M}_i$ and $k \in \mathcal{K}_i$.

$$o_i = \frac{{C_{Old}}^K}{{C_{New}}^K} \quad (7.13)$$

Where g_i is given by the minimum SINR that the BS can obtain in its WRAN in the future coalition minus the minimum SINR in the current coalition. If $g_i < 0$, then there is no advantage and the player does not want to belong to the future coalition. Instead, o_i is given by the channel assignment complexity of the current coalition over that of the future coalition. To evaluate the channel assignment complexity, we consider the number of members of a coalition (C_{Old} and C_{New}) raised to the number of channels (K). This complexity represents all the possible BS-channel combinations. If $o_i > 1$ then the computational cost decreases, whereas if $o_i < 1$, the complexity increases. Among all the coalitions, the players ask to join the $\mathcal{C}_{New}$ where $\frac{g_i}{o_i}$ is maximized and greater than 1, if any. This ratio states the reduction of gain led by the channel assignment complexity.

Afterwards, the members of $\mathcal{C}_{New}$ have to reach an agreement, i.e., they need to evaluate Eq. (7.12) and Eq. (7.13) when the new player joins the coalition. Each player $i \in \mathcal{C}_{New}$ evaluates the SINR obtained if the player requiring to join the coalition is tuned on its current channel k. If $\min SINR^k_{i,j} < \gamma_j$ or $\min SINR^k_{j,i} < \gamma_i$ with $j \in \mathcal{M}_i$ for a player i, then the player that asks for the union is added to the coalition, because it can cause disruptive interference to the coalition. Otherwise the asking player is only added if $\frac{g_i}{o_i} > 1 \ \forall i \in \mathcal{C}_{New}$. Clearly, this choice is due to the fact that a coalition accepts a new member only if it can benefit from the union. Observe that the players' choices depend on the interference suffered and the channel assignment complexity.

In conclusion, if a player makes a decision that modifies the coalition structure, the new channel allocation needs to be computed.

No amount of experimentation can ever prove me right; a single experiment can prove me wrong.

Albert Einstein

8 Performance Evaluation

8.1 Introduction

In this Chapter we evaluate and analyze the performance of *NoRa* and *HeCtor* using an ad-hoc event-driven simulator in C++. To the best of our knowledge, this is the first event-driven simulator for CNs.

8.1.1 Outline of the Chapter

In Section 8.2 we describe experimental setup, parameters and metrics used in our ad-hoc event-driven simulator for CNs. Then, in Section 8.3 we compare several game theoretic frameworks. Section 8.3.1 presents a comparison between cooperative and non-cooperative game theoretic frameworks in term of fairness and Section 8.3.2 compares our frameworks with the MMGMS algorithm.

8.2 Network Simulator

Our network simulator is devided from ns-2 [53] and models the behavior of CNs measuring the interaction between network cognitive devices.

8.2.1 Experimental Setup

The experiments were conducted considering the DTVs in the square area of 200×200 km^2 around San Francisco as depicted in Fig. 8.1. We consider two types of PUs: DTVs and microphones. We assume 10 microphones with a transmission range of 10 m, and 20 DTVs with a transmission power in $[1,\ 5000]$ kW [50] depending on the channel used. SUs are represented by BSs and CPEs that create WRANs in number equal to the number of BSs.

How to optimally place the BSs is a well-known issue in wireless networks. A placement algorithm has to take into consideration the ability to provide sufficient signal strength for the whole planning area, called *coverage*, and the ability to provide sufficient radio resources for all users that need to be served, called *capacity planning*. The placement of BSs is strictly related to the antenna configuration and the radio resource management, such as transmission power, obstacles in the air, and propagation properties. There are many studies concerning this problem

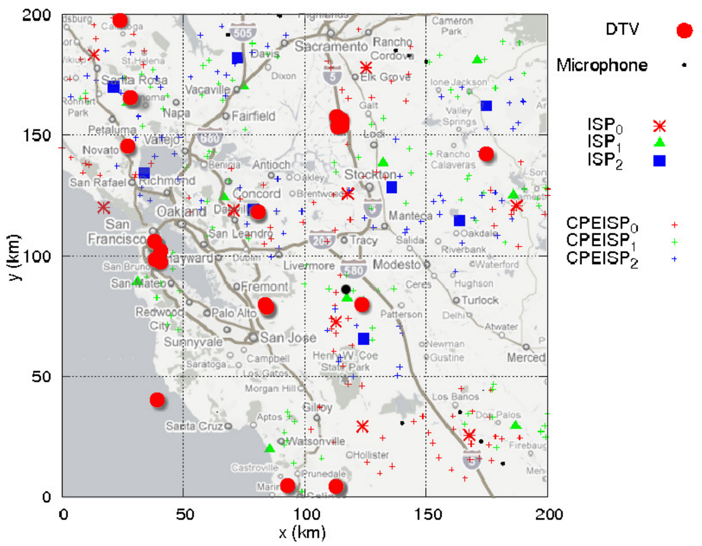

Figure 8.1: Network topology mapped on the San Francisco Bay area [50].

[8, 115]. We take as a candidate reference the solution proposed in [115] and place the BSs based on the user density and traffic demand, considering that the number of BS potential site locations is limited due to territorial characteristics and health issues. Notice that, in the final placement, WRANs belonging to different ISPs can be partially or completely overlapped geographically.

Figure 8.1 shows the $N = 28$ BSs that belong to 3 different ISPs, $\mathscr{I} = \{ISP_0, ISP_1, ISP_2\}$ considered in our experiments. Specifically, ISP_0 and ISP_1 have 10 BSs, meanwhile ISP_2 has 8 BSs. To simplify the simulation environment and comparisons, but without loss of generality, we take into consideration only download traffic flows (from BS to CPEs) and the same number of CPEs in every WRAN. The traffic flow between BS and CPE can be *web* [79] or *voice over IP* (VoIP).

Table 8.1: Simulation Parameters.

Parameter	Description
Web traffic	ON-OFF traffic [79]
VoIP traffic	uninterrupted traffic with packet size 35 bytes, of duration 20 ms
BS transmission range	$[20,\ 23]$ km
Maximum coalition dimension	maxC $= 5$
Modulation scheme	$3/4$ 16QAM hence 11.23 Mbit/s [56]
Path loss exponent	$\eta = 2$ propagation in free space
Signal wavelength	$\lambda = 0.48622$
Background noise power	$\mathcal{W} = -95$ dBm
Maximum transmission power	$P^{Max} = 70$ dB
Capture threshold	$\gamma_l \in [3,\ 4]$, $l \in \mathcal{L}$

Simulation parameters and simulator characteristics of our ad-hoc event-driven simulator are presented in Table 8.1. As a packet scheduler, we choose the *Deficit Round Robin* (DRR) and we schedule the traffic in a strictly prioritized way. The

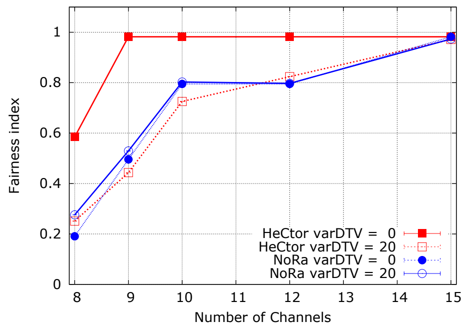

Figure 8.2: Fairness index when the number of channels increases from 8 to 15 and there are 70 web flows and 70 VoIP flows.

VoIP traffic is served first due to its delay constraints. Concerning the MAC layer, we take IEEE 802.22 standard as reference where the frame is 10 ms and consists of a downlink sub-frame, uplink sub-frame and Coexistence Beacon Protocol (CBP) burst. The downlink and uplink are reserved for 8.3 ms that can be arbitrarily divided. We choose to have $1/3$ for the uplink and $2/3$ for the downlink. The simulation time is set to 50 s and the warm up time is 10 s.

8.2.2 Metrics

We use the following metrics to evaluate the three game theoretic frameworks in Section 8.3.

(i) Throughput in bytes/s for each WRAN. It is defined as the number of bytes received by the CPEs in a WRAN in the unit of time.

(ii) Overall network throughput in bytes/s. It is the sum of the throughputs of all the WRANs in the unit of time.

(iii) Fairness, which is ratio between the minimum throughput over the maximum throughput achieved by all the WRANs.

8.3 Simulation Results

The channel assignment algorithms implemented for comparison are our game theoretic frameworks,*NoRa* and *HeCtor*, and the existing scheme, MMGMS [102]. For *HeCtor*, we set maxC $= 5$ because it satisfies the time constraint required by the IEEE 802.22 channel move time (two seconds). In fact using experimental values we observe that coalitions of 5 members determine the new channel assignment in less than 2 s.

In order to implement the protocol interference model required by the MMGMS, we assume that the interference range is equal to twice the transmission range. Moreover, we set the MMGMS channel switching probability to 0.5. Each algorithm

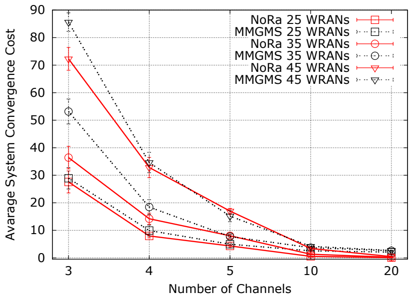

Figure 8.3: Comparison between *NoRa* and the MMGMS scheme in terms of average system convergence cost.

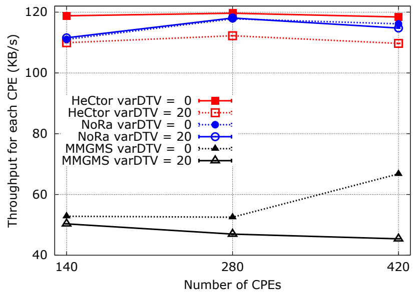

Figure 8.4: Average throughput for each CPE when the number of flows in the network increases from 140 to 420. Number of channels is 8.

assumes all channels with the same characteristics, i.e., the BSs do not distinguish between the channels.

8.3.1 Cooperative vs. Non-Cooperative

Figure 8.2 presents the comparison between *NoRa* and *HeCtor* in terms of the fairness index described in Section 8.2.2. *HeCtor* achieves the best fairness when no changes occur in the DTVs channels, thus underlining the cooperation among the BSs as in Fig. 8.5. In contrast, when the DTVs change their transmission channels, there is a lack of fairness among BSs due to higher complexity in computing the channel assignment, which slows down its adaptability. In contrast, *NoRa* behaves in the same way no matter how the environment changes.

8.3.2 NoRa and HeCtor vs. MMGMS

Our first comparison is in terms of *system convergence cost*, i.e., the number of stages to obtain an NE point, between *Nora* and the MMGMS scheme [103] when the number of channels varies and for different numbers of WRANs. Figure 8.3 demonstrates that *NoRa* converges faster in all cases and, in particular, when the number of channels is high. Unfortunately due to the different interference model, protocol interference model for MMGMS and physical interference model for *NoRa*, it is not possible to compare these two game models in terms of the number of channels used or in terms of local and/or global interference. Hence, in the following we compare our games using the metrics defined in Section 8.2.2.

Figure 8.4 shows how the self-coexistence algorithms react when the number of CPEs and, hence the traffic flows, vary in the network. We can see how the throughput per CPE increases when the number of CPEs varies from 140 to 280, which means that there is still space for more flows. In contrast, when the number of CPEs is 420, the throughput per CPE decreases because the network cannot handle so many flows due to increased interference. As shown in Fig. 8.6, *HeCtor* adapts more slowly than *NoRa* when the DTVs change transmission channels.

In order to understand how BSs behave differently based on the self-coexistence

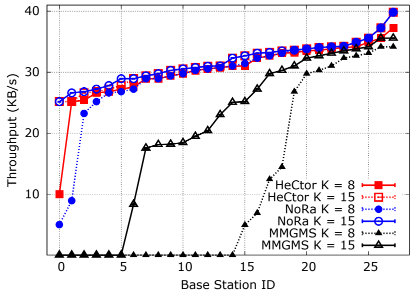

Figure 8.5: Throughput for each BS in the network, considering 140 web flows, 140 VoIP flows, and that no DTV changes its transmission channel. The number of channels is 8 or 15.

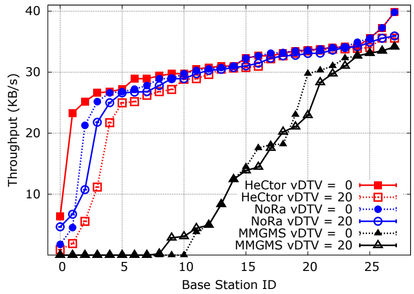

Figure 8.6: Throughput for each BS in the network, considering 140 web flows, 140 VoIP flows, and 9 channels. We compare the case where no DTV changes its channel, with the case where all the DTVs change their channels every 15 s.

algorithm, we show in Fig. 8.5 the throughput achieved by each BS when 140 web flows and 140 VoIP flows are served. As shown in Fig. 8.3, the system convergence cost is greater when the MMGMS scheme is used. Thus, there are more stages where the transmissions fail and the throughput is smaller. Moreover, the proposed cooperative scheme turns out to be a better choice than the non-cooperative scheme when the number of channels is 8. This is because the cooperation among BSs produces less interference when the number of channels is comparable with the required resources. When the number of channels is 10, we observe that *NoRa* and *HeCtor* are equivalent in terms of throughput performance.

In a similar way, Fig. 8.6 compares the throughput for each BS when no DTV changes its transmission channel with the case where all the DTVs change their transmission channels every 15 s. Observe that *NoRa* and *HeCtor* attain a higher throughput in both cases as compared with MMGMS. In fact, as explained before, MMGMS has a lower convergence speed. Moreover, when DTVs change their channels, there are more BSs where all the transmissions fail. Comparing *NoRa* and *HeCtor*, we observe how *HeCtor* has a higher throughput when no changes occur. However, when DTVs change channels, the throughput is comparable to *NoRa*. This is due to the fact that *HeCtor* is computationally more intensive than *NoRa*, i.e., it requires more time to reallocate the channels. During this time the BSs cannot transmit, thus implying that the throughput decreases. On the other hand, *NoRa* has the same throughput whether or not the DTVs change. This means that *NoRa* can adapt very fast with changes in the surroundings.

8.4 Conclusions

The implementation of an ad-hoc event-driven simulator enables us to study the behavior of our algorithms more thoroughly and consequently to compare them with other existing algorithms in the literature.

The simulation results show that the cooperative and non-cooperative frameworks mainly differ in terms of how fast they can adapt to changes in coordination among devices, which results in higher computational complexity.

In general, cooperation among devices has advantages in terms of network throughput, however, when licensed users change their transmission parameters more frequently, the performance in *NoRa* and *HeCtor* are the same because the cooperative framework needs more time to adjust its parameters to the new spectrum usage. Hence, when PUs often change their transmission channels, the non-cooperative scheme *NoRa* seems to be a better choice.

It is forbidden to kill, therefore all murderers are punished unless they kill in large numbers and to the sound of trumpets.

Voltaire

9 Conclusions

In this Part, we analyzed the self-coexistence problem in cognitive access networks. We first presented general concepts of game theory, which is an extensively used approach in the literature to address the self-coexistence problem. Then, we described two novel game theoretic approaches along with several game theoretic solutions from the literature.

The two novel game theoretic frameworks, *NoRa* and *HeCtor*, follow a non-cooperative communication paradigm among selfish cognitive device, and a cooperative approach, respectively. We formulated these two games in a completely distributed way and evaluated the interaction among cognitive devices using the physical interference model. Additionally, we implemented an ad-hoc event-driven simulator in C++, which enables us to make a thorough comparison of the throughput of various methods.

We conclude that the cooperation among BSs achieves a higher throughput but at the cost of higher computational complexity, which leads to a loss of throughput in cases where rapid changes occur in the channels occupancy. In contrast, the non-cooperative game framework attains the same throughput independent of the variability in channel occupancy, hence the BSs adapt faster to such changes.

Future work will include an evaluation of the proposed game theoretic frameworks using channels with different characteristics, thus providing each BS with the capacity to decide on the best channel according to the application QoS requirements.

Part III

Coordination Problem in Cognitive Multi-Hop Networks

You will never find time for anything. If you want time, you must make it.

Charles Buxton

10 Cognitive Multi-Hop Networks

10.1 Introduction

Cognitive multi-hop networks (C-MHNs) are attracting an always growing community of researchers thanks to the possibility of creating and extending the abilities of pervasive communication applications to cognitive environments [3, 31]. However, several issues have to be address in C-MHNs, such as topology changes, resource discovery and resource allocation [26].

10.1.1 Outline of the Chapter

The Chapter is organized as follows. Section 10.2 summarizes issues related to C-MHNs, which are then classified in cognitive ah-hoc networks (C-AHNs) and cognitive wireless mesh networks (C-WMNs) in Sections 10.2.1 and Sections 10.2.2, respectively.

Given the general overview, we focus on the major problem which afflicts C-MHNs, the coordination problem among CDs. Section 10.3 describes the coordination problem as a common control channel problem describing issues and approaches proposed in the literature to solve it. Section 10.4 describes some techniques to solve the control channel problem and Section 10.5 focus on the clustering approach. Concluding, Section 10.6 describes solutions which use the clustering approach to solve the common control channel problem.

10.2 Issues in Cognitive Multi-hop Networks

Topology changes afflict cognitive and non-cognitive multi-hop wireless networks. In non-cognitive networks (NCNs), end-users mobility is seen as the main reason for topological changes because routes formed over multiple hops may periodically experience disconnections. Solutions addressing the adaptation to these changes have been proposed in the literature [89]. In addiction to the end-users mobility, in C-MHNs topology changes due to PUs presence must be considered. For this reason, C-MHNs require a different approach compared to non-cognitive networks in order to handle end-user mobility and variability in time and space of available resources.

In order to simplify the topology change problem in C-MHNs, we consider only changes due to PU presence and we postpone mobility issues as future work. Considering only PUs presence we have that, from CD to CD, the available resources vary in time and space and are so local instead of identical in all the CDs. This leads in the problem of establishing end-to-end paths because CDs do not know the views that the other CDs have about the environment. For this reason resource availability informations have to be *disseminated* at least between neighboring devices, and possibly beyond that, to ensure point-to-point and consequently end-to-end communications.

From the previous considerations emerge that in C-MHNs routing and resource allocation problems become more complex than in non-cognitive multi-hop networks because the set of relay CDs to forward data has to be chosen dynamically based on resources and bandwidths available over all necessary links. Each link in the end-to-end path could be on a different channel according to the resource availability seen by each device and it could be subject to frequent channel switching due to PUs presence. Thus, a collaboration between routing and spectrum allocation is required to establishing end-to-end paths. Maintaining end-to-end paths involves not only the traffic load, but also how many different channels are used in the path, the number of PUs induced channels switching events, and consideration of periodic spectrum sensing. Moreover, a fair resource sharing among all flows in the network should be considered. Concluding, end-to-end paths have to be established and maintained, hence tight couplings with resource availability views and allocation decisions is necessary. Consequently, resulting communication solutions are most likely to be based on cross-layer interactions.

10.2.1 Cognitive Ad-Hoc Networks

Ad-hoc networks are characterized by the absence of a centralized support, and hence must rely on local coordination to disseminate network informations, such as topology knowledges. In non-cognitive ad-hoc networks (NC-AHNs), informations are disseminated using periodic beacon messages on a common channel for all non-cognitive devices. In cognitive ad-hoc networks (C-AHNs), instead, the existence of a common channel always available to all CDs is not guaranteed hence a periodic beacon should be sent on all the available channels. However, sending beacons over all the possible channels is not efficient and often not feasible. Thus, C-AHNs are highly probable to have incomplete topology information, which leads in an increase in collisions among CDs as well as interference to PUs.

10.2.2 Cognitive Wireless Mesh Networks

Wireless mesh networks are formed by a backbone and several end-users as described in Section 2.1.

When cognitive capabilities are added to WMNs, they can be added to both, backbone devices and end-users to obtain a C-WMN. In order to handle the coordination problem in C-WMNs, we focus on the backbone tier alone and hence on the so called *cognitive mesh devices* (C-MDs) which are SUs able to sense the environment, learn form it and avoid harmful interference to PUs.

C-MDs are wireless connected in a multi-hop fashion and need to agree on a communication channel in order to transmit packets and guarantee Internet access through mesh gateways to end-users. The problem that has to be handled is how two neighboring C-MDs, which sense different environments, can agree on a common communication channels available for both. In fact, each C-MD is aware of its own available resources but has not knowledges on the resources available to its neighbors. From this consideration we derive that the major challenge in C-WMNs is the necessity of a coordination mechanism among C-MDs which allows the exchange of informations needed to guarantee connectivity and hence compute end-to-end paths, resource allocation, etc.

Issues in Cognitive Wireless Mesh Networks

Several are the functionalities needed in order to manage a C-WMN: spectrum sensing, spectrum decision, and spectrum sharing.

During the *spectrum sensing*, a C-MD monitors and captures information about channels detecting PUs presences. Hence, the spectrum sensing requires assimilation of information from several C-MDs in order to improve accuracy and fairness in sharing available resource. The information dissemination leads in the necessity of coordination mechanism among C-MDs.

Spectrum decision consists in selecting the best available channel among the chunks identified during the spectrum sensing. In C-WMNs, the spectrum decision involves jointly spectrum selection and route formation because it needs reliable route formation and packet delivery over multiple hops. Hence a single C-MD observation uncertainty can be minimized through a coordination mechanism.

Finally, *spectrum sharing* avoids collisions among C-MDs. It includes channel and power allocations to avoid interference with PUs and other C-MDs.

All the previous functionalities require exchange of informations and hence a coordination mechanism among C-MDs. The coordination problem among C-MDs can be addressed assuming the existence of a centralized control entity or implementing a control messages exchange mechanism. The existence of a centralized control entity in C-WMNs is not guaranteed because C-MDs form a multi-hop backbone tier and hence there are not guarantees on the existence of a connection from each C-MD to the centralized control entity. The control messages exchange, instead, is a suitable solution for C-WMNs but it is not without challenges particularly since C-MDs experience spectrum variability over time and location and hence a fixed control channel suitable for every C-MD could not exist. In Chapter 11, we address the coordination problem among C-MDs as an exchange control messages problem proposing a solution based on local common control channels.

10.2.3 Related Works

Several studies have been published in the recent years describing and analyzing issues in C-MHNs. A comprehensive overview on challenges and solutions was proposed in [4], where the authors focused on the most promising approaches and commented the future research roadmaps.

Several are the issues addressed in the literature concerning C-MHNs. We recall routing problems [59], distributed spectrum sharing problems [11], end-to-end bandwidth allocation problems [112], and spectrum sensing problems [31]. Among the approaches used to solve these issues we identify MAC schemes (evaluated and classified in [100]), multi-channel MAC protocols [76], and cooperative techniques

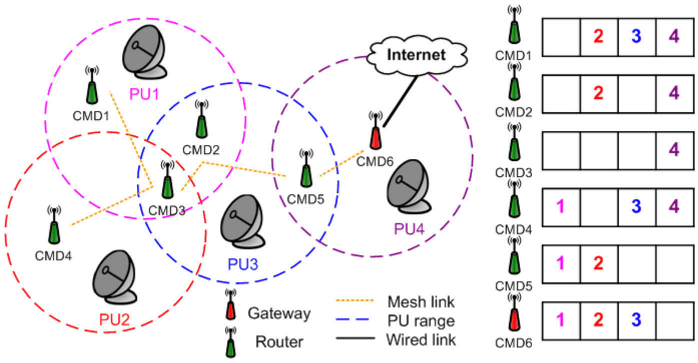

Figure 10.1: A Cognitive Wireless Mesh Network Architecture. On the right side a representation of the available channels for each cognitive mesh device is shown.

for C-MDs [109]. Moreover, the authors in [103] addressed the problem of mesh end-users which gradually join each C-MD and work as relay for other mesh end-users. Beside these, the most treated problem was the coordination problem addressed as a common control channel problem.

10.3 Common Control Channel Problem

The control channel problem addresses the willing of network devices to coordinate among themselves in order to satisfy network requirements and perform resource allocation algorithms, spectrum access coordination, network topology, transmission power, bandwidth requirements, etc. In general, the needed information for each network device can vary depending on the algorithm that nodes want to run and in particular in a cognitive network, C-MDs could exchange informations regarding PUs presence and location (spectrum sensing statistics).

Several approaches have been proposed in the literature to address the control channel problem in NCNs. However, these are not applicable to CNs due to time and space variability of available resources. For example in NCNs, network devices may use a predefined control channel to negotiate the data channel assignment and to reserve the medium for future transmissions. In CNs the existence of a predefined control channel which is available for transmission for every CD is not guaranteed due to space and time variability in available resources, as shown in [119] where the authors proved that a common available channel for all the CDs does not exist even for a small number of PUs. However, they asserted that every CD shares a significant number of common available channels with all of its neighbors, i.e., nearby CDs have very similar views of spectrum availability.

Figure 10.1 shows an example of C-WMN where four PUs exist. We notice how neighboring C-MDs share common channels but not all of them share the same channel. Hence, distinct "local" control channels may be allocated in different neighborhoods with the objective of share coordination information locally (i.e., within the one-hop neighborhood), or globally via multiple hops.

In order to address the variability of available control channels, and hence the absence of static and global control channels, network partitioning approach may be used. The most common partitioning approach is the *clustering approach* in which a CN is divided into clusters. In each cluster, CDs share a common control channel and exchange their individual sensing results which are combined to get the final result, such as sensing, routing, resource allocation, etc.

10.4 Control Channel Problem Approaches

The control channel problem has been addressed with several approaches in the literature. We distinguish *static* [61] and *dynamic* approaches in Section 10.4.1, then in Section 10.4.2 we propose *in-band* control channels and *out-of-band* control channel approaches.

10.4.1 Statical and Dynamical Approaches

Statical control channel approaches consider a static and universal control channel. Dynamic control channel approaches, instead, consider a time variable control channel.

In statical control channel approaches we identify three issues. First, static spectrum usage, i.e. a static channel is assigned before deployment, increasing complexity and cost. Second, scarcity of control resources, i.e. a fixed channel limits scalability in terms of device density, traffic and spectrum ranges. Finally, security issues, i.e. a simple jamming attack of the fixed control channel would disrupt the entire network.

The static spectrum usage can be solved using a dynamic control channel approach. The scarcity of control resources, instead, afflicts both, statical and dynamical, approaches. In fact, a control channel could easily become the bottleneck of the network especially in multi-hop scenarios, where the amount of control information is very high. Moreover, a static control channel approach is in contrast with the opportunistic nature of CNs and hence is not longer treated in this paper.

Different from the dedicated control channel approaches, either static or dynamic, is the technique proposed in [11], where the authors implemented a virtual control channel based on multiple-rendezvous. They assume that cognitive devices visit channels in a pseudo-random fashion and exchange control information whenever they happen to meet on any channel. To simplify the multiple-rendezvous they assumed that time is divided into allocation periods of a predetermined duration, called slots. The slot division is a strong requirement because assume synchronization and hence a previous agreement among CDs.

Studies which declare that control channels are not needed also exist. In [13], the authors asserted that systems based on multi-carrier code division multiple access (MC-CDMA) do not need a control channel. However, the MC-CDMA scheme proposed in [13] does not consider the potential mismatches between the available spectrum at the transmitter side and at the receiver side, which are typical in CNs because of geographic separation between transmitters and receivers. Hence in C-MHNs, the control message exchange is mandatory to ensure successful communication between transmitters and receivers and to exchange network control messages which are unavoidable in any network.

10.4.2 In-band and Out-of-band Approaches

An *out-of-band* control channel can be identified with a small aside channels, i.e. a channel not used for data transmissions. In CNs, an out-of-band channel can be chosen among the unlicensed or licensed channels.

We identify two types of unlicensed bands usable as out-of-band control channels in cognitive networks: industrial scientific and medical (ISM) bands [23] and ultra wide bands (UWB). If an ISM band is used, the disadvantage is the interference produced by unlicensed devices. Hence, the use of unlicensed bands cannot guarantee the reliability of control transmissions, which are critical for cognitive operations. An UWB could be used to solve the interference problem by unlicensed devices.This solution, which was first proposed in [99], is appealing for several factors: UWB communications cause negligible interference to narrow band transmissions, all devices are able to discover each other over the UWB channel using a common spreading code, and UWB radio interfaces feature very low complexity and power consumption. However, UWBs are designed to work in indoor environments to cope with rich multi-path components into channels and hence several are the disadvantage in using them as control channels in CNs. We identify two major disadvantages: UWB systems have short transmission ranges, while CNs are usually proposed for medium to long-range communications and they fail in addressing the initial link establishment problem [46] in either centralized or decentralized networks.

If a licensed band is used, the disadvantages are: a reduced effective bandwidth for data transmission and the difficulty in finding a fixed and reliable out-of-band control channel, since all the available channels may be leased from PUs.

An *in-band* control channel approach uses the same channels used to transmit data in order to transmit control informations leading in a reduction in available bandwidth for data transmissions. An in-band control channel introduces low overhead and does not need to take care of interferences produced by unlicensed devices. An in-band technique was proposed in [62] where the authors exploited the possibility to set up a network without an a-priori selected control channel. Some CDs send beacon messages sequentially or randomly on the available channels, while other devices scan the spectrum. Therefore, CDs can establish a direct contact only when one of them receives the beacon transmitted by the others. The time required by two specific devices to meet could be long, introducing what we call *timing problem*. The authors asserted that in the worst case two devices will contact each other in $K^2 \times T_s$ seconds, where T_s is the time that a device spend on each channel waiting for a beacon and K is the number of channels.

10.5 Clustering Approaches

In this section we introduce different approaches used in the literature to address the clustering problem. Let us first introduce concepts and motivation behind clusters formation protocols.

A cluster is a group of linked devices, called *members*, which usually share common interests and characteristics. Clusters are usually implemented to create cooperation among members and improve performance. If a distributed environment is considered, network devices execute distributed algorithms in order to coordinate

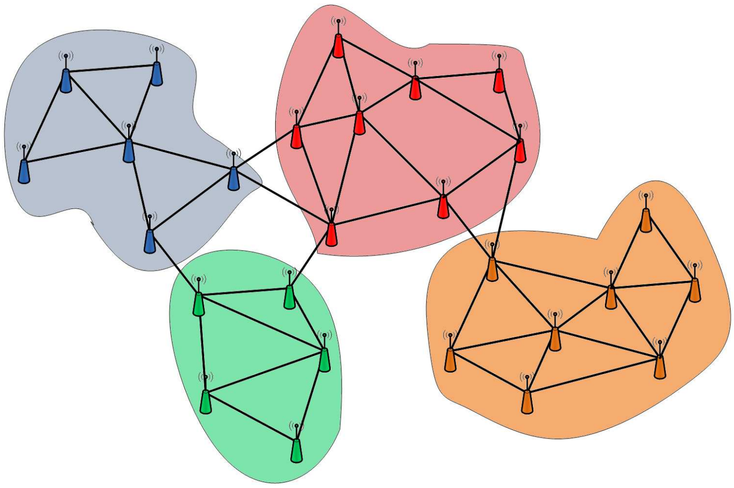

Figure 10.2: Clustered Network.

and form clusters. A coordination mechanism among members of a cluster can be obtained assuming a control channel common to all the cluster members, i.e. two network devices can belong to the same cluster if they use the same coordination channel. Hence, a cluster is characterized by a common control channel among all the members. Figure 10.2 shows an example of clustered network where each color correspond to a different cluster.

The problem of distributed clustering in wireless networks with fixed number of channels has been extensively studied in the literature [15, 16, 29, 63, 67, 111] and the objectives addressed by these works are various, such as self-organization, power saving, channel access, routing, etc.

The main distinction in clustering approaches is made based on: (i) how members join a cluster, (ii) how a cluster head is elected. A cluster head is elected among the cluster's members and takes care of all the operations performed by that cluster, hence the *cluster head election problem* is an inportant issue in clustering approaches. In Section 10.5.1 we analyze how members join a cluster distinguishing leader-first and cluster-first approaches, while in Section 10.5.2, we analyze the cluster head election problem proposing highest-degree, lowest-ID and node-weight heuristic approaches. We conclude our considerations in Section 10.5.3 presenting some security issues related to cluster formation protocols.

10.5.1 Leader-first and Cluster-first Approaches

Clustering problems are classified into leader-first approaches and cluster-first approaches based on when a cluster head is elected.

In *leader-first* approaches, a cluster head is first elected based on metrics such as degree of connectivity, mobility, residual energy, and ID [15, 16, 29].

In *cluster-first* approaches, instead, each cluster is formed before a cluster head is elected [63, 67, 111]. For example in [111], the authors proposed a fully connected cluster formation approach where clustering decisions are made purely based on connectivity. Such approaches require all the devices in one cluster agree on the same membership before electing their cluster head.

10.5.2 Highest-degree, Lowest-ID and Node-weight Heuristics

Choosing cluster heads optimally is an $\mathcal{NP}$-hard problem, however several heuristics have been proposed in the literature. We identify three different families: highest-degree heuristics, lowest-ID heuristics, and node-weight heuristics [17].

Highest-degree heuristics choose as cluster head the member with the highest degree, where the degree of a node is defined as the number of its one-hop neighbors. Experiments demonstrated that the highest-degree approach has a low rate of cluster heads change but the throughput is low [29]. Moreover, a cluster head may not be able to handle a large number of members due to resource limitations even if these members are its one-hop neighbors.

Lowest-ID heuristics assign a unique ID to each device and choose the member with the minimum ID as cluster head for each cluster. The lowest-ID heuristic performance is better compared with the highest-degree heuristic in terms of throughput, but members IDs are arbitrarily assigned numbers, hence this approach does not consider qualifications of network devices. In both, highest-degree and lowest-ID heuristics, the clustering algorithm is based on the link-cluster architecture where a link exists between two network devices if they are in the communication range each other.

Node-weight heuristics assign a weight (a real number ≥ 0) to each network device based on its suitability of being a cluster head. A network device is chosen to be a cluster head if its weight is higher than any of its neighbors' weights. Otherwise, it joins a neighboring cluster. The number of updates required is smaller than in highest-degree and lowest-ID heuristics [29]. However, a network device has to wait for all the responses from its neighbors to make its own decision to be a cluster head. Moreover, in CNs computing cluster heads becomes very expensive since CD weights could vary when network architecture and surrounding environment vary.

10.5.3 Security Issues

Cluster formation protocols may be afflicted by security issues. In leader-first approaches malicious devices may lie about their metrics to make themselves elected as cluster heads. As a result, a malicious cluster head can control all its cluster members. Similarly, none of the cluster-first protocols can guarantee a consistent view of the surrounding environment when malicious devices send false information. To address these issues, Vasudevan et al. [116] proposed two secure leader election algorithms by using a trusted authority to certify each network device metric used in the leader election process.

This approach is not suitable for CNs due to the presence of malicious devices and PUs, which can appear in every time instant. In fact due to connectivity problems, it is not realistic assume that no messages are lost or delayed and all the CDs are reliable.

10.6 Control Channel and Clustering

Clustering approaches have been extensively used in the literature to address the control channel problem in CNs. For example, clustering approaches were proposed in [119], where the authors assumed that CDs self-organize in coordination groups in a distributed manner, and in [66] where the clustering formation was used in order to solve the problem of dynamically assign control channels.

In [119], devices in each coordination group exchange control messages using a local available common channel which changes dynamically in response to PUs activities. The channel available to the largest set of one-hop neighbors is selected as control channel in each coordination group. This approach minimizes the set of distinct channels used for control, but also reduces the common set of channels within each coordination group. This can lead to a *frequent re-clustering* due to variations in PUs activities and small number of common channels in each coordination group. Thus, this scheme may not respond timely enough to the variations in the available channels because of its long re-organizing time.

In [66], the authors implemented a distributed cluster agreement algorithm that provides a desirable balance between two competing factors: the set of common available channels within each cluster and the cluster size. They grouped neighboring devices with similar channel availability in the same cluster and they assumed a time-slotted system. However, a time-slotted system requires a priori informations regarding scheduling and synchronization among CDs. To address the synchronization problem, the authors in [66] assumed that all devices can synchronize using PUs signals to initially acquire a common time reference.

Try to learn something about everything and everything about something.

Thomas Hardy

11 Connor

11.1 Introduction

In order to addressed the coordination problem in C-WMNs, and hence the control channel problem, we propose a distributed *control channel formation protocol* (*Connor*) [43]. Using *Connor*, C-MDs self-organize into clusters based on similarity of available channels and they share a common control channel used for control purpose.

We take a cluster-first approach where a weight is assigned to each CD based on the cluster topology. To the best of our knowledges no techniques for C-WMNs using fully distributed clustering approaches have been proposed in the literature so far.

Connor includes two phases: the *Cluster Formation Phase* (CFP), which is divided into a *discovery stage* and an *establishment stage*, and the *Keep Alive Phase* (KAP). During the discovery stage, neighboring C-MDs exchange information to get know to each other, meanwhile during the establishment stage, devices create clusters distributively. KAP keeps clusters and network status up to date. Figure 11.1 shows the phases and stages of *Connor*.

11.1.1 Outline of the Chapter

The Chapter is organized as follows. Section 11.2 describes the terminology used along the Chapter and Section 11.3 introduces *Connor*. CFP and KAP are detailed described in Section 11.4 and Section 11.5, respectively. We conclude the description of *Connor* by proposing our approach to treat the connectivity problem among clusters in Section 11.6 and the re-clustering problem in Section 11.7.

11.2 Model and Terminology

In this section, we introduce some terminology regarding clustering approaches in conjunction with C-WMNs.

Devices belonging to a cluster are called *members* and form a mini multi-hop network under the same control channel. Of the members, we identify two types of C-MDs with particular features, *cluster head device* (CHD) and *cluster border*

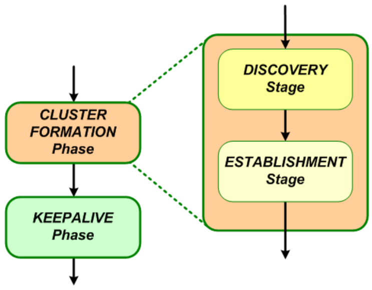

Figure 11.1: Control Channel Formation Protocol.

devices (CBDs), meanwhile all the other members are called *cluster ordinary devices* (CODs).

A CHD guarantees intra-cluster communication by aggregating data from CODs and CBDs, hence reducing the amount of data sent into the network. All the information that a cluster member receives is sent to the CHD which then processes the information and takes related actions. In addition, a CHD can arrange a time-slotted schedule for wireless channel access. This means that message collisions can be reduced by allowing only one C-MD to access the channel at any time.

CBDs guarantee inter-cluster communication and hence they create bridges between clusters. In order to do that, a CBD shares a control channel with more than one cluster and it is able to switch between control channels when required. Channel switching has to be scheduled carefully to avoid inefficiencies and lack of connectivity.

Given a C-WMN the set of channels is identified by $\mathcal{K}$ where K is its cardinality, while the set of C-MDs is $\mathcal{N}$ with cardinality N. The set of available channels can vary from C-MD to C-MD at every time instance and is defined as the set of chunks of the frequency spectrum where PUs do not transmit. We indicate with $\mathcal{K}_i(t)$ the set of available channels for the C-MD $i \in \mathcal{N}$ at the time instance t and with $K_i(t)$ its cardinality.

Depending on PU activities, the set of clusters can also vary at every time instance t. We indicate with $\mathcal{B}(t)$ the set of clusters at time t, with $B(t)$ its cardinality and with $\mathcal{B}_p$ a specific cluster where $p = \{1, \ldots, B(t)\}$. To avoid any heavy mathematical notation, we do not indicate the time variable if not necessary. Note that each C-MD always belongs to a cluster, hence a cluster could have a single C-MD as a member.

11.3 Connor: Control Channel Formation Protocol

Connor is a clustering approach which addresses the control channel problem in C-WMNs where C-MDs self-organize themselves into clusters, based on the similarity of available channels and on topological constraints. We take a cluster-first approach where a weight is assigned to each C-MD based on the cluster topology. To the best of our knowledge no techniques for C-WMNs using fully distributed clustering approaches without synchronization have previously been proposed.

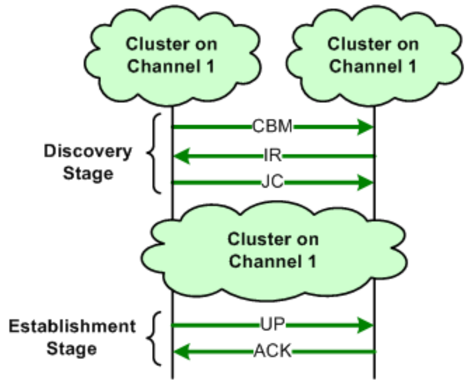

Figure 11.2: Cluster Formation Phase.

11.4 Cluster Formation Phase

We consider a distributed system architecture where no centralized control entity or synchronization mechanism among C-MDs are required. In such a scenario each C-MD, $i \in \mathcal{N}$, needs to perform three main actions. (i) Identify the set of available channels ($\mathcal{K}_i$) using a sensing protocol. (ii) Discover its one-hop C-MDs, for short *neighbors*, broadcasting beacons to establish a first contact in order to set up a communication link. Note that the neighbor discover is at the basis of any network communication. (iii) Agree on a common available channel in order to exchange control information and hence form a cluster.

Figure 11.2 shows the messages exchanged during the CFP, in which we identify two stages: a discovery stage, described in Section 11.4.1, and an establishment stage, described in Section 11.4.2. Control messages exchanged during the CFP are described in Section 11.4.3.

11.4.1 Discovery Stage

Connor needs information about available resources and neighbors. Resources are known performing spectrum sensing, while neighbors are discovered broadcasting beacons. At time $t = 0$ we have N C-MDs and $B(0) = N$ clusters. First, C-MDs perform spectrum sensing and initialize $\mathcal{K}_i$, $\forall i \in \mathcal{N}$. Then, each C-MD $i \in \mathcal{N}$ needs to inform and to be informed about its neighbors. The neighbor discovery is performed by each C-MD using repeated control messages, called *cluster beacon messages* (CBMs), and channel listening. Synchronization among C-MDs is not required in *Connor*, hence each C-MD starts transmission and listening autonomously and asynchronously.

Given a cluster $\mathcal{B}_s$ and a C-MD $i \in \mathcal{B}_s$, i transmits a CBM on the first channel of the available channel list, say $k \in \mathcal{K}_{\mathcal{B}_s}$. If a C-MD $j \in \mathcal{B}_r$ with $j \neq i$ and $\mathcal{B}_s \neq \mathcal{B}_r$, is listening on the same channel $k \in \mathcal{K}_{\mathcal{B}_r}$, then j receives the CBM which is forwarded to its CHD, say H_r. The CHD decides to form a cluster based on the *cluster formation constraints* described in Section 11.4.4. Otherwise, i repeats transmission and listening on k for a certain number of times, say $MaxTxRx$, waiting for a response from its neighbors. In Section 12.2 we show how $MaxTxRx$

affects the neighbor discovery time. If no replay is received then i repeats the same procedure on the following channel in $\mathcal{K}_{\mathcal{B}_s}$ and so on until a response from a neighbor is received or a neighbor is met during the listening. If no neighbors are met, a C-MD re-starts the procedure from the first channel in $\mathcal{K}_{\mathcal{B}_s}$. The previously described procedure is infinitely repeated by all C-MDs.

Note that since C-MDs need to broadcast CBMs on all the available channels, the neighbor discovery time is in general longer than that of a single channel network. To speed up the process, the discovery phase is not conducted on all the available channels but it stops at the first channel where a neighbor is met.

In order to better understand *Connor*, let us describe an example with two clusters, the sender cluster $\mathcal{B}_s$, the receiver cluster $\mathcal{B}_r$ and their sets of available channels $\mathcal{K}_{\mathcal{B}_s} = \{1, 2, 3\}$ and $\mathcal{K}_{\mathcal{B}_r} = \{1, 3\}$, respectively.

(i) C-MD $i \in \mathcal{B}_s$ broadcasts CBMs on channel 1.

(ii) C-MD $j \in \mathcal{B}_r$ listens on channel 1, hence it receives a CBM which is forwarded to the cluster head H_r.

(iii) H_r checks the cluster formation constraints: number of hops and common available set of channels. The *number of hops* is checked in the worst case computing $max\Delta_{\mathcal{B}_s} + max\Delta_{\mathcal{B}_r} + 1 < max\Delta$, where $max\Delta$ is the maximum number of hops admitted into a cluster and $max\Delta_{\mathcal{B}_p}$ with $p \in \{1, \dots, B\}$ is the maximum number of hops in the cluster $\mathcal{B}_p$. The *common available set of channels* is the intersection of $\mathcal{K}_{\mathcal{B}_s}$ and $\mathcal{K}_{\mathcal{B}_r}$, which must be > 2.

(iv) If the cluster formation constraints are satisfied, H_r sends a control message to j, called *inquiry request* (IR), which is then forwarded to i.

(v) The IR sent by $j \in \mathcal{B}_r$ and received by $i \in \mathcal{B}_s$ is forwarded to the cluster head H_s which does not need to check the cluster formation constraints as they are automatically satisfied. In fact, we assume that the cluster formation constraints are the same for all the C-MDs belonging to a C-WMN.

(vi) H_s needs to send a *join confirmation* (JC) message to $\mathcal{B}_r$ in order to send the information necessary to form the merged cluster, say $\mathcal{B}_u = \mathcal{B}_s + \mathcal{B}_r$, and not included in the CBM.

(vii) Finally, j receives the JC and forwards it to H_r.

(viii) When both clusters have the same information each other, H_s and H_r compute H_u, which is the cluster head of the merged cluster. The procedure to compute H_u is called *cluster head election* and it is described in Section 11.4.5. After the cluster head election, the discovery stage ends. In the establishment stage, if possible, $\mathcal{B}_s$ and $\mathcal{B}_r$ merge into a single cluster with a common control channel.

If the cluster formation constraints are *not* satisfied the following situations can happen.

- If $B_s = B_r = 1$ then the CBM is discarded because it means that $\mathcal{K}_{\mathcal{B}_s} \cup \mathcal{K}_{\mathcal{B}_r} < 2$ and hence i and j cannot belong to the same cluster. To speed up the process, j will immediately change listening channel.

- If $(B_s > 1 \ \& \ B_r = 1)$ or $(B_s = 1 \ \& \ B_r > 1)$ then the *cluster border device procedure* described in Section 11.6 takes place.
- If $B_s > 1 \ \& \ B_r > 1$ then to avoid conflict on the control channel $\mathcal{B}_s$ or $\mathcal{B}_r$ have to switch control channel and hence re-cluster as described in Section 11.7.

To avoid inconsistent conditions we set two timeouts T_{IR} and T_{JC} for IR and JC, respectively. T_{IR} is reset when the corresponding JC arrives, while T_{JC} is reset when the establishment stage starts. If T_{IR} or T_{JC} elapses, $\mathcal{B}_s$ and $\mathcal{B}_r$ assume that the merge is not possible and carry on with their individual operations. In order to recover lost packets and speed up the CFP, C-MDs involved in a discovery stage transmit and listen starting from the same channel where the discovery stage failed.

11.4.2 Establishment Stage

The establishment stage is proposed in order to form the merged cluster. During this stage, H_s and H_r send their information on $\mathcal{B}_s$ and $\mathcal{B}_r$ to H_u using an *update message* (UP) which works as an *acknowledgement message* (ACK) for the discovery stage. H_u confirms the received information by sending an ACK to all the C-MDs in $\mathcal{B}_u$. The ACK is also used to inform all the C-MDs in $\mathcal{B}_u$ of the new merged cluster and its corresponding CHD.

If a C-MD in $\mathcal{B}_u$ does not receive the ACK in a period of time equal to T_{UP}, then the C-MD assumes that something went wrong and performs the re-clustering procedure as described in Section 11.7.

At the end of the establishment stage, the new merged cluster is formed and can carry on with the normal cluster operations.

11.4.3 Exchanged Messages

Three messages are involved in the discovery stage as shown in Fig. 11.2: *cluster beacon message* (CBM), *inquiry request* (IR) and *join confirmation* (JC). Given two C-MDs, $i \in \mathcal{B}_s$ and $j \in \mathcal{B}_r$, in the following we show the exchanged messages in details.

CBMs sent by i contains:

- Cluster ID of $\mathcal{B}_s$.
- Distance in number of hops from the furthest C-MD, say p, and i into $\mathcal{B}_s$. We indicate the distance in number of hops between C-MD i and p as $\Delta_{i,p}$.
- Common available channel list, i.e. the intersection of the available channels for all the C-MDs into $\mathcal{B}_s$, say $\mathcal{K}_{\mathcal{B}_s}$.

These are the minimum information needed to check the cluster formation constraints or to start a cluster border device procedure (see Section 11.6).

The IR sent by j, which receives the CBM sent by i, contains:

- Cluster ID of $\mathcal{B}_r$.
- List of cluster members, $\mathcal{B}_r$.
- Distance in number of hops from each C-MD to any other C-MD in $\mathcal{B}_r$.

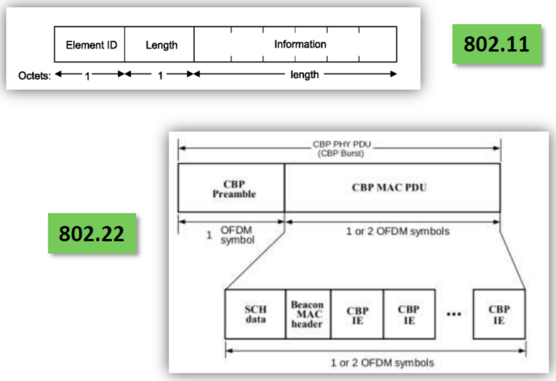

Figure 11.3: Information Elements in IEEE 802.11 and IEEE 802.22 standards.

- Common available channel list, $\mathcal{K}_{\mathcal{B}_r}$.

The JC sent by i contains:

- Cluster ID of $\mathcal{B}_s$.
- List of cluster members, $\mathcal{B}_s$.
- Distance in number of hops from each C-MD to any other C-MD in $\mathcal{B}_s$.

JC contains the missing information contained in the IR but not in the CBM. The information transmitted by IR and JC are used to compute the CHD of the merged cluster.

Note that these messages can be sent as *Information Elements* (IEs) in the beacon payload in IEEE 802.11 and IEEE 802.22 standards. Figure 11.3 shows the IEs in the two previously mentioned standards. Hence, the frequency on which CBMs are sent is the same given by standard.

11.4.4 Cluster Formation Constraints

In *Connor* we identify three cluster formation constraints: delay constraint, common available channels constraint and border device constraint.

The *delay constraint* regulates that the number of hops between the CHD and the furthest C-MD in a cluster must be limited. That is, we want to limit the information dissemination time in each cluster to avoid situation in which C-MDs far away from the CHD receive information that are out of date. In fact, the control overhead increases dramatically with the number of relaying devices and the number of hops [20]. Moreover, we want to take the amount of control information disseminated low. In fact, control information is broadcast by the CHD to its cluster members and it is clear that if the number of hops from the CHD to the furthest C-MD increases, also the control traffic broadcast increases and hence more resources are used in order to disseminate control information. A similar reason conducts us to propose a protocol where the CHD is in a central position respect to its cluster members. As consequence, the delay constraint limits also the number of cluster members.

The *common available channels constraint* establishes that the number of common available channels among cluster members must be greater than two. In fact, one common available channel is used as *control channel* and another one as *backup control channel*. This constraint helps during the *re-clustering procedure* as shown in Section 11.7.

The *border device constraint* determines that two neighboring clusters cannot have the same control channel. In fact, a *cluster border device* (CBD) is a bridge between two clusters with different control channels.

11.4.5 Cluster Head Election

Many cluster-based protocols require the election of a *cluster head device* (CHD) for facilitating protocol operations. During the establishment stage, sender ($\mathcal{B}_s$) and receiver ($\mathcal{B}_r$) clusters have the necessary informations to perform a *cluster head election* to determine the CHD of the merged cluster.

The cluster head election is conducted by assigning a weight to each C-MD in a cluster in order to have the CHD in a central position with respect to the number of hops between C-MDs. The weight of a C-MD $i \in \mathcal{B}_u$, indicated by w_i is given by Eq. (11.1).

$$w_i = \frac{\mu_u}{\rho_i} \tag{11.1}$$

Where the distance in number of hops between each couple of C-MDs $i, j \in \mathcal{B}_u$ with $i \neq j$ is indicated by $\Delta_{i,j}$, $\mu_u = \max_{i,j \in \mathcal{B}_u, i \neq j} \Delta_{i,j}$ is the number of hops from the two furthest C-MDs in $\mathcal{B}_u$, and $\rho_i = \max_{j \in \mathcal{B}_u, i \neq j} \Delta_{i,j}$ is the number of hops between i and the furthest C-MD in its cluster ($\mathcal{B}_u$). The cluster head H_u is the one where w_i is maximized. If more than one C-MD has the same maximum weight then if one is the CHD of $\mathcal{B}_s$ or $\mathcal{B}_r$, it is also the CHD of $\mathcal{B}_u$, otherwise the one with the highest device ID is chosen.

11.5 Keep Alive Phase

In order to guarantee coherence among clusters in a C-WMN, *keep alive messages* (KAs) are used. KAs are used to keep cluster status information fresh and to recover from crashes and lost packets.

Each C-MD in each cluster periodically sends a KA to its CHD to let it know that it is still alive and communicate changes in the cluster status information. In the same way, the CHD of each cluster sends a KA back to all the cluster members.

Each CHD maintains a table containing an entry for each cluster member with its information. An *aging policy* is applied to this table. Given a cluster $\mathcal{B}_s$ with H_s as CHD, an entry of a node $i \in \mathcal{B}_s$ is valid for an amount of time and is refreshed when a KA arrives from i. If i does not receive KAs from H_s, for a certain amount of time, it leaves the cluster and re-clusters as described in Section 11.7. In the same way, if H_s does not receive KAs from i, then i is disassociated from $\mathcal{B}_s$ when its entry expires.

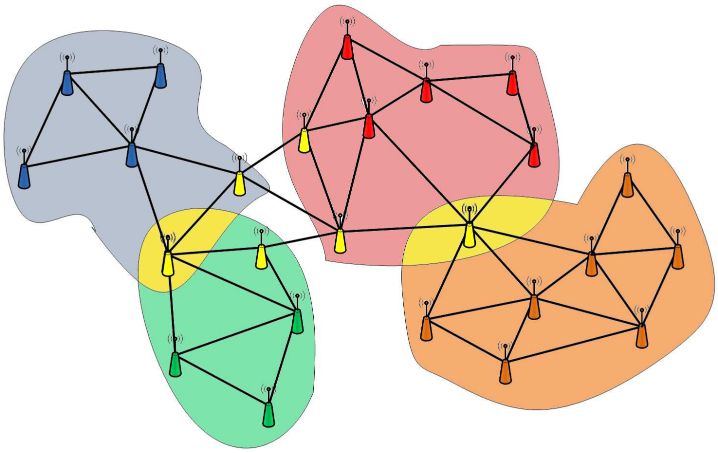

Figure 11.4: Potential cluster border devices (in yellow).

A C-WMN is a multi-hop network, for this reason KA messages can be merged in order to reduce the amount of control traffic in the network.

11.6 Cluster Border Device Procedure

In order to have inter-cluster communication and hence detect *cluster border devices* (CBDs), C-MDs also listen on channels that are different from their control channel. Figure 11.4 shows, in yellow, potential CBDs.

A CBD is a C-MD belonging to a cluster and that connects its own cluster to another cluster. Hence, it can communicate on both control channels. When a C-MD $j \in \mathcal{B}_r$ receives a message from a neighbor belonging to a different cluster and on a different control channel, say $i \in \mathcal{B}_s$, j sends a message called a *border device inquiry* (BD_{inq}) to its cluster head, H_r. A BD_{inq} contains: device ID and cluster ID of i in order to inform H_r of the possible bridging. H_r decides if i and j will act as CBDs for clusters $\mathcal{B}_s$ and $\mathcal{B}_r$, respectively. If bridging is possible, j sends a message to i called a *border device request* (BD_{req}). The C-MD i responds with a *border device confirmation* (BD_{conf}) message to j in order to establish the inter-cluster link between $\mathcal{B}_s$ and $\mathcal{B}_r$. Hence, H_r sets j as CBD to reach $\mathcal{B}_s$ and H_s sets i as CBD to reach $\mathcal{B}_r$. Devices i and j synchronize in order to meet on the common channel from time to time and perform the inter-cluster communication.

A CBD cannot be established when an inter-cluster communication already exists between two clusters or when a C-MD is already a CBD for a different cluster. If a C-MD is already a CBD for the same cluster, then it does not send a BD_{inq} to its CHD, while if a C-MD is a CBD for another cluster, then the border device procedure does not start in order to avoid frequent channel switching on the same device. Each CHD maintains a table, managed with an *aging policy*, which contains information on CBDs and on how to reach neighboring clusters.

The inter-cluster bridges are always bi-directional and only the C-MD that receives the first message, in the previous example j, asks to its cluster head whether or not it has become a CBD. This because j handles the complexity of the inter-cluster communication and switches between two channels to guarantee the connectivity, meanwhile i stays always on its control channel.

11.7 Re-clustering and Backup Control Channel

When a PU emerges on the current control channel, C-MDs need to quickly vacate the channel and set up a different control channel and/or a different cluster structure. We indicate this mechanism with the term *re-clustering*.

Setting up a new cluster structure is expensive because C-MDs, which sense a PU presence, have to leave the current control channel and perform the entire CFP. However two techniques can be used to speed up a re-clustering procedure: (i) CHDs could broadcast a new control channel for the entire cluster, or (ii) predetermined backup control channels could be used.

Using the first technique, C-MDs which sense PUs send a message to their CHDs in order to notify the PU presence. The CHD computes the new control channel and broadcasts its decision on the current control channel to inform cluster members about the change. Using this procedure, the amount of messages sent on the channel occupied by the PU is high. In contrast we want to keep this interference to the minimum. For this reason a backup control channel technique has been adopted.

Using a backup control channel technique, a CHD chooses a backup control channel and informs its cluster members when the current control channel is still free of PUs. When C-MDs identify a PU on the current control channel, they inform their CHD and switch to the backup control channel. The CHD decides if the remaining members *switch* to the backup control channel or *stay* on the current control channel based on the size of the remaining cluster. Note that the decision to switch or stay is broadcast by the CHD on the current control channel only for those cluster members that do not sense PUs. This approach thus limits the amount of messages sent on the current control channel because the only information is sent to the CHD by the members that can no longer use the current control channel.

In both cases there is assumed to be more than one common channel available in each cluster and hence a *control channel migration* without re-clustering is possible. That is, we assume that the re-clustering procedure is not repeated every time the set of available channels changes at a C-MD. Instead, each cluster gradually adapts its set of common available channels to spectrum variations and the re-clustering procedure is avoided as long as at least one common channel is still available among members of a cluster.

A control channel migration is also engaged when a C-MD receives a CBM from a neighboring cluster on the same control channel. In this case one of the two clusters has to switch channel. The cluster that has to switch control channel is the one with the smallest number of members to reduce the complexity of the procedure. The CHD of the cluster which migrates control channel sends a KA to all its members in order to notify the required switch on the backup control channel.

11.7.1 Example

Let us illustrate an example. A cluster $\mathcal{B}_s = \{a, b, c, d, e\}$ with $H_s = \{c\}$ has as control channel k_s and as backup control channel k_{s*}. C-MDs a and b sense a PU on channel k_s, hence a and b send a message to their H_s and change control channel to k_{s*}, i.e., we tolerate a minimum interference to PUs. H_s decides if the remaining C-MDs ($\{c, d, e\}$) switch control channel to k_{s*} or stay on k_s. This choice is based on the number of remaining C-MDs. If $\frac{B_s}{2} + 1$ C-MDs can use k_s then the cluster $\mathcal{B}_s$ is still in place with control channel k_s, otherwise they all switch on k_{s*}. In our example we have a remaining cluster $\mathcal{B}_s = \{c, d, e\}$ on channel k_s. Instead nodes a and b are on channel k_{s*} and send a KA, but they do not receive a KA back from c and for this reason they conclude that the cluster is broken and they start the CFP again as single C-MDs. Considering that a and b start from k_{s*} it is likely that they form a cluster $\mathcal{B}_r = \{a, b\}$ on k_{s*}. This procedure guarantees rapid re-clustering.

If the CHD decides that the entire cluster should switch its control channel to the backup control channels, then the CHD sends to $\mathcal{B}_s = \{c, d, e\}$ a KA on k_s containing this information. Then the following KAs are sent on k_{s^*}. Hence a and b, which are already on k_{s^*}, receive the new KA and the cluster after the control channel migration is $\mathcal{B}_s = \{a, b, c, d, e\}$.

A theory is something nobody believes, except the person who made it. An experiment is something everybody believes, except the person who made it.

Albert Einstein

12 Performance Evaluation

12.1 Introduction

In order to evaluate *Connor*, we implemented the control mechanism, described in Chapter 11, in our *ad-hoc event-driven simulator* for CNs. The experiments were conducted by randomly placing PUs (DTVs and microphones) and C-MDs in a square area of 100×100 km^2. We considered 10 microphones with a transmission range of 10 m and DTVs with transmission power in $[1,\ 5000]$ kW. If not specified otherwise, the number of DTVs is 10. The number of C-MDs depends on the specific simulation and range from 15 to 90, transmission ranges, say $TxRange$, are in $\{5, 10, 15, 20, 25, 30\}$, the number of channels is in $\{6, 8, 10, 15, 20\}$ and $max\Delta = 5$. The number of common channels in each cluster for *Connor* is set to 2 if not specified otherwise. We average our results over 50 different scenarios.

12.1.1 Outline of the Chapter

The Chapter is organized as follows. In Section 12.2, we outline the characteristics of *Connor* proposing comparisons varying $TxRange$ and $MaxTxRx$, and then in Section 12.3 we make a comparison with *SyncCFP* [119], an existing algorithm in the literature.

12.2 Connor Analysis

In order to analyze *Connor*, we show how the $TxRange$ of each C-MD and $MaxTxRx$ influence our clustering protocol with different numbers of C-MDs and channels.

Figure 12.1 shows the number of clusters for a different $TxRange$. The number of clusters decreases when the $TxRange$ increases because the number of one-hop neighbors for each C-MDs is larger and hence they can form fewer clusters with more C-MDs each. In Fig. 12.1 we can also notice that when the number of channels increases, *Connor* forms fewer clusters because it is easier find a free channel in common among neighboring C-MDs.

Figure 12.2 shows the number of clusters for $MaxTxRx$ equal to 1 and 10. We see that the $MaxTxRx$ plays a minor role when the number of C-MDs is small (less than 45). In contrast when the number of C-MDs grows, the number of clusters decreases when $MaxTxRx$ increases. Hence, a bigger $MaxTxRx$ improves the

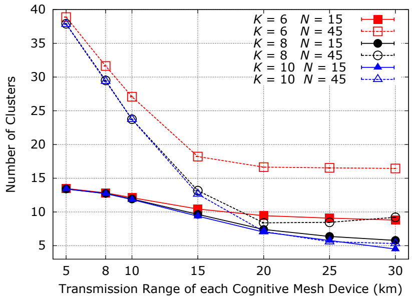

Figure 12.1: Transmission range of each C-MD vs. number of clusters.

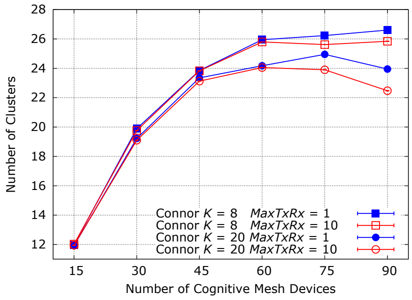

Figure 12.2: Number of C-MDs vs. number of clusters with $TxRange = 10$ and $K = 8$.

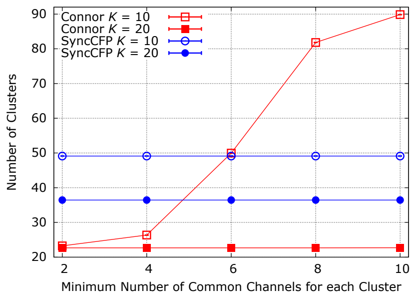

Figure 12.3: Minimum number of common channels for each cluster vs. number of clusters with $N = 90$ and $TxRange = 10$.

meeting probability and C-MDs can create bigger clusters with common control and backup channels.

12.3 Connor vs. SynCFP

The main difference between *Connor* and *SyncCFP* is the synchronization mechanism, which is required by *SyncCFP* but avoided by *Connor*. Synchronization among CMDs is very difficult to achieve in distributed systems such as C-WMNs. In *SyncCFP* synchronization means that neighboring C-MDs exchange messages, containing channel availability, on a pre-determined channel. The channel chosen for control is the one shared by the largest number of neighboring C-MDs.

Figure 12.3 shows how the constraint on the minimum number of common channels in a cluster influences the number of resulting clusters. *SyncCFP* does not have this constraint, hence its result does not depend on the minimum number of common channels. *Connor*, on the other hand, is influenced by the constraint on the number of common channels if the number of channel is low. The scenario presented has K equal to 10 or 20, 90 C-MDs and $TxRange = 10$ km. Note that when $K = 10$ if the minimum number of required common channels increases, then the number of clusters increases as well. When $K = 20$ instead, *Connor* creates a smaller number of clusters compared to *SyncCFP*.

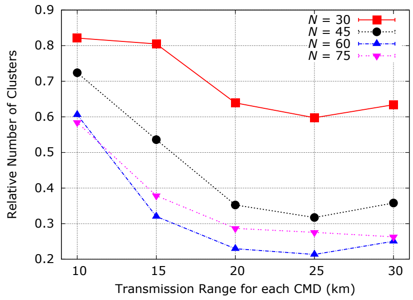

Figure 12.4: Transmission range vs. relative number of clusters with $K = 10$.

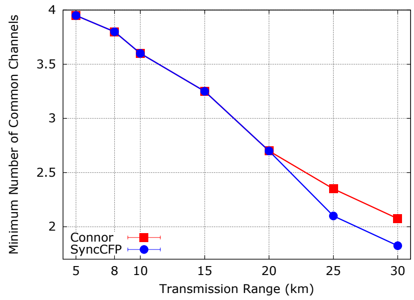

Figure 12.5: Transmission range vs. minimum number of common channels with $K = 10$ and $N = 30$.

In the following, we propose two comparisons between *Connor* and *SyncCFP* in the terms of number of clusters that they form. The first is a *static* analysis where PUs do not change their channel occupancy and a *dynamic* analysis where PUs vary their transmission channels.

We introduce a metric called *relative number of clusters* which is the number of clusters formed by *Connor* divided by the number of clusters formed by *SyncCFP*. This means that if the relative number of clusters is < 1, then *Connor* behaves better than *SyncCFP*.

Figure 12.4 shows the relative number of clusters when the $TxRange$ of each C-MD grows from 10 km to 30 km. We consider 10 channels, 10 DTVs, and the number of C-MDs is 30, 45, 60 and 75. *Connor* always behaves better in the proposed configurations, in particular when the number of C-MDs grows. However when $TxRange = 30$ km, *Connor* partially loses its improvements because it looks for two common channels, while *SyncCFP* only asks for a single common channel as confirmed by Fig. 12.5.

Figure 12.5 shows the minimum number of common channels on the overall clusters when the $TxRange$ varies from 5 km to 30 km and there are 30 C-MDs and 10 channels. Both algorithms have the same minimum number of common channels if this number is greater than two.

We now propose and explain a particular scenario in which *SyncCFP* behaves better than *Connor*. We set $K = 10$, the $TxRange$ to 10 km or 30 km, and the number of DTVs is set to 5. We also change DTV channels after 500 s and 1000 s on a simulation of 1500 s.

Figure 12.6 shows that *Connor* creates more clusters than *SyncCFP*. However, we show in Fig. 12.7 how these clusters are formed over time. To simplify the figure we take $N = 45$, but equivalent results are obtained for different numbers of CMDs. Note that the re-clustering procedure adopted in *Connor* does not require the clusters to break but they can immediately recover. Using *SyncCFP*, on contrary, C-MDs need more time to reorganize. Hence, the number of clusters reached in the stable state for *SyncCFP* is smaller, but C-MDs spend a long time forming clusters.

In conclusion, *Connor* achieves a better performance in most of the proposed scenarios without requiring synchronization among CMDs. It allows more rapid re-

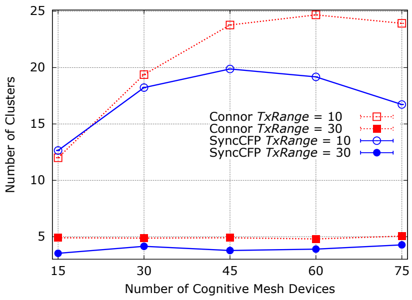

Figure 12.6: Number of C-MDs vs. number of clusters with $K = 10$ and 5 DTVs changing channels each 500 s.

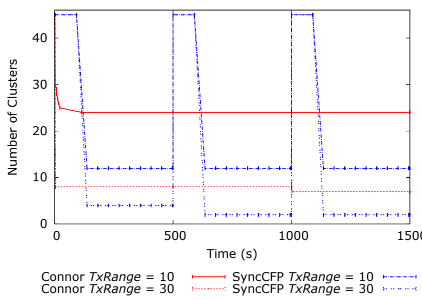

Figure 12.7: Time vs. number of clusters with $K = 10$, $N = 45$ and 5 DTVs.

clustering, which in turn leads to a more stable configuration.

We don't devote enough scientific research to finding a cure for jerks.
Bill Watterson

13 Conclusions

We have proposed a clustering algorithm to address the control channel problem in cognitive wireless mesh networks. Our algorithm, called *Connor*, does not require synchronization among cognitive mesh devices and allows rapid re-clustering when changes occur in channel occupancy by primary users. We have shown that *Connor* efficiently forms a limited number of clusters with common control and backup channels. Compared with the existing clustering algorithms in the literature, which requires synchronization, *Connor* performs better in most cases without imposing synchronization.

Part IV

Thesis Conclusions

It's what you learn after you know it all that counts.

John Wooden

14 Conclusions and Future Works

We analyzed the spectrum shortage problem under different scenarios proposing solutions for MRMC-WMNs, C-ANs and C-WMNs.

In the first part, we addressed the spectrum shortage problem in MRMC-WMNs analyzing the join routing and channel assignment problem and proposing several approaches presented in the literature. The techniques proposed are distinguished in optimization approaches and empirical approaches. The former have to deal with too a high execution time, while the latter produce solutions with poor performance results. In order to overcome these limitations we proposed a mixed approach, called *G-PaMeLA*, which is a divide-and-conquer approach to splitting in local sub-problems the joint channel assignment and routing algorithm for MRMC-WMNs. The performance of *G-PaMeLA* has been compared with different JCAR solutions presented in the literature using a packet-level simulator configured with several combinations of allowable channels and NICs per router. The comparison shown that the execution time of *G-PaMeLA* is relatively low, which makes it feasible for real MRMC-WMNs with non-specialized hardware, even for large networks with tens of mesh routers. Moreover, the results demonstrated that our scheme significantly improves network performance, in terms of the packet loss of all traffic flows and throughput fairness. However, channel assignment techniques for MRMC-WMNs used in order to address the spectrum shortage problem have been overwhelmed by the use of cognitive networks, which were the focus of the rest of this thesis.

In the second part, we analyzed the self-coexistence problem in C-ANs. A common mathematical tool used in the literature in order to address this problem is game theory which was described along with several game theoretic solutions from the literature. We proposed two game theoretic frameworks, called *NoRa* and *HeCtor*. The former follows a non-cooperative communication paradigm among selfish cognitive devices, while the latter is a cooperative approach. We formulated these two games in a completely distributed way and evaluated the interaction among cognitive devices using the physical interference model. Additionally, we implemented an ad-hoc event-driven simulator in C++, which enables us to make a comparison of the performance of various methods. We concluded that the cooperation among cognitive devices achieves higher throughputs but at the cost of higher computational complexity, which leads to a smaller throughput in cases where rapid changes occur in channels occupancy. In contrast, the non-cooperative game theoretic framework attains the same throughput independent of the variability in channels occupancy, hence cognitive devices adapt faster to such changes. Future work will include an evaluation of the proposed game theoretic frameworks using channels with different characteristics, thus providing each BS with the capacity to decide on the best channel according to the application QoS requirements.

In the third part, we proposed a clustering algorithm to address the control channel problem in C-WMNs. Our algorithm, called *Connor*, does not require synchronization among cognitive mesh devices and allows a fast re-clustering when changes occur in channels occupancy by licensed users. We showed that *Connor* efficiently forms a limited number of clusters with common control and backup channels. Moreover, we extended our ad-hoc event-driven simulator in C++ for cognitive networks adding the control level and we compared *Connor* with a synchronized clustering algorithm (*SyncCFP*) proposed in the literature. We concluded that *Connor* performs better than *SyncCFP* in most of the cases in term of number of channels used for control purposes and time to reach and stay on stable configurations. Future work will include a deeply study of C-WMNs which are still under standardization and on which a lot of work has to be done before real implementations.

Acknowledgments

The authors would like to thank Dr. Stefano Avallone and Dr. Mainak Chatterjee for sharing their source code for comparison purposes and for helpful discussions.

The authors would like to thank the anonymous reviewers for their comments that help improving the manuscripts on which this thesis is based on.

Bibliography

[1] *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, Jun. 2007.

[2] *IEEE P802.11s/D3.0. Draft standard - IEEE 802.11. Amendment 10: Mesh Networking*, IEEE, Mar. 2009.

[3] I. Akyildiz, M. Vuran, and S. Mohanty, *A survey on spectrum management in cognitive radio networks*, IEEE Communications Magazine, 46 (2008), pp. 40–48.

[4] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, *Spectrum Management in Cognitive Radio Ad Hoc Networks*, IEEE Network - Special issue title on networking over multi-hop cognitive networks, 23 (2009), pp. 6–12.

[5] I. F. Akyildiz and X. Wang, *A survey on Wireless Mesh Networks*, vol. 43, IEEE Communication Magazine, Sept. 2005.

[6] R. Al-Zubi, M. Z. Siam, and M. Krunz, *Coexistence Problem in IEEE 802.22 Wireless Regional Area Networks*, in IEEE GLOBECOM, Honolulu, Hawaii, USA, 2009, pp. 5801–5806.

[7] M. Alicherry, R. Bhatia, and L. Li, *Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks*, in ACM MobiCom, Cologne, Germany, Aug. 28–Sept. 2 2005, pp. 58–72.

[8] E. Amaldi, A. Capone, F. Malucelli, and C. Mannino, *Optimization Problems and Models for Planning Cellular Networks*, Springer Science, 2006.

[9] K. R. Apt and A. Witzel, *A Generic Approach To Coalition Formation*, International Game Theory Review, 11 (2009), pp. 347–367.

[10] T. Arnold and U. Schwalbe, *Dynamic Coalition Formation and the Core*, Journal of Economic Behavior and Organization, 49 (2002), pp. 363–380.

[11] A. Asterjadhi, N. Baldo, and M. Zorzi, *A Distributed Network Coded Control Channel for Multihop Cognitive Radio Networks*, IEEE Network: The Magazine of Global Internetworking, 23 (2009), pp. 26–32.

[12] S. Avallone, I. F. Akyildiz, and G. Ventre, *A Channel and Rate Assignment Algorithm and a Layer-2.5 Forwarding Paradigm for Multi-Radio Wireless Mesh Networks*, IEEE/ACM Transactions on Networking, 17 (2009), pp. 267–280.

[13] E. Azarnasab, R. Kempter, N. Patwari, and B. Farhang-Boroujeny, 1–3 Aug., pp. 472–481.

[14] T. Başar and G. Olsder, *Dynamic Non-Cooperative Game Theory*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[15] D. J. Baker, *The Design and Simulation of a Mobile Radio Network with Distributed Control*, IEEE Journal on Selected Areas in Communications, 2 (1984), pp. 226–237.

[16] S. Basagni, *Distributed Clustering for Ad Hoc Networks*, Fremantle, Australia, 23–25 Jun. 1999, pp. 310–315.

[17] S. Basagni, I. Chlamtac, and A. Farago, *A Generalized Clustering Algorithm for Peer-to-Peer Networks*, 1997.

[18] L. Berlemann, G. R. Hiertz, B. H. Walke, and S. Mangold, *Radio resource sharing games: enabling QoS support in unlicensed bands*, IEEE Network, 19 (2005), pp. 59–65.

[19] K. Bian and J.-M. Park, *A Coexistence-Aware Spectrum Sharing Protocol for 802.22 WRANs*, in IEEE ICCCN, Washington DC, USA, 2009, pp. 1–6.

[20] G. Bianchi, *Performance Analysis of the IEEE 802.11 Distributed Coordination Function*, IEEE Journal on Selected Areas in Communications, 18 (2000), pp. 535–547.

[21] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, *Architecture and evaluation of an unplanned 802.11b mesh network*, in ACM MobiCom, Aug. 28–Sept. 2 2005, pp. 31–42.

[22] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. Van Leeuwen, *Approximations for λ-coloring of graphs*, The Computer Journal, 47 (2004), pp. 193–204.

[23] T. X. Brown, *An Analysis of Unlicensed Device Operation in Licensed Broadcast Service Bands*, in IEEE DySPAN, Baltimore, MD, USA, 8–11 Nov. 2005, pp. 11–29.

[24] L. Cao and H. Zheng, *Distributed spectrum allocation via local bargaining*, in IEEE SECON, Santa Clara, CA, USA, 26–29 Sept. 2005, pp. 475–486.

[25] A. Caprara and M. Fishetti, *Branch-and-cut Algorithms*, M. Dell'Amico, F. Maffioli, and S. Martello, editors, Annotated Bibliographies in Combinatorial Optimization, 1997.

[26] M. Cesana, E. Ekici, and Y. Bar-Ness, *Networking over Multi-Hop Cognitive Networks*, IEEE Network, (2009), pp. 4–5.

[27] D. Chafekar, V. S. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, *Approximation Algorithms for Computing Capacity of Wireless Networks with SINR constraints*, in IEEE INFOCOM, Phoenix, AZ, USA,, 13–18 Apr. 2008, pp. 1166–1174.

[28] D. Challet and Y. C. Zhang, *Emergence of Cooperation and Organization in an Evolutionary Game*, Physica A: Statistical and Theoretical Physics, 246 (1997), pp. 407–418.

[29] M. Chatterjee, S. K. Das, and D. Turgut, *WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks*, Cluster Computing, 5 (2002), pp. 193–204.

[30] M. Chen, S. C. Liew, Z. Shao, and C. Kai, *Markov Approximation for Combinatorial Network Optimization*, in IEEE INFOCOM, San Diego, CA, USA, 14–19 Mar. 2010, pp. 1–9.

[31] K. R. Chowdhury and I. F. Akyildiz, *Cognitive Wireless Mesh Networks with Dynamic Spectrum Access*, IEEE Journal on Selected Areas in Communications, 26 (2008), pp. 168–181.

[32] C. Cicconetti, V. Gardellin, L. Lenzini, and E. Mingozzi, *PaMeLA: A Joint Channel Assignment and Routing Algorithm for Multi-Radio Multi-Channel Wireless Mesh Networks with Grid Topology*, in IEEE MASS, Macau SAR, P.R. of China, Oct. 12–15 2009, pp. 199–207.

[33] N. Clemens and C. Rose, *Intelligent Power Allocation Strategies in an Unlicensed Spectrum*, in IEEE DySPAN, Baltimore, MD, USA, 8–11 Nov. 2005, pp. 37–42.

[34] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, *Optimization Models for Fixed Channel Assignment in Wireless Mesh Networks with Multiple Radios*, in IEEE SECON, Santa Clara, CA, USA, 26–29 Sept. 2005, pp. 463–474.

[35] S. K. Das, V. Gardellin, and L. Lenzini, *Game Based Self-Coexistence Schemes in Cognitive Radio Networks*, Cognitive Radio MANETs, Springer, 2010.

[36] R. Draves, J. Padhye, and B. Zill, *Routing in multi-radio, multi-hop wireless mesh networks*, in ACM MobiCom, Philadelphia, PA, USA, 26 Sept.–1 Oct. 2004, pp. 114–128.

[37] *Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies FCC Report and Order*, Mar. 2005. FCC-05-57A1.

[38] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, 1991.

[39] V. Gardellin, *Cognitive Networks and Game Theory: A Winning Combination*, in Ph.D. Defence Corner, InfQ, Pisa, Italy, 7–9 Jul. 2010.

[40] V. Gardellin, S. K. Das, and L. Lenzini, *A Fully Distributed Game Theoretic Approach to Guarantee Self-Coexistence among WRANs*, in IEEE INFOCOM Workshop on Cognitive Wireless Communications and Networking, San Diego, CA, USA, 19 Mar. 2010.

[41] ———, *Cooperative vs. Non-Cooperative: Self-Coexistence among Selfish Cognitive Devices*, in submitted to IEEE International Conference, 2011.

[42] V. Gardellin, S. K. Das, L. Lenzini, C. Cicconetti, and E. Mingozzi, *G-Pamela: A Divide-and-Conquer Approach for Joint Channel Assignment and Routing in Multi-Radio Multi-Channel Wireless Mesh Networks*, Journal of Parallel and Distributed Computing, Elsevier, 71 (2011), pp. 381 – 396.

[43] V. Gardellin, L. Lenzini, and I. Savocchia, *A Clustering Approach to Address the Control Channel Problems in Cognitive Wireless Mesh Networks*, in submitted to IEEE International Conference, 2011.

[44] D. Goodman and N. Mandayam, *Network assisted power control for wireless data*, Mobile Networks and Applications, 6 (2001).

[45] D. Grandblaise, C. Kloeck, T. Renk, P. Bag, P. Levine, K. Moessner, J. Yang, M. Pan, and K. Zhang, *Microeconomics inspired mechanisms to manage dynamic spectrum allocation*, in IEEE DySPAN, Dublin, Ireland, 17–20 Apr. 2007, pp. 452–461.

[46] C. Han, J. Wang, and S. Li, *A Spectrum Exchange Mechanism in Cognitive Radio Contexts*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (2006), pp. 1–5.

[47] Z. Han and K. J. Ray Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*, Cambridge University Press, New York, NY, USA, 2008.

[48] S. Hart and A. Mas-Colell, *A simple adaptive procedure leading to correlated equilibrium*, Econometrica, 68 (2000), pp. 1127–1150.

[49] G. Hosseinabadi, M. H. Manshaei, and J.-P. Hubeaux, *Spectrum Sharing Games of Infrastructure-Based Cognitive Radio Networks*, tech. rep., 2008.

[50] http://tvmap.org/.

[51] http://www.ampl.com/, *version 11.21*.

[52] http://www.ilog.com/products/cplex, *version 11.21*.

[53] http://www.isi.edu/nsnam/ns/, *version 2.33*.

[54] http://www.ntia.doc.gov/osmhome/osmhome.html.

[55] http://www.ntia.doc.gov/osmhome/spectrumreform/.

[56] *IEEE P802.22/DRAFTv2.0: Draft Standard for Wireless Regional Area Networks Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands*, May 2009.

[57] H. Ishibuchi, T. Nakashima, H. Miyamoto, and C. Oh, *Fuzzy Q-learning for a multi-player non-cooperative repeated game*, in IEEE International Conference on Fuzzy Systems, vol. 3, 1997, pp. 1573–1579.

[58] A. Karnik, A. Iyer, and C. Rosenberg, *Throughput-Optimal Configuration of Fixed Wireless Networks*, IEEE/ACM Transactions of networking, 16 (2008), pp. 1161–1174.

[59] H. Khalifé, N. Malouch, and S. Fdida, *Multihop cognitive radio networks: to route or not to route*, IEEE Network: The Magazine of Global Internetworking, 23 (2009), pp. 20–25.

[60] K. H. Kim and K. G. Shin, *On accurate measurement of link quality in multi-hop Wireless Mesh Networks*, in ACM MobiCom, Los Angeles, CA, USA, 23–26 Sept. 2006, pp. 38–49.

[61] P. Kolodzy, S. Miller, K. Ryan, and J. Evans, *DIMSUMNet: New Directions in Wireless Networking Using Coordinated*, Taormina, Italy, 13–16 Jun. 2005, pp. 78–85.

[62] Y. R. Kondareddy, P. Agrawal, and K. Sivalingam, *Cognitive Radio Network setup without a Common Control Channel*, 16–19 Nov. 2008, pp. 1–6.

[63] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, *A Cluster-based Approach for Routing in Dynamic Networks*, ACM SIGCOMM Computer Communication Review, 27 (1997), pp. 49–65.

[64] P. Kyasanur and N. Vaidya, *Routing and interface assignment in multi-channel multi-interface wireless networks*, in IEEE WCNC, New Orleans, LA, USA, 13–17 Mar. 2005, pp. 2051–2056.

[65] A. M. Law and W. D. Kelton, *Simulation modeling and analysis*, Third edition, McGraw-Hill, 2000.

[66] L. Lazos, S. Liu, and M. Krunz, *Spectrum Opportunity-based Control Channel Assignment in Cognitive Radio Networks*, in IEEE SECON, 22–29 Jun.

[67] C. R. Lin and M. Gerla, *Adaptive Clustering for Mobile Wireless Networks*, IEEE Journal on Selected Areas in Communications, 15 (1997), pp. 1265–1275.

[68] X. Lin and S. Rasool, *A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-hoc Wireless Networks*, in IEEE INFOCOM, Anchorage, AK, USA, 6–12 May 2007, pp. 1118–1126.

[69] H. Liu and S. Li, *Reserch on the Mechanism of WRAN Self-Coexistence*, in IEEE International Symposium on Electromagnetic Compatibility, Honolulu, Hawaii, USA, 8–13 Jul. 2007, pp. 67–70.

[70] A. Lucena and J. E. Beasley, *Branch and cut algorithms*, Advanced in Linear and Integer Programming, J.E. Beasley, Ed. Oxford University Press, 1996.

[71] Q. H. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*, Wiley-Interscience, 2007.

[72] G. Mailath and L. Samuelson, *Repeated Games and Reputations: Long-Run Relationships*, Oxford University Press, New York, NY, USA.

[73] M. K. Marina and S. R. Das, *A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks*, in ICST BROADNETS, Boston, MA, USA, 3–7 Oct. 2005, pp. 381–390.

[74] S. Mathur, N. B. Mandayam, and L. Sankar, *Coalitions in Cooperative Wireless Networks*, IEEE Journal on Selected Areas in Communication, Special Issue on Game Theory in Communication Systems, 26 (2008), pp. 1104–1115.

[75] R. Menon, A. B. MacKenzie, R. M. Buehrer, and J. H. Reed, *Game Theory and Interference Avoidance in Decentralized Networks*, in Software Defined Radio Technical Conference and Product Exposition, 15–18 Nov, 2004, pp. 113–118.

[76] Y. Minghao, H. Lianfen, and C. Huihuang, *Multi-Channel MAC protocol for Cognitive Wireless Mesh Network*, in Anti-counterfeiting, Security and Identification, 20–23 Aug. 2008, pp. 203–206.

[77] J. Mitola, *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio Dissertation*, (2000).

[78] A. H. Mohsenian-Rad and V. W. S. Wong, *Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks*, IEEE Transactions on Wireless Communications, 6 (2007), pp. 4432–4440.

[79] M. Molina, P. Castelli, and G. Foddis, *Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering Through HTML-REDUCE*, IEEE Network, 14 (2000), pp. 46–55.

[80] D. Monderer and L. S. Shapley, *Potential Games*, Games and Economic Behavior, 14 (1996), pp. 124–143.

[81] A. Muthoo, *Bargaining theory with applications*, Cambridge University Press, New York, NY, USA, 1999.

[82] R. B. Myerson, *Game Theory: Analysis of Conflict*, Harvard University Press, 1991.

[83] J. O. Neel, J. H. Reed, and R. P. Gilles, *The role of game theory in the analysis of software radio networks*, in Symposium A Quarterly Journal In Modern Foreign Literatures, 2002.

[84] J. V. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

[85] N. Nie and C. Comaniciu, *Adaptive Channel Allocation Spectrum Etiquette for Cognitive Radio Networks*, Mobile Networks and Applications, 11 (2006), pp. 779–797.

[86] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[87] G. Owen, *Game theory, 3rd Ed.*, Academic Press. Inc, New York, 1995.

[88] P. Pawelczak, R. Prasad, L. Xia, and I. Niemegeers, *Cognitive radio emergency networksâĂŞrequirements and design*, in IEEE DySPAN, 2005, pp. 601–606.

[89] G. Pei and M. Gerla, *Mobility management in hierarchical multi-hop mobile wireless networks*, 11–13 Oct. 1999, pp. 324–329.

[90] J. D. Poston and W. D. Horne, *Discontiguous OFDM Considerations for Dynamic Spectrum Access in Idle TV Channels*, in IEEE DySPAN, Baltimore, MD, USA, 8–11 Nov. 2005, pp. 607–610.

[91] S. Ramanathan, *A Unified Framework and Algorithm for Channel Assignment in Wireless Networks*, ACM Wireless Networks, 5 (1999), pp. 81–94.

[92] A. Raniwala and T. Chiueh, *Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network*, in IEEE INFOCOM, Miami, FL, USA, 13–17 Mar. 2005, pp. 2223–2234.

[93] A. Raniwala, K. Gopalan, and T. Chiueh, *Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks*, ACM Mobile Computing and Communications Review, 8 (2004), pp. 50–65.

[94] D. Ray, *A Game-Theoretic Perspective on Coalition Formation*, Oxford University Press, New York, NY, USA, 2007.

[95] J. Robinson and E. W. Knightly, *A Performance Study of Deployment Factors in Wireless Mesh Networks*, in IEEE INFOCOM, Anchorage, AK, USA, 6–12 May 2007, pp. 2054–2062.

[96] R. W. Rosenthal, *A Class of Games Possessing Pure-Strategy Nash Equilibria*, International Journal of Game Theory, 2 (1973), pp. 65–67.

[97] A. Rubinstein, *Perfect equilibrium in a bargaining model*, Econometrica: Journal of the Econometric Society, 50, pp. 97–109.

[98] A. Sahai, S. M. Mishra, and R. Tandra, *Spectrum Sensing: Fundamental Limits*, (2010), pp. 1–21.

[99] M. E. Sahin and H. Arslan, *System Design for Cognitive Radio Conmunications*, 8–10 Jun. 2006, pp. 1–5.

[100] H. A. B. Salameh and M. Krunz, *Channel access protocols for multihop opportunistic networks: challenges and recent developments*, IEEE Network, 23 (2009), pp. 14–19.

[101] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, *Coalition Structure Generation with Worst Case Guarantees*, Artificial Intelligence, 111 (1999), pp. 209–238.

[102] S. Sengupta, R. Chandramouli, S. Brahma, and M. Chatterjee, *A Game Theoretic Framework for Distributed Self-Coexistence Among IEEE 802.22 Networks*, in IEEE GLOBECOM, New Orleans, LA, USA, 30 Nov.–4 Dec. 2008, pp. 1–6.

[103] S. Sengupta, M. Chatterjee, and R. Chandramouli, *A coordinated distributed scheme for cognitive radio based IEEE 802.22 wireless mesh networks*, in ICC 2008 workshop, 19–23 May 2008, pp. 461–465.

[104] S. Sengupta, M. Chatterjee, and S. Ganguly, *An Economic Framework for Spectrum Allocation and Service Pricing with Competitive Wireless Service Providers*, in IEEE DySPAN, Dublin, Ireland, 17–20 Apr. 2007, pp. 89–98.

[105] S. Sengupta, M. Chatterjee, and K. A. Kwiat, *A Game Theoretic Framework for Power Control in Wireless Sensor Networks*, IEEE Transactions on Computers, 59 (2010), pp. 231–242.

[106] L. S. Shapley, *Stochastic games*, Proceedings of the National Academy of Sciences, 39 (1953), pp. 1095–1100.

[107] H. Skalli, S. K. Das, L. Lenzini, and M. Conti, *Traffic and interference aware channel assignment for multi-radio wireless mesh networks*, Technical Report, 2006.

[108] H. Skalli, S. Ghosh, S. K. Das, L. Lenzini, and M. Conti, *Channel Assignment Strategies for Multi-radio Wireless Mesh Networks: Issues and Solutions*, IEEE Communications Magazine, 45 (2007), pp. 86–95.

[109] Y. Song, C. Zhang, and Y. Fang, *Joint Channel and Power Allocation in Wireless Mesh Networks: A Game Theoretical Perspective*, IEEE Journal on Selected Areas in Communications, 26 (2008), pp. 1149–1159.

[110] G. L. Stuber, *Principles of Mobile Communication*, Kluwer Academic Publishers, 2001.

[111] K. Sun, P. Peng, P. Ning, and C. Wang, *Secure Distributed Cluster Formation in Wireless Sensor Networks*, in ACM ACSAC, Miami Beach, FL, USA, 11–15 Dec. 2006, pp. 131–140.

[112] J. Tang, R. Hincapié, G. Xue, W. Zhang, and R. Bustamante, *Fair Bandwidth Allocation in Wireless Mesh Networks with Cognitive Radios*, 59 (2009), pp. 1487–1496.

[113] S. Tao and M. Krunz, *Coordinated Channel Access in Cognitive Radio Networks: A Multi-Level Spectrum Opportunity Perspective*, in IEEE INFOCOM, Rio de Janeiro, Brazil, 19–25 Apr. 2009, pp. 2976–2980.

[114] M. Van Der Schaar and F. Fu, *Spectrum Access Games and Strategic Learning in Cognitive Radio Networks for Delay-Critical Applications*, Proceedings of the IEEE, 97 (2009), pp. 720–740.

[115] M. Vasquez and J. K. Hao, *A Heuristic Approach for Antenna Positioning in Cellular Networks*, Journal of Heuristics, 7 (2001), pp. 443–472.

[116] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, and D, *Leader election algorithms for wireless ad hoc networks*, in DARPA Information Survivability Conference and Exposition, 22–24 Apr. 2003, pp. 261–272.

[117] B. Wang, Y. Wu, and K. R. Liu, *Game Theory for Cognitive Radio Networks: An Overview*, Computer Networks, 54 (2010), pp. 2537–2561.

[118] Y. Xu, J. C. S. Lui, and D. M. Chiu, *On oligopoly spectrum allocation game in cognitive radio networks with capacity constraints*, Computer Networks, 54 (2009), pp. 925–943.

[119] J. Zhao, H. Zheng, and G. H. Yang, *Distributed Coordination in Dynamic Spectrum Allocation Networks*, in IEEE DySPAN, Baltimore, MD, USA, 8–11 Nov. 2005, pp. 259–268.