UNIVERSITÀ DEGLI STUDI DI PALERMO

Department of Energy, Information Engineering and Mathematical Models

PHD Course in: **Matematica e Automatica per l'Innovazione Scientifica e Tecnologica**

Settore scientifico disciplinare **ING-INF/04**

Performance Optimization in Wireless Local Area Networks

PhD Candidate
Ing. Romina Badalamenti

PhD Course Coordinator
Prof. Francesco Alonge

Advisor
Prof. Laura Giarré

XXIV CICLO - Year 2015

DOTTORATO

To my sweet son Roberto,
that with his smiles has supported me
also in difficult times.

Abstract

Wireless Local Area Networks (WLAN) are becoming more and more important for providing wireless broadband access. Applications and networking scenarios evolve continuously and in an unpredictable way, attracting the attention of academic institutions, research centers and industry. For designing an efficient WLAN is necessary to carefully plan coverage and to optimize the network design parameters, such as AP locations, channel assignment, power allocation, MAC protocol, routing algorithm, etc... In this thesis we approach performance optimization in WLAN at different layer of the OSI model. Our first approach is at *Network layer*. Starting from a Hybrid System modeling the flow of traffic in the network, we propose a Hybrid Linear Varying Parameter algorithm for identifying the link quality that could be used as metric in routing algorithms. Go down to *Data Link*, it is well known that CSMA (Carrier Sense Multiple Access) protocols exhibit very poor performance in case of multi-hop transmissions, because of inter-link interference due to imperfect carrier sensing. We propose two novel algorithms, that are combining Time Division Multiple Access for grouping contending nodes in non-interfering sets with Carrier Sense Multiple Access for managing the channel access behind a set. In the first solution, a game theoretical study of intra slot contention is introduced, in the second solution we apply an optimization algorithm to find the optimal degree between contention and scheduling. Both the presented solutions improve the network performance with respect to CSMA and TDMA algorithms. Finally we analyze the network performance at *Physical* Layer. In case of WLAN, we can only use three orthogonal channels in an unlicensed spectrum, so the frequency assignments should be subject to frequent adjustments, according to the time-varying amount of interference which is not under the control of the provider. This problem make necessary the introduction of an automatic network planning solution, since a network administrator cannot continuously monitor and correct the interference conditions suffered in the network. We propose a novel protocol based on a distributed machine learning mechanism

in which the nodes choose, automatically and autonomously in each time slot, the optimal channel for transmitting through a weighted combination of protocols.

Acknowledgments

My first thanks go to my advisor, Professor Laura Giarré for all the guidance and support she has provided me over the course of this PhD. I am grateful for all her efforts and for helping me to constantly challenge and improve myself.

A special thank you goes to Professor Ilenia Tinnirello for her advice, kindness and support provided during these years.

I would also like to thank all members of the Department of Energy, Information engineering and Mathematical models, University of Palermo, and especially Professor Francesco Alonge and Professor Maria Stella Mongiovì for their technical coordination and assistance.

A special thank goes to late Mrs. Bonanno, that with her sweetness and kindness always tried to help others.

Contents

1 HLPV Identification in wireless ad-hoc networks **7**

1.1 Introduction 7

1.2 Model 9

1.2.1 System Model 9

1.2.2 Traffic Model 9

1.3 HLPV identification of the traffic model 11

1.4 Numerical Results 16

1.5 Future Works 18

2 Medium Access: Contention vs. Scheduling **24**

2.1 Introduction 24

2.2 Problem Formulation 27

2.2.1 Network Structure and Traffic Model 28

2.2.2 Resource Allocation 29

2.3 Utility-Based Resource Allocations in Multi-Hop Networks . . 30

2.3.1 Solution Approach 30

2.3.2 Graph coloring 30

2.3.3 Throughput assessment 32

2.3.4 Coloring Algorithms 34

2.3.5 Station Utility 35
2.3.6 Station Best Response 37
2.3.7 Numerical Results 38
2.3.8 Conclusions 41
2.4 Optimal Resource Allocation in Multi-Hop Networks: Contention vs. Scheduling 42
2.4.1 System Model 43
2.4.2 Transport Model 43
2.4.3 Channel Access Model 44
2.4.4 Network Partitioning 45
2.4.5 Frame Decision Optimization 50
2.4.6 Optimization problem 50
2.4.7 Numerical results 52
2.4.8 Conclusion 54

3 Frequency Planning **55**
3.1 Introduction 55
3.2 Dynamic Frequency Planning 58
3.2.1 Motivations 58
3.2.2 Supporting Channel Switching 60
3.3 Learning Scheme 61
3.4 Numerical Results 66
3.5 Experiments in a real testbed 69
3.5.1 Experiment Methodology 69
3.5.2 Results 71
3.6 Conclusions 72

List of Figures

1 The 7 Layers of the OSI Model. 2

1.1 Hybrid Model for the queue at a link. 20

1.2 Input/output data over an interval of $30s$ and sampling time T=0.1s for each flow $i = 1, 2, 3$. 20

1.3 Parameters $p1$, $p2$ and σ over an interval of $30s$ and sampling time $T = 0.1s$. 21

1.4 Parameters' identification in region 1. 21

1.5 Parameters' identification in region 3. 22

1.6 Parameters' identification in region 4. 22

1.7 Parameters across regions. 23

2.1 An example of medium access in a network with 5 nodes and a frame composed of 3 slots. 28

2.2 Queue Length at each node, as a function of the strategy of a given contending node. 38

2.3 Average throughput under the SC coloring scheme, for different incompatibility graphs and comparison with standard CSMA/CA. 39

2.4 Average throughput under the CFA coloring scheme, for different incompatibility graphs and comparison with standard CSMA/CA. 41

2.5 Network Topology. 46
2.6 Network divided in two random subnets. 46
2.7 Network divided in three random subnets. 47
2.8 Network divided in two chosen subnets. 47
2.9 Aggregated throughput varying node's subset. 54

3.1 An example of frequency planning based on the interference observed by the Access Points. 58
3.2 An example of dynamic (synchronous and asynchronous) channel switching performed at each beacon interval. 60
3.3 Operation of the algorithm. 63
3.4 Frequency plan in synchronous protocol without Carrier Sense. 64
3.5 time slots in asynchronous mode. 65
3.6 Number of packets successfully transmitted in 300 TS from each node comparing results obtained keeping in count or not traffic load in synchronous mode. 67
3.7 Frequency plan in synchronous protocol with Carrier Sense. . 68
3.8 Number of packets successfully transmitted in 300 TS from each node comparing results obtained keeping in count or not traffic load in asynchronous mode. 70
3.9 Network Topology. 71
3.10 Standard DCF transmission. 71
3.11 Arbitrary Frequency planning. 72
3.12 Frequency Planning. 73

List of Tables

2.1 Measurements and estimates of throughput. 42

2.2 Comparison between aggregated throughput dividing the network in subnets. 48

2.3 Comparison between the obtained (max, min, aggregated) optimal values of throughput with the one obtained with, respectively, CSMA end TDMA changing of the radius of visibility. . 53

List of Abbreviations

WLAN	Wireless Local Area Network
CSMA/CA	Carrier Sense Multiple Access Collision Avoidance
AP	Access Point
OSI	Open Systems Interconnection
TDMA	Time Division Multiple Access
LQSR	Link Quality Source Routing
MAC	Medium Access Control
PHY	Physical Layer
GPS	Global Positioning System
MWS	Maximum Weight Scheduling
LQF	Longest-Queue-First
GMS	Greedy Maximal Scheduling
GSM	Global System for Mobile Communications
FDMA	Frequency Division Multiple Access
LPV	Linear Varying Parameter
HLPV	Hybrid Linear Varying Parameter
LMS	Least Mean Square
EDCA	Enhanced distributed channel access
NE	Nash Equilibrium
CFA	Choosing the First Available color
SC	Select and Compare
DSSS	Direct Sequence Spread Spectrum

Introduction

In recent years we have witnessed the great spread of wireless technologies and the increasing of the networks in size and use; moreover the great variety of mobile devices and services requiring resources have made radio resource planning and optimization an attractive research topic. Thanks to their flexibility, low cost and ease of deployment, *Wireless Local Area Networks* (WLAN) are the most important technologies for wireless broadband access. Network layout and its configuration influence overall performance of a specific WLAN. For designing an efficient WLAN is necessary to carefully plan coverage and to optimize the network design parameter, such as AP locations, channel assignment, AP power allocation, MAC protocol, routing, etc... In the past, planning of the network was conceived as a static task. Now, in an heterogeneous radio environment, with mobile nodes, and conditions that are constantly changing, is necessary to apply dynamic network reconfigurability (for example [1]) featuring software defined radio [2, 3, 4, 5], arriving to nodes equipped with hardware card performing programmable MAC protocols [6]. In this thesis several algorithms for improving network's performance in mesh and ah-hoc networks at different layers of the OSI model (showed in fig.1) are proposed: starting from an algorithm for identifying the channel quality that could be used as metric for applying routing algorithms (layer 3), passing for two algorithms for improving MAC access (layer 2), arriving to the physical layer with frequency planning (layer 1).

At the *Routing Layer*, link quality is considered a performance metric in different protocols: in [7] Link Quality Source Routing (LQSR) is presented, that select a routing path according to link quality metrics; in this paper three different performance metrics compared with minimum hop-count met-

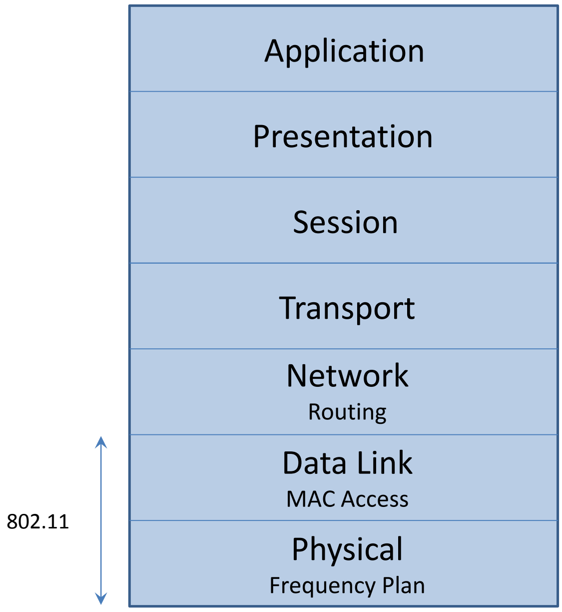

Figure 1: The 7 Layers of the OSI Model.

ric are presented, but in a mobile environment minimum hop-count metric has achieved the best performance. In [8] a multi-radio LQSR taking into account both link quality and minimum hop-count metrics is proposed. Another proposal is Multi-Path routing [10], that tries to perform load balancing and high fault tolerance selecting multiple path between source and destination. When a link is broken on a path can be chosen another path in the set of existing path. A drawback of this protocol is the complexity. In [9] a hierarchical routing is presented, that proposes to divide nodes in groups. Each cluster has one or more cluster heads. Some node can communicate with more clusters, acting as gateway, to maintain connectivity. In high density networks this algorithm gives good performance, but the complexity of maintaining hierarchy and cluster heads that could be bottleneck may degrade the performance of the routing protocol. Another class of routing protocol proposed is geographic routing, in which packets are forwarded using only the position of neighbor nodes and destination node obtained through GPS or similar location techniques. Topology change has less impact in this protocol

with respect to the others, but packet delivery is not granted even if a path exists between source and destination. In order to guarantee delivery, planar-graph-based geographic routing algorithm has also been proposed [11], but they require much higher communication overhead.
It has been shown that in literature an important metric in routing algorithm is the link quality. In this thesis we present an identification algorithm based on an *Hybrid Linear Varying Parameter* model of a network that could be used for evaluate instantaneous bandwidth, i.e. link quality, that could be used as metric in a routing algorithm.
WLAN are based on IEEE802.11 standard family, that is divided into two main layers: Medium Access Control layer (MAC) and Physical Layer (PHY). A MAC protocol decides which nodes can transmit data at each time instant in order to avoid that two active links interfere with each other. The original 802.11 CSMA/CA Medium Access Control (MAC) has shown significant shortcomings when facing the breakthrough rate improvements made available by the latest PHY enhancements (802.11n, 802.11ac), as well as when applied to scenarios and contexts such as ad hoc and mesh networks, vehicular environments, directional antennas, quality of service support, real time media streaming support, multi-channel operation, dynamic spectrum access, and many others. Designing high-performance MAC algorithms to achieve maximum possible throughput in WLAN is of primary importance. Since in many networks, especially when formed by a big number of nodes it is not possible to have a centralized control and resources at nodes could be limited, MAC protocol should be distributed and have low complexity and overhead. In literature there are a great number of proposed solutions: Maximum Weight Scheduling (MWS) algorithm is based on length of the queue in each nodes [12], it is throughput optimal but requires to solve NP-hard optimization problem in a centralized manner at each time slot. An alternative to this protocol with lower complexity is Longest-Queue-First (LQF) Scheduling. This algorithm schedule first nodes with longer queues disabling interfering nodes. It has demonstrated [13] that when topology network satisfy a *local-pooling* condition this algorithm obtains good performance in terms of throughput and delay, but in general topology it could not achieve optimal performance [14, 15] and when the size of the network grows also the

signaling and the time overhead can increase [16]. The most used protocols in wireless networks are *Carrier Sense Multiple Access* CSMA/CA algorithms. Applying these protocols in a distributed manner is very simple and they give good performance, but in presence of *hidden terminal* nodes, nodes that are in visibility with receiver node but not in reciprocal visibility, the performance could severely degrade. In [17] an analytical model is derived to calculate the throughput of a CSMA-type algorithm in multi-hop wireless networks describing the evolution of schedules through a Markov chain. Based on this result, a distributed algorithm was developed in [18] to adaptively choose the CSMA parameters to meet the traffic demand without explicitly knowing the arrival rates. On the basis of results in [17, 18], [19] proposes a discrete-time distributed queue-length based CSMA/CA protocol that combine CSMA with distributed GMS leading to collision free data transmission schedules. Another direction of designing CSMA algorithms for better delay performance is to use actual or virtual multi-channels, which is motivated by resolving the temporal link starvation of CSMA via de-correlating the temporal accessibility to the wireless medium. In [20] a CSMA algorithm with multiple frequency agility is presented, such that more than one frequency channel is available and a link can transmit on at most one of the channels. [21] proposes an algorithm called VMC-CSMA in which multiple virtual channels (defined by dividing the time line) are used to emulate a multi-channel system and address the starvation problem. The algorithm randomly selects a virtual channel, and the schedule corresponding to this chosen channel is used at each time slot. In [35] a fully-distributed CSMA based MAC protocol is presented. Employing virtual multi-channels it obtains good performance both in terms of throughput and delay in general wireless networks. Another class of MAC algorithms is *Time Division Multiple Access* (TDMA) in which transmission of each node is scheduled in a different time slot. This class of protocols avoids collisions, but may lead to channel wastes. In this thesis we propose two algorithms for MAC access based on a combination between TDMA and CSMA algorithms. We divide the nodes in subnets, scheduling transmissions of subnets in different time slots, and performing CSMA behind a subnet.

Another classical family of problems in WLAN planning and optimization

is the frequency assignment. The problem was brought into focus with the introduction of the second generation (2G) cellular networks; in particular, Global System for Mobile Communications (GSM) networks, that use a combination of Frequency Division Multiple Access (FDMA), TDMA, and random access. Frequency assignment problem could be approached in different ways [22, 23, 25, 27, 28, 32, 34]. The basic approach is finding a frequency assignment which is feasible to assignment constraints and interference constraints. Other approaches are the minimum interference problem and the minimum span frequency assignment problem. In the first problem, co-channel and adjacent channel interference are minimized. The second problem minimizes the difference between the highest and the lowest frequency used in solution. Frequency assignment is typically viewed as a graph coloring problem [26, 27]. [29, 33] propose a cross-layer design and optimization considering the information flows across the network layers to enable solutions that are globally optimal to the entire system and thus facilitating the optimal layer design. [24, 30] present another approach, considering layering as optimization decomposition, in which the overall optimization problem is decomposed in in subproblems corresponding to each layer. The interfaces among layers are quantified as functions of the optimization variables coordinating the subproblems. In this thesis we present a fully-distributed frequency planning based on a machine learning mechanism, in which each node chooses autonomously and automatically the transmission channel.
The remainder of the thesis is organized in three parts, respectively, addressing the above described issues related to WLAN networks:

- In the first chapter an HLPV algorithm for identifying link quality [66, 68] is presented. We consider a hybrid model for the flow of traffic in communication networks, and identifying the region in which the link is working we could evaluate its instantaneous bandwidth.

- In the second chapter two MAC algorithms are presented, in the first [67] we propose a simple approach based on preallocating temporal slots in which different sets of nodes are allowed to contend for the channel access. Since the approach does not completely prevent contentions for accessing the wireless channel, we also propose a game-theoretical analysis of contention strategies for multi-hop networks. In the second algorithm [69] we propose to convenient partition the network, applying an optimization algorithm for resource allocation.

- In the third chapter we propose a novel protocol [108] based on a distributed machine learning mechanism in which the nodes choose, automatically and autonomously in each time slot, the optimal channel for transmitting through a weighted combination of protocols.

Chapter 1

HLPV Identification in wireless ad-hoc networks

1.1 Introduction

Wireless ad-hoc networks consist of a number of untethered nodes able to communicate with each other by means of intermediate nodes, collaboratively forwarding ongoing traffic. Because of the nature of the wireless medium, data communications in ad-hoc networks are intrinsically broadcast, so that links exist between any pairs of nodes that are within the transmission range of each other. These features make ad-hoc networks easy to be deployed and suitable for a large number of applications, spanning from low-range sensor networks targeted to distributed monitoring, to high-range mesh networks targeted to build infrastructure-less transport networks.
In a WLAN, ensure that each node is aware of the performance of a link, allows the routing algorithm to make optimal decisions in case it is possible to reach a destination cross multiple path. Otherwise nodes should base their knowledge exclusively on static information. In this chapter we present an HLPV identification algorithm that could be used for evaluating link quality in wireless ad-hoc networks. We start our analysis from a model presented in [51] in which the flow of traffic in communication networks is modeled using a Hybrid System, which combines continuous-time dynamics with event-based

logic. Linear Parameter Varying, Hybrid Switching and Piece-Wise Affine models arise from different applications and have been studied in connection to various control schemes. Linear models allow for efficient design and software tools but often fail to provide a faithful description of the real systems. The use of LPV models as an approximation of nonlinear models became popular in the nineties [36]. In literature, a large amount of work on LPV control design and methods to obtain a set of stabilizing controllers proliferated since then. Actually, many industrial and advanced applications benefit the use of such techniques. Identification of LPV models has been approached since then in various way, by using State Space (SS) [37], [38], [39], Input/Output (I/O) [40], [41], representation The terminology Linear Parameter time-Varying (LPV) systems has been introduced in [36]. A discrete LPV system is represented in SS as

$$
\begin{aligned}
x_{k+1} &= A(p(k))x_k + B(p(k))u_k \\
y_k &= C(p(k))x_k + D(p(k))u_k
\end{aligned}
$$

The exogenous parameter $p(k)$ is assumed *a priori* unknown. However, it can be measured or estimated upon operation of the system. Rather than modeling the dynamical evolution of a particular variable, one can treat it as an exogenous independent parameter, [36], [42], and the survey [43], as well as reference therein. Since the beginning of the nineties, the literature on LPV Control and LPV Identification has started growing almost in parallel. The first attempt to solve the problem of LPV identification dates back to the paper of [38], where Linear Fractional Transformation (LFT) LPV system identification problem is solved having single time-varying block and state space measurements. The problem was shown to be equivalent to a linear regression, and certain conservative conditions for persistence of excitation were given.

Since the end of the last century, the literature on LPV have tremendously been spread and it is almost impossible to exhaustively list all the contributions here, but the main methodologies can be clustered according to various paradigm and criteria, in [44] an overview of all the approach of LPV identification and more recent solved problems has been presented. On the other hand, the importance of switching and hybrid dynamical systems [45] has been recognized, relying also upon the fact that robust stabilizing control can be designed, and new identification technique have been extensively considered in the literature, [46], [47]. Hybrid systems are dynamical systems consisting of components with continuous and discrete behavior and thus they have required a new hybrid systems and control theory that has been developed in the past years and is still an active research topic. A survey of the principal identification methods for hybrid systems can be found in [46], [48] and [49]. An interesting paper that deals with an identification problem close to the one discussed here is [50]. In the rest of the chapter we present an application of HLPV in ad-hoc networks.

1.2 Model

1.2.1 System Model

The network structure can be represented through an edge labeled graph $G = (V, E)$. Specifically, the nodeset V includes all the nodes i of the network and the edgeset E includes all the pairs of adjacent nodes (i, j) that are in radio visibility of each other. We call each edge $(i, j) \in E$ as link ℓ, that is labeled with its maximum transmission rate B_{max} reached only when we are in the best condition. $B^\ell \leq B_{max}$ is the effective transmission rate, that in real environment is not constant over time, and depends on nearby radio conditions, and among the other factors, is function of the distance between nodes i and j and of the possible presence of obstacles.

1.2.2 Traffic Model

To model the channel traffic load, we take into account and extend the model originally proposed by [51]. For a given topology, we consider some of the nodes as sources and some are considered the final receivers. Actually we consider that the routing of the flows has been set, so for each flow the path is pre-assigned. We consider a communication network, with n_l links, that we assume unidirectional, crossed by n flows. Each link has a finite buffer, so when the queue reaches its maximum value the incoming packets will be dropped. In each link we define four vectors: q^ℓ that represents link's queue value, s^ℓ is the rate packets arriving at the node j for the link ℓ that connects nodes j and k, r^ℓ is the transmission rate through the link and d^ℓ is drop rate at link ℓ. Component f of these vectors is the contribution of the flow f to the total value. We neglect drops that occur in transmission medium. In this model the input variable is s^ℓ, the state variable q^ℓ, and the output is the transmission rate at the link ℓ, $y = r^\ell$. We can model the queue dynamics as:

$$\dot{q}^\ell = s^\ell - h(s^\ell) - k(s^\ell, q^\ell)$$

$$y = \left[r_f^\ell\right]'$$

Where $d^\ell = h(s^\ell)$ and $r_f^\ell = k(s^\ell, q^\ell)$.

According to [51], queue dynamics can be described with a system in which, we use three different models depending of the value of $\sum_{i=1}^n q_{fi}^\ell$ and $\sum_{i=1}^n s_{fi}^l$. As showed in fig. 1.1, these are the parameters that allow us to transit from a state to another. In each case the expression of h and k will be different and the dependence on s^ℓ and q^ℓ changes depending on the different dynamic state of the queue:

- **Queue Empty** - $\sum_{f=1}^n q_f^\ell = 0:$ in this case there are no drops and all incoming packets are transmitted. In this case we have :

  $$\dot{q}^\ell = s^\ell - k(s^\ell)$$

  Where

  $$k_{fi}(s^\ell) = \begin{cases} s_f^\ell, & \sum_{f=1}^n s_f^\ell \le B^\ell \\ B^\ell \frac{s_f^\ell}{\sum_{f=1}^n s_f^\ell}, & \sum_{f=1}^n s_f^\ell > B^\ell \end{cases}$$

- **Queue neither empty nor full** - $0 < \sum_{f=1}^n q_f^\ell < q_{max}^\ell$ or $\sum_{f=1}^n q_f^\ell = q_{max}^\ell$ and $\sum_{f=1}^n s_f^\ell \le B^\ell$: also in this case there are no drops but the packets are transmitted at the head of the queue (B^ℓ bytes per unit of times). In this case we can rewrite the dynamics as:

$$\dot{q}^\ell = s^\ell - k(q^\ell)$$

Where

$$k_f(q^\ell) = B^\ell \frac{q_f^\ell}{\sum_{f=1}^n q_f^\ell}$$

- **Queue full** - $\sum_{f=1}^n q_f^\ell \geq q_{max}^\ell$ and $\sum_{f=1}^n s_f^\ell > B^\ell$: in this case the packets are transmitted at the head of the queue as in the previous case, but all the packets that exceed the maximum value of the queue will be dropped. In this case:

$$\dot{q}^\ell = s^\ell - h(s^\ell) - k(q^\ell)$$

Where

$$h(s^\ell) = s^\ell - B^\ell \frac{s_{fi}^\ell}{\sum_{f=1}^n s_f^\ell}$$

$$k_f(q^\ell) = B^\ell \frac{q_f^\ell}{\sum_{f=1}^n q_f^\ell}$$

Interest in the identification and validation of this model derives by the importance of on-line estimate the traffic load and the effective value of the bandwidth in the network, to implement a distributed optimization enhancing performance in multi-hop transmissions.

Aim of this chapter is to show how to recast and identify such hybrid model as Hybrid LPV (HLPV). Interpret the behavior of a complex system as the effect of a linear model and a scheduling variable describing nonlinearity.

1.3 HLPV identification of the traffic model

According to [59] and [60], the literature can be divided into two main groups: identification methods based on state space (SS) vs. Input/Output (I/O) models, referring to the structure of the identified model. We can divide also hybrid models in state space (SS) and Input/Output (I/O) representation. We focus our attention on the second one.

Equivalence among SS and I/O and proper discretization have been extensively discussed in literature, [53], then we notice that it is also easy to extend the above regressor to include the past terms of p_k, such as $p_{k-1}, p_{k-2}, ..,$ as required. In Piece-Wise-Affine (PWA) systems, [54], σ_k is given by

$$\sigma_k = i \quad \text{iff} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \Omega_i, \quad i = 1, \ldots, s \tag{1.1}$$

Following the discussion in [53, 55, 56, 57] a meaningful I/O representation of an HLPV model is the following one: **Definition HLPV$_{IO}$**

$$y(k) = -\sum_{i=1}^{n_a} (a_i \diamond \rho_k) y(k-i) + \sum_{j=0}^{n_b} (b_j \diamond \rho_k) u(k-j-d) + g(\rho_k) \tag{1.2}$$

where $\rho_k = [p_k, \sigma_k]$, and $\diamond$ denotes the evaluation of a function f over the trajectory of ρ_k, i.e. $f \diamond \rho_k = f(p_k, p_{k-1}, p_{k-2}, \ldots p_{k-\hat{n}}, \sigma_{k-\hat{n}}, \ldots, \sigma_k)$ (see [53]) with $\hat{n} \triangleq \max\{n_a, n_b\}$ and g a function taking into account the affine terms. In some cases, the selected I/O description can be simplified considering special forms of dynamic dependence on p, [58] allowing SS realizations without introducing dynamic dependence on p. The selected I/O description is appealing to describe the behavior of many practical applications and to derive a corresponding SS representation. The importance of subsequently determining a suitable SS representation is due to the fact that many efficient control techniques are formulated using state space information. An LPV model is characterized by the measurability of the scheduling variables. Then, under the assumption of measurability of ρ_k, it would be possible to proceed in the solution of the identification problem. Indeed, for several applications it is possible to include additional information concerning the scheduling variables. First we want to show how to simply recast the wireless ad Hoc traffic hybrid model as HLPV. We consider a single link ℓ and we assume that the number n of data flows is known and fixed to its maximum, which is equal to the number N_s of sources. Note that this is a restrictive assumption because in general n can be also time-varying, $n \in [0, N_s]$. We drop the superscript ℓ for sake of clarity. Such a hybrid model can be recast as a hybrid LPV model, by defining $p1 := \sum_{i=1}^{n} q_{fi}$ and $p2 := \sum_{i=1}^{n} s_{fi}$ and assuming that the parameter $p = [p1 \quad p2]'$ is the measurable varying parameter. This parameter takes into account an aggregated value of the queue and the total incoming flow rate to the link. For each link the input is the $n-$dimensional vector $u = s$ in which every components is random with unitary variance, the state is the n-dimensional $x = q$ and the measured output is given by the n-dimensional vector $z = [z_i]$ where

$$z_i = y_i + v, \quad i = 1, \ldots, n$$

and v is a measuring noise equally distributed on each flow f_i. Moreover, we assume that the theoretical value of the band $B^\ell = B_{max}$ is known and the maximum value of the queue q_{max}, but the effective value assumed by the band is unknown and we rename it as $b_w \in [B_{min} \quad B_{max}]$. We assume that the condition of an empty queue corresponds to $p1 < \epsilon$, with ϵ sufficiently small.

Numerical simulations have been carried out to gather data with $B_{min} = 6Mbit$, $B_{max} = 54Mbit$, $\epsilon = 0.0001$, and $q_{max} = 0.03Mbit$. We consider three flows crossing a link. The system is simulated in continuous time, but parameters are kept constant during sampling time and data have been collected at each sampling time interval. The corresponding simulated data are shown in Figure 1.2, for each flow ($i = 1, 2, 3$). The measured output is obtained by adding a uniform random noise v with unitary variance to the output y. The simulation duration is $30s$ and the sampling time is $T = 0.1s$.

The above hybrid model assumes the following $HLPV_{SS}$ form:

$$\begin{array}{rcl} \dot{x} & = & A(\rho)x + B(\rho)u \\ y & = & C(\rho)x + D(\rho)u \\ z & = & y + v \end{array} \tag{1.3}$$

where $\rho = [p' \quad \sigma]'$ is the parameter vector and the switching parameter σ is defined as follow:

$$\sigma = \sigma_k = j, \quad j = \{1,2,3,4\} \iff p \in \Omega_j$$

$$\Omega = \begin{cases} \Omega_1 & \text{if}(p1 \leq \epsilon \quad \text{AND} \quad p2 \leq b_w) \\ \Omega_2 & \text{if}(p1 \leq \epsilon \quad \text{AND} \quad p2 > b_w) \\ \Omega_3 & \text{if}(\epsilon < p1 < q_{max}) \\ & \text{OR} \quad (p1 = q_{max} \quad \text{AND} \quad p2 \leq b_w) \\ \Omega_4 & \text{if}(p1 > q_{max}) \\ & \text{OR} \quad (p1 = q_{max} \quad \text{AND} \quad p2 > b_w) \end{cases} \tag{1.4}$$

In 1.4 the sets introduced are related to the dynamical model introduced in section 1,Ω_1 is the region corresponding to the condition queue empty, Ω_3 queue neither empty nor full, Ω_4 queue full. Ω_2 is a degenerate region, a point between Ω_1 and Ω_3. Note that the above regions are depending on the unknown parameter b_w. Actually we use a first approximation of this regions where the known maximum value of the band B_{max}, instead of the effective value b_w is used.

Notice that, because the particular structure of the model, all the matrices become diagonal matrices with the same entries for $i = 1, \dots, n$: $A = diag(a_i) = diag(a)$,$B = diag(b_i) = diag(b)$, $C = diag(c_i) = diag(c)$ and $D = diag(d_i) = diag(d)$ with:

$$a(\rho) = \begin{cases} 0 & \text{if} \quad j = 1,2 \\ -\frac{b_w}{p1} & \text{if} \quad j = 3,4 \end{cases}$$

$$b(\rho) = \begin{cases} 0 & \text{if} \quad j = 1 \\ 1 - \frac{b_w}{p2} & \text{if} \quad j = 2 \\ 1 & \text{if} \quad j = 3 \\ \frac{b_w}{p2} & \text{if} \quad j = 4 \end{cases}$$

$$c(\rho) = \begin{cases} 0 & \text{if} \quad j = 1,2 \\ \frac{b_w}{p1} & \text{if} \quad j = 3,4 \end{cases}$$

$$d(\rho) = \begin{cases} 1 & \text{if} \quad j = 1 \\ \frac{b_w}{p2} & \text{if} \quad j = 2 \\ 0 & \text{if} \quad j = 3, 4 \end{cases}$$

Because the above state space model is diagonal and the entry terms are all the same, a first order scalar state-space model for each single flow, $\Sigma = \{a(\rho), b(\rho), c(\rho), d(\rho)\}$ can be taken into account by itself.

For the present example, the parameters are kept constant during the sampling time interval, so that the switching can occur only at the sampling time instant and then, both the I/O representation and the discrete-time model can be easily derived. The corresponding continuous time input-output model is obtained simply by deriving the output expression and substituting the derivative term $\dot{x}$:

$$\gamma(\rho)\dot{y}_i(t) = \alpha(\rho, \dot{\rho})y_i(t) + (\beta(\rho))u_i(t)$$

where

$$\gamma(\rho) = \begin{cases} 0 & \text{if} \quad j = 1, 2 \\ 1 & \text{if} \quad j = 3, 4 \end{cases}$$

$$\alpha(\rho, \dot{\rho}) = \begin{cases} c(\rho) & \text{if} \quad j = 1, 2 \\ a(\rho) - \frac{\dot{p1}}{p1} & \text{if} \quad j = 3, 4 \end{cases}$$

and

$$\beta(\rho) = \begin{cases} d(\rho) & \text{if} \quad j = 1, 2 \\ c(\rho)b(\rho) & \text{if} \quad j = 3, 4 \end{cases}$$

The discretization of the former model, via a backward Euler approximation with a ZOH, gives raise to a simple HLPV$_{IO}$:

$$\mathcal{A}(\delta, \rho_k, \rho_{k-1})y_i(k) = \mathcal{B}(\delta, \rho_k)u_i(k) \tag{1.5}$$

where $\mathcal{A}(\delta, \rho_k, \rho_{k-1}) = a_0(\rho_k, \rho_{k-1}) + a_1(\rho_k)\delta$ and $\mathcal{B}(\delta, \rho_k) = b_0(\rho_k)$ and the discrete time parameters are related to the continuous time parameters as follows:

$$a_0(\rho_k, \rho_{k-1}) = \begin{cases} 1 & \text{if} \quad j = 1, 2 \\ (2 - Ta(\rho_k))p1_k - p1_{k-1} & \text{if} \quad j = 3, 4 \end{cases}$$

$$a_1(\rho_k) = \begin{cases} 0 & \text{if} \quad j = 1, 2 \\ p1_k & \text{if} \quad j = 3, 4 \end{cases}$$

$$b_0(\rho_k) = \begin{cases} d(\rho_k) & \text{if} \quad j = 1, 2 \\ Tc(\rho_k)b(\rho_k)p1_k & \text{if} \quad j = 3, 4 \end{cases}$$

Clearly, the functional dependence on ρ_k, ρ_{k-1} is unknown and is one of the object of the identification. Moreover, model (1.5) can be easily recast with the structure given in (1.2) as

$$y(k) = (\alpha_1 \diamond \rho_k)y(k-1) + (\alpha_0 \diamond \rho_k)u(k),$$

with $(\alpha_1 \diamond \rho_k) = \alpha_1(p1_k, p1_{k-1}, \sigma_k)$ and $(\beta_0 \diamond \rho_k) = \beta_0(p1_k, p2_k, p1_{k-1}, \sigma_k)$.

Depending on the available a-priori information on the model, some possible different identification solutions can be adopted, as described in [44] Here we report the application of the following solution. *Off line identification.* Assume that the regions are known. Design the identification experiment choosing the input at the sources, measuring $p2$ at each link and selecting the variation of $p1$ satisfying the persistence excitations conditions as in [52]. Divide the data according to each region j and use the standard I/O LPV identification method, recasting model (1.5) in a psuedo-regressor form and identify each LPV model corresponding to a value of j.

1.4 Numerical Results

We now explain our results, mediated over seven simulations. Actually our goal is identify the model to validate it, in future works we consider identification of the effective band b_w, considering the regions unknown. In each region, we have recast model in a psuedo-regressor form [1.5], an then we have performed Least Mean Square algorithm. We consider

$$\begin{aligned} \mathcal{B}(\delta,\rho) &:= b_0(\rho) + b_1(\rho)\delta + \ldots + b_{n_b}(\rho)\delta^{n_b} \\ \mathcal{A}(\delta,\rho) &:= I + a_1(\rho)\delta + \ldots + a_{n_a}(\rho)\delta^{n_a} \end{aligned} \tag{1.6}$$

with $n_a = 2$ and $n_b = 1$. For a_i and b_i we choose polynomial functional dependence for the parameters:

$$\begin{aligned} a_i(\rho) &= a_i^1 + a_i^2\rho + \ldots + a_i^N\rho^{N-1} \\ b_j(\rho) &= b_j^1 + b_j^2\rho + \ldots + b_j^N\rho^{N-1}. \end{aligned} \tag{1.7}$$

with $N = 3$. So in each region we have identified the coefficients in the matrices:

$$\Theta := \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ b_0^1 & b_0^2 & b_0^3 \\ b_1^1 & b_1^2 & b_1^3 \end{bmatrix}$$

In Figure 1.4, Figure 1.5, and Figure 1.6 we see the simulation results of LMS algorithm in regions 1,3 and 4 respectively.

We underline that we do not have identification in region 2, this is because region 2 is a degenerate region i.e. a point, and using random input, guaranteeing persistence of excitation, we have not crossed this region. The convergence values of the coefficients in the regions (in the figures we see only the first part of the simulation and not the convergence) are:

$$\Theta_1 := \begin{bmatrix} -0.1642 & 0 & 0 \\ -0.1846 & 0 & 0 \\ 0.3903 & 0 & 0 \\ 0.1643 & 0 & 0 \end{bmatrix}$$

$$\Theta_3 := \begin{bmatrix} -0.1684 & 0 & 0 \\ -0.1902 & 0 & 0 \\ 0.3530 & 0 & 0 \\ 0.1716 & 0 & 0 \end{bmatrix}$$

$$\Theta_4 \ := \begin{bmatrix} -0.2135 & -0.0245 & 0 \\ -0.3603 & -0.0374 & 0 \\ 0.7993 & 0.1030 & 0 \\ 0.1319 & 0 & 0 \end{bmatrix}$$

In Figure 1.7 we see the global identification, i.e. we see parameters crossing the different regions during the simulation's time. In this figure we see the convergence value of previous figures.

1.5 Future Works

HLPV models inherit properties of LPV and PWA affine systems with the addition of some knowledge on the scheduling variables.

A possible off line identification algorithm has been implemented. Other more efficient solutions need to be investigated. We list now possible future algorithms that need to be implemented.

1) *On line identification.*

   a) Assume that the regions are known. Design the identification experiment choosing the input at the sources, measuring $p2$ at each link, and selecting the variation of $p1$ satisfying the constraints $C.1$ and $C.2$ and the persistence excitations conditions as in [52], and, at the same time, guaranteeing that the parameter σ is kept constant in a sufficiently long time to guarantee convergence of the algorithm. Then, use the standard I/O LPV identification method, recasting model (1.5) in a psuedo-regressor form and identify each LPV model.

   b) Assume that the regions are not known, but the number $s = 4$ is known. Design an identification experiment as in 2a). A posteriori, separate each LPV model, from the identified one.

2) *Two step identification.* Assume that the approximating regions need to be validated. First, perform a validation of the a-priori information

on p, to find approximated regions and validate them. Second, run one of the previous identification scheme 1) or 2). This solution is still under investigation.

3) *Global experiment.* In case we would like to perform an on line identification and the assumption that σ is kept constant in a sufficiently long time cannot be guaranteed, or if the model is more general than the one obtained in the present example, then a global experiment for the model given as in (1.2) need to be performed. A suitable identification algorithm determining in one-shot both the functional dependence on p and on σ need to be performed. This general solution is still under investigation.

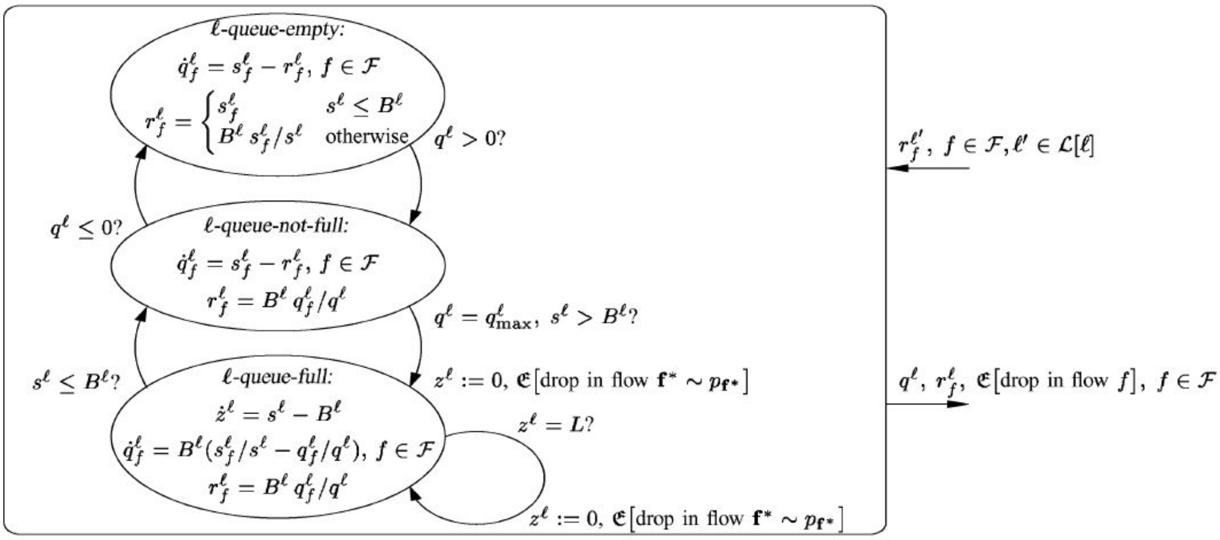

Figure 1.1: Hybrid Model for the queue at a link.
Source: [51]

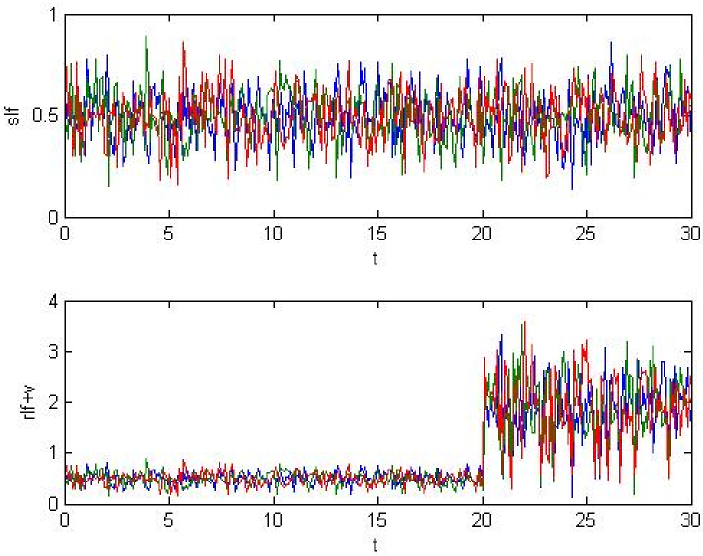

Figure 1.2: Input/output data over an interval of $30s$ and sampling time T=0.1s for each flow $i = 1, 2, 3$.

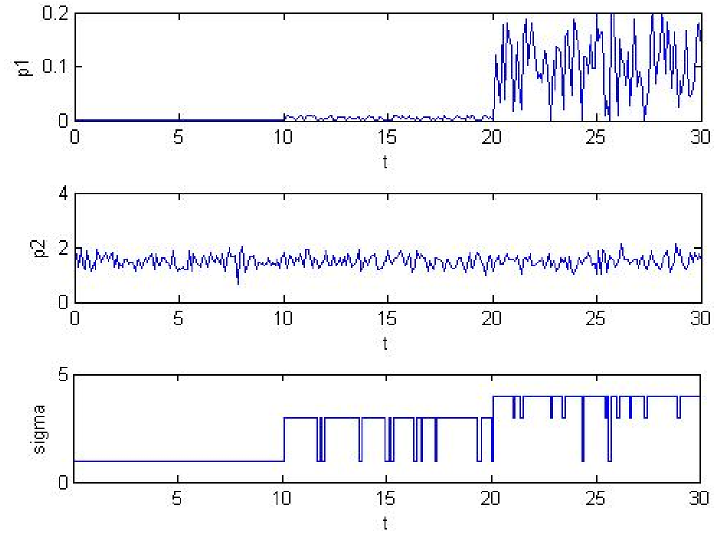

Figure 1.3: Parameters $p1$, $p2$ and σ over an interval of $30s$ and sampling time $T = 0.1s$.

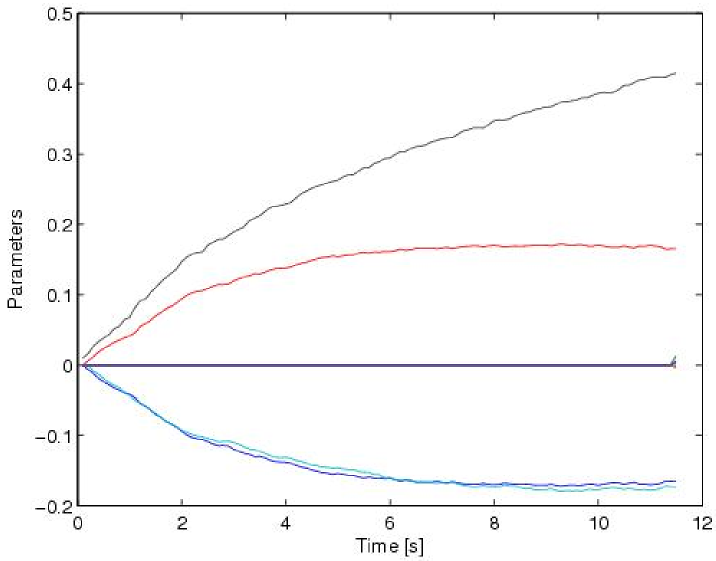

Figure 1.4: Parameters' identification in region 1.

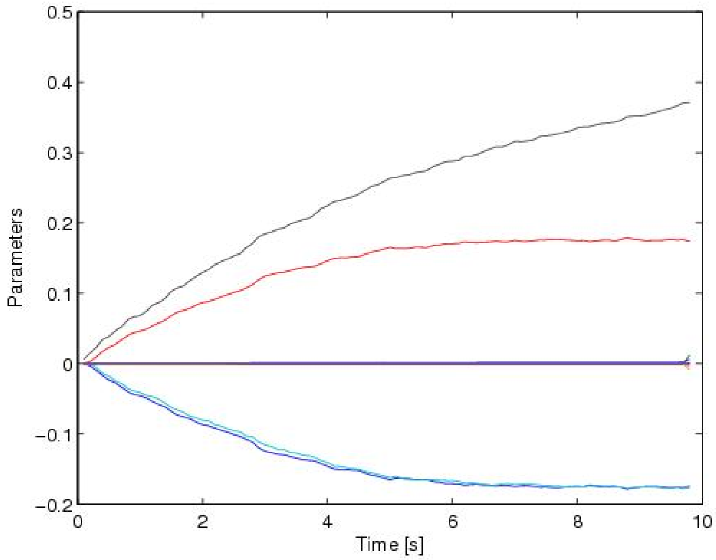

Figure 1.5: Parameters' identification in region 3.

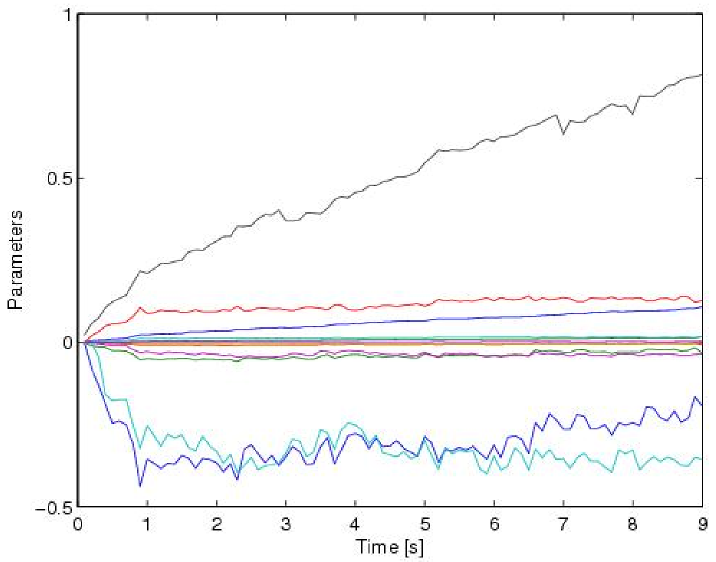

Figure 1.6: Parameters' identification in region 4.

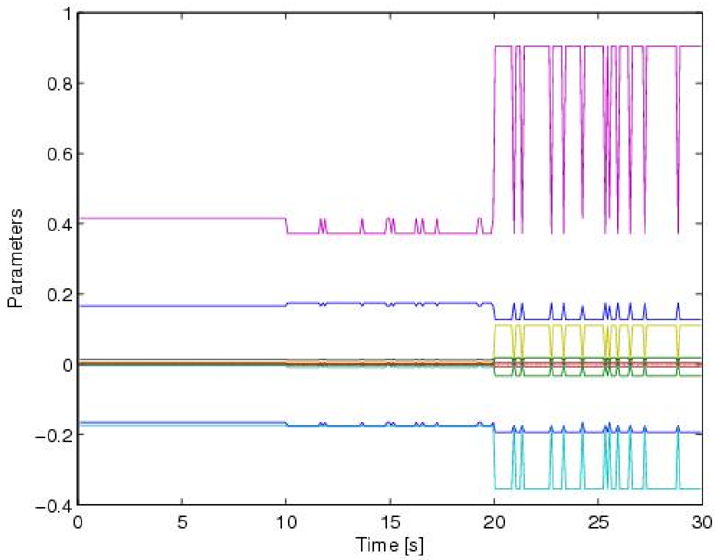

Figure 1.7: Parameters across regions.

Chapter 2

Medium Access: Contention vs. Scheduling

2.1 Introduction

In wireless networks is of primary importance deciding when a node can send a packet, and when it have to listen for receiving a packet. Making the right decision could be difficult and could cause medium wastes and degradation of network's performance in terms of throughput. We consider solutions which are *centralized* or *distributed*. *Centralized* solutions as *polling* and *TDMA (Time Division Multiple Access)* with a centralized computation are simple and they guarantee total absence of collisions, but are applicable only in small and trivial networks. In greater networks we have to consider *distributed* algorithms that could be equipped with *schedule-based* protocols or *contention-based* protocols. In schedule-based protocols the node to be transmitted is regulated in each time; scheduling could be fixed or on demand. This approach needs a synchronization mechanism. Most ad-hoc networks rely on contention-based medium access protocols, since the use of carrier sense and random backoff mechanisms is a simple and well-established solution for distributedly managing multiple-access over a shared channel bandwidth. However, it is well known that*CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)* protocols exhibit very poor performance in case of multi-hop transmissions. This is due to the inter-link interference caused by imperfect carrier sensing, i.e. the impossibility that a transmitter detects a signal interfering to the intended receiver and originating from a node out of its carrier-sense range. The collisions due to this phenomenon, called hidden-node collisions, can severely degrade the network throughput as the transmission rate of each node increases. Theoretical bounds on the attainable limits of throughput in presence of imperfect carrier sensing have been studied. In the seminal paper [61] bounds were determined for a network with arbitrarily or randomly deployed nodes under the assumption that an ideal scheduling scheme for arbitrating node transmissions can be implemented. In [62] some analysis extensions have been considered for quantifying the impact of mobility and node cooperation on such bounds. The hidden terminal problem of CSMA/CA protocol is addressed in many papers, i.e. [63]. Moreover, in [64] a time-division scheduling for ad-hoc networks is presented, with an analysis of the TDMA policy. Apart from the bound identification, a crucial problem for actual network deployments is the implementation of an efficient node coordination scheme. The scheme must be able to minimize the signaling overhead required for coordinating multiple node transmissions, while guaranteeing a significant performance improvement over CSMA/CA protocols. For improving CSMA performance has been presented several paper based on the idea of adapting contention aggressiveness basing on the basis of queue's length [65],[70],[71],[72]. Motivated by the study in [73] in which has proved that this approach can suffer high performance degradation because of channel asymmetries, packet collisions at flow receivers and dynamic traffic pattern, [74] proposes a system for proportional fairness keeping in count optimization and robustness. They relax the assumption of channel symmetry and introduce a robustness function for reducing performance degradation due to collisions decreasing channel access for avoiding interference when the number of contending flows increase. A strong point of CSMA algorithms is the flexibility in adapting when network's conditions change in topology or in traffic load. In order to become competitive, scheduled protocols should have a continuous adaptation method. In [76] and [77] it has been proposed to alternate a contention phase and a scheduled phase. In the first phase nodes exchange topology information that are used for computing the scheduling in the next phase. The problem in this approach is that change in topology or traffic load could not be aligned with the phases of the algorithm. In [79] a *topological allocation* through a distributed algorithm that can operate within scheduled and contention-based MAC protocols is proposed. This algorithm evaluates node's topological persistence that is the fraction of time that it is permitted to transmit, and after identifying ideal persistences for a given topology and traffic load, topological persistences is given as input to MAC protocols. This approach has several limitations: the algorithm assumes a priori knowledge of traffic demands and local topology; it does not accommodate changes in topology or traffic demands; the algorithm requires synchronization. This algorithm is improved in [78] in which node's persistence is also computed but a scheduled MAC protocol able to adapt to change in topology and traffic load is presented. Our approach is to find a distributed resource allocation algorithm combining TDMA for grouping contending nodes in non-interfering sets and CSMA/CA for managing the channel access behind a set. This pre-allocation mechanism of channel holding times can significantly reduce the channel wastes due to hidden node collisions and has been recently considered also in some standardization task groups working on mesh networks and literature, for optimizing both the network capacity and the energy consumption [80] in Zigbee networks, or coping with bidirectional traffic flows over chain topologies exploiting network coding [81]. In the rest of the chapter we focus on the problem of determining the optimal resources allocation, that is how many resources guarantee to each node in order to obtain the best performance in terms of throughput. The rest of the chapter is organized as follows:

- In section 3.4 we formulate the problem;
- In section 2.3 we present a solution that consists in scheduling potentially interfering transmissions in different time slots, while allowing in-range nodes to transmit in the same time slot but subject to a CSMA/CA mechanism that avoids collision. The problem is formulated in terms of a map coloring problem. Since color allocations may leave some level of contention, by assigning the same color to nodes in radio visibility, a game theoretical study of intra-slot contention is

introduced.

- Motivated by results obtained in section 2.3 showing that combining contention and scheduling gives an improvement on both approaches CSMA and TDMA, we asked ourselves what is the optimal degree between contention and scheduling. We present an algorithm which finds a conveniently partition of the network, scheduling the transmissions of different subsets in different time-slots. We apply an optimization algorithm for deciding how many time-slots are guaranteed to each subnet. The number of subsets giving the best performance in a particular network is the optimal degree between contention and scheduling for that network.

2.2 Problem Formulation

We consider a single channel radio network made of a set V of nodes distributed uniformly over a given area. Each node $i \in V$ can communicate only with a subset of adjacent nodes V_i. We say that i is (radio) *visible* only to the nodes in V_i. Differently, i is *hidden* to the remaining nodes in $V \setminus V_i$. We assume that radio visibility is symmetric and that the communication between pair (i, j) of adjacent nodes presents a maximum transmission rate r_{ij}, function of the distance between nodes i and j and of the possible presence of obstacles. The channel time is divided into elementary allocation units called slots. Each slot is able to accommodate a random backoff delay and the transmission time of the maximum allowed packet size at the minimum rate (followed by an explicit acknowledgment). Only the subset of nodes to which a generic channel slot is pre-allocated are enabled to perform the CSMA/CA function for transmitting on that slot. The slot allocations are maintained on a per-frame basis: being x the total number of allocation slots, a sequence of x consecutive slots is a channel frame in which, slot by slot, the same sorted list of nodes are enabled to transmit. Figure 2.1 shows an example of medium access in a network with 5 nodes, in which a channel frame of 3 slots is considered. In the first slot, where only station 1 and 2 can access the medium, station 1 wins the contention (i.e. extracts

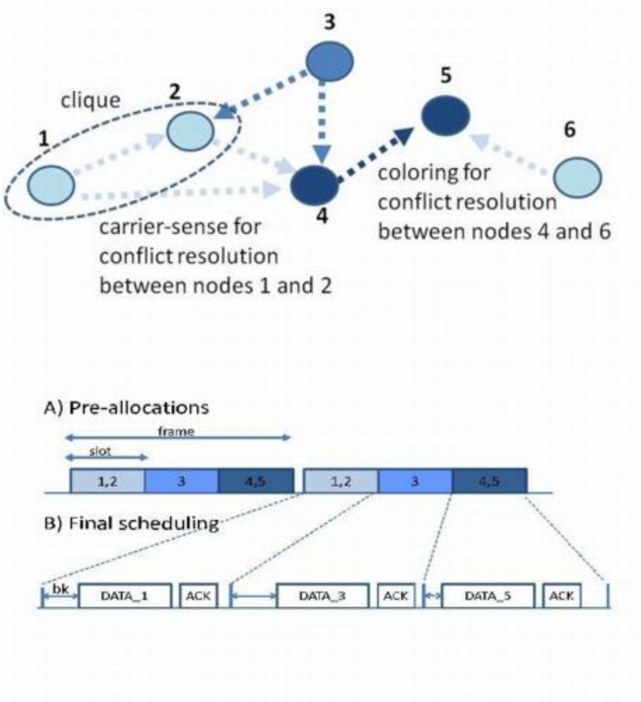

Figure 2.1: An example of medium access in a network with 5 nodes and a frame composed of 3 slots.

the lowest backoff delay). The second slot is used by station 3 only, while the third slot is reserved to the contention between stations 4 and 5. The reason for pre-allocating channel slots to a sub-set of stations (thus grouping in independent sets the stations allowed to transmit simultaneously) is the mitigation of the hidden node problem. For example, if stations 1 and 3 are hidden to each other (as shown in the figure) and wish to transmit to station 2 (which is able to hear both the stations), the previous allocation avoids any collision possibility. Conversely, transmissions originated by stations 1 and 2, which are in reciprocal visibility, are separated by the CSMA/CA protocol. We formally define the problem of slot allocations in what follows.

2.2.1 Network Structure and Traffic Model

We represent the network structure through an edge labeled graph $G = (V, E)$. Specifically, the nodeset V includes all the nodes i of the network and the edgeset E includes all the pairs of adjacent nodes (i, j) that are in radio visibility to each other. Each edge $(i, j) \in E$ is labeled with its maximum transmission rate r_{ij}.

Since the system time is slotted, we also model the traffic source at each node in terms of per-slot packet probability. Specifically, we assume that each node i has a fixed probability λ_i to generate a packet in each slot. In order to avoid interactions with the routing protocol, we consider only one-hop packet deliveries. Packets are destined to a randomly selected node among the neighbor ones. For isolated nodes, i.e. nodes without neighbors, the traffic is assumed to be broadcast. In addition, we assume that the packet size is of a fixed value D for all the nodes, whose transmission time is always compatible with the slot size.

2.2.2 Resource Allocation

We assume that the reader is familiar with CSMA/CA protocols, which regulate the final channel access within an allocated channel slot. Although most CSMA/CA protocols use a slotted backoff scale for efficiency reasons and for implementation limits (since the carrier sense cannot be instantaneous), we assume that backoff values are uniformly extracted in a continuous range $[0, b]$, thus implying that collisions cannot be originated by the extraction of two identical backoff values. In order to implement a slot allocation mechanism, two basic functionalities have to be provided in the network: i) a mechanism for inferring the network topology; ii) a mechanism for keeping a common time reference among the nodes. For both the aspects, we consider that an independent signaling channel is available (managed by a random access scheme) and nodes in radio visibility can exchange control information (e.g. the list of neighbor nodes). We also assume that nodes do not have data storage constraints, while processing capabilities may depend on the specific network scenario.

2.3 Utility-Based Resource Allocations in Multi-Hop Networks

In this section, the problem is formulated in terms of a map coloring problem, which has a vast and well established literature [83, 84]. Therefore, we simply adapt some existing coloring algorithms to ad-hoc networks, by trying to identify the most effective trade-offs between complexity, signaling overheads and performance gain. Since color allocations may leave some level of contention, by assigning the same color to nodes in radio visibility, a game theoretical study of intra-slot contention is also introduced. In the above context, the two main problems of our interest are the following ones.

Problem 1. *Determine a distributed protocol that sets the number x of slots in a frame and the slots allocations, in order to maximize the per-node throughput in saturation conditions, i.e. in presence of greedy sources whose packet generation rate is $\lambda_i = 1$.*

Problem 2. *Determine a distributed protocol that allows the allocations of slots of a frame in order to minimize the average delivery delay for generic source rates λ_i.*

2.3.1 Solution Approach

In this section, we discuss the possibility of reducing Problems 3 and 4 to a set of Minimum Graph Coloring (MGC) problems.

2.3.2 Graph coloring

Let $G(V, E)$ the network graph including the set V of nodes distributed over a given area and the set E of edges connecting radio visible nodes. Each node $i \in V$ is labeled with the number a_i of slots to be allocated to it according to the traffic it must support. Generally speaking a single slot is allocated to each node. However, in case of heterogeneous packet generation rates λ_i (which may actually model nodes belonging to heterogeneous number of paths and aggregating traffic packets generated by multiple sources), some nodes may require more slots to drain their traffic.

We define as *incompatibility graph of type I* the node labeled graph $H_E(V, F_E)$ whose edges in F_E join the pair of nodes $(j,k) \in V \times V$ whose frames may collide if transmitted simultaneously. By definition $H_E = G^2$, that is,

$$F_E = \{(j,k) : \exists i \in V \text{ s.t. } (j,i),(i,k) \in E\}.$$

We can see Problems 3 and 4 as a MGC problem that determines the minimum cardinality of a coloring of the nodes of H_E such that each node is colored with as many different colors as its label. Then, each color corresponds to a specific slot allocated to the node on the frame.

Obviously, the network transport capacity is critically affected by the cardinality x of a coloring of H_E, since each node i receives a_i transmission chance only every x slots. For example, assuming a uniform transmission rate r among all the edges, the node transmission rate is upper bounded by $a_i \cdot r/x$.

In defining the incompatibility graph of type I, we have not considered the carrier sense functionality that intrinsically makes orthogonal (i.e. non-interfering) the transmissions between visible nodes. Edges connecting visible nodes in the incompatibility graph of type I may result redundant and some, if not all of them, may be removed, possibly drastically reducing the number of colors necessary for the graph. In this context, we define as *incompatibility graph of type II* the node labeled graph $H_\emptyset(V, F_\emptyset)$ whose edges in F join the pair of nodes $(j,k) \in V \times V$ that are of the reciprocally hidden and whose frames may collide if transmitted simultaneously. By definition $H_\emptyset = G^2 - G$, that is,

$$F_\emptyset = \{(j,k) : \exists i \in V \text{ s.t. } (j,i),(i,k) \in E \text{ but } (j,k) \notin E\}.$$

Removing edges from the incompatibility graph can make the per-node transmission rate S_i (also called node throughput) heterogeneous, even in the case of uniform rate r and a_i allocations. Indeed, nodes receiving slots not shared with visible nods receive a throughput bounded by $a_i \cdot r/x$, while nodes sharing the slot with neighbor nodes experience, in the worst case, a throughput reduction equal to the number of contending nodes.

The graph H_E and $H_\emptyset$ define the two extreme cases in which either all or none of the pair of reciprocally visible nodes are considered. Obviously, even intermediate situations may be defined. Let 2^E be the power set of the edgeset E.

For each $e \in 2^E$, we can consider the coloring problems of the incompatibility graphs $H_e = (V, F_\emptyset \cup e)$, and the per-node and aggregated throughput, S_i^e and $S_{tot}^e = \sum_{i \in V} S_i^e$. Then, the optimal coloring scheme is the coloring referring to the incompatibility graph H_e which maximizes the value of S_{tot}^e, for $e \in 2^E$.

2.3.3 Throughput assessment

Consider a H_e, for $e \in 2^E$, graph. For each node $i \in V$, let us define its *associated after coloring clique* as the maximal clique on the graph G that includes i and is formed by nodes of the same color of i. Let d_i^e be the size of such a clique and let $a_i = 1\ \forall i$.

Then, if we assume a uniform rate r for all the links in E, we can guarantee a per-node *collision free throughput*

$$\rho_i^e = \frac{r}{x^e d_i^e} \tag{2.1}$$

where, x^e is the number of colors used in H_e. The rationale behind (2.1) is the following. For each node $i \in V$, we have to share the slot associated to its allocated color with $d_i^e - 1$ contending nodes. On average, node i will win the backoff contention only once every d_i^e frames. Collisions with adjacent nodes are avoided by means of the carrier sense functionalities, while collisions with other nodes using the same colors are avoided by the coloring algorithm (which re-assign the same slot only when nodes are distant more than two hops).

Given a graph H_e, the maximum number of needed colors is upper bounded by $\Delta^e + 1$, where Δ^e is the maximum node degree of the graph. In addition, coloring H_e with at maximum $e + 1$ colors can be easily attained with a distributed protocol, such as *Brooks-Vizing*, [87]. The following condition then holds:

$$\frac{r}{\Delta^E+1} \quad \geq \quad \max_{i \in V} \frac{r}{(\Delta^e+1)d_i^e} \qquad \forall e \in 2^E$$

Since Δ^e is an upper bound on the number of needed colors, the previous condition implies that the lower bound of the collision-free throughput guaranteed to each node is higher for the incompatibility graph H_E.

Let us now consider the aggregated collision-free throughput ρ^e_{tot}. After coloring, the throughput sum perceived by all the nodes belonging to each clique is obviously r/x^e, thus resulting in a total throughput equal to:

$$\rho^e_{tot} = \frac{r}{x^e} c^e$$

where c^e is the total number of cliques resulting from the coloring of the incompatibility graph H_e. Obviously, when $H_e = H_E$ such a number corresponds to the number of nodes n (since 1-hop nodes are allocated on different channels). It follows that we can also express the average per-node throughput $E[\rho^e]$ as $\rho^e_{tot}/n = \frac{r}{x^e E[d^e]}$ (where $E[d^e] = n/c^e$ represents the average after coloring clique size).

For each $e \in 2^E$, we note that the collision free throughput is only a guaranteed lower bound on the actual throughput S^e_{tot} that we can obtain coloring the graph H_e. In fact, S^e_{tot} can be a higher throughput. Let x^e be the number of colors used for H_e. Consider a generic node i and let x_i the number of colors used for coloring its adjacent nodes on H_e. When $x_i < x^e + 1$, we may obtain a transmission rate for i greater than the one guaranteed by the collision free throughput if we allow i to transmit during the slots associated to its color and to colors different form the ones of its adjacent nodes on H_e. If i is the only node with such a privilege, we will obtain a throughput (with no collision) higher than the collision free throughput. Differently, if we concede the above transmission privilege to all the nodes with $x_i < x^e + 1$, we cannot guarantee that extra slot allocations result in a free transmission. Nevertheless, we obtain an overall throughput higher than the collision free throughput, as long as each node does not transmit to the slots associated to the adjacent nodes on H_e. If the competition for extra slots is extended to all the slots of the frame (thus including the potential interfering nodes), the collision-free throughput cannot be guaranteed anymore. Therefore, we are currently investigating on the risks and benefits of enabling extra slot allocations, by means of a game-theoretical analysis.

2.3.4 Coloring Algorithms

Coloring algorithms have been widely explored In literature. Some examples of popular solutions are the *Luby*'s algorithm [85], the *Johanson*'s algorithm [86], and their variants [87].

We consider here the algorithm proposed in [75]. A preliminary exchange of control information is necessary for evaluating at each node i the global degree of the network Δ or the local number of neighbors δ_i. Let x_{max} the global or the local maximum number of available colors. According to the basic algorithm, each uncolored node has to perform the following steps:

1. *First coloring* Randomly pick a color from a list of available colors.

2. *Conflict Resolution* If none of your (1-hop or 2-hop) neighboring nodes has chosen the same color, keep it as definitive color, otherwise remove it form the list and try again the next step.

3. *List update* If the color list is empty, add new colors. The list is updated starting from $\min\{c+1, x_{max}\}$ color, where $c = \max\{\text{neighboring node colors}\}$.

This algorithm is called *SC* algorithm, for recalling its characteristic of first *Selecting* a color and then *Comparing* the selected color with potential interferes.

In order to optimize the number of used colors, we also considered a simple modification of the previous scheme. Instead of randomly picking a color from the available ones, each node first updates the list of available colors (as in the third step of the previous scheme) and then selects the color with the lowest index. This scheme is called *CFA* algorithm, since it is based on *Choosing* the *First Available* color. While the nodes belonging to different cliques cannot interfere because they are allowed to transmit only in different frame slots, nodes belonging to the same after-coloring clique can hear each other and have to use a CSMA protocol to share the same frame slot.

In section 2.3.3 we have assumed that nodes sharing the same frame slot have the same probability to win the contention, i.e. in long terms they obtain the same number of transmission grants. However, this assumption can be restrictive. Indeed, nodes could be motivated in using heterogeneous access probabilities[1] for delivering more traffic. For examples, nodes relaying traffic of many other nodes could need more channel resources than nodes transmitting only their own packets, for preventing buffer overflows and packet drops. Therefore, here we consider a game-theoretical analysis of intra-clique contention. While the impact of heterogeneous access probabilities have been considered for 1-hop wireless networks [88], according to our knowledge this problem has not been explicitly addressed for multi-hop networks.

2.3.5 Station Utility

We consider a set of N nodes belonging to the same clique and sharing the same frame slot. Let τ_i be the slot access probability (i.e. the node strategy) and let λ_i be the per-slot packet generation probability of a generic station i (with $i = 1, 2, \cdots N$). Since we are not modeling traffic paths and routing schemes, we assume that λ_i takes into account both the node internal and external (i.e. coming from other nodes) packet enqueuing rate and that each node selects (uniformly) all the neighbor nodes as relays. We also assume that during the network topology discovery phase, nodes exchange information about their traffic patterns, thus notifying the resulting λ_i values to the neighbor nodes.

A key aspect to be investigated is the definition of the utility function driving

[1]Heterogeneous access probabilities can be supported by using a protocol such as the IEEE 802.11 EDCA, which differentiates the channel access probabilities of different traffic classes, or by configuring non-standard access parameters by means of open-source drivers accessing the card configuration registers.

the configuration of τ_i parameters. In single hop networks, such a utility has been usually expressed in terms of transmission throughput percevied by each contending node. Being $p_i = 1-\prod_{j=1}^{N}(1-\tau_j)/(1-\tau_i)$ the collision probability suffered by station i because of other node channel accesses, the transmission throughput μ_i of station i can be expressed as $\tau_i(1-p_i)$ packets/slot. Under a utility function $J_i = \mu_i$, the station best response corresponds to play $\tau_i = 1$ and the channel resources can be completely wasted whenever at least two different stations play this strategy.

The transmission throughput has been already proved to be an inconsistent utility function for bi-directional traffic scenarios [88], where nodes are interested not only in transmitting the locally-generated traffic packets, but also in receiving the packets generated by the peer application. Therefore, it is even more inconsistent for multi-hop networks, where the transmission throughput of a given node directly affects the throughput of the neighbor nodes using such a node as a relay. In other words, nodes cannot be only interested in maximizing their transmission rate, since they need to leave resources to the neighbor nodes which will carry their traffic towards multi-hop destinations.

These considerations motivate the definition of a utility function describing the whole transport capacity of the clique. Since in our model each node of the clique exploits all the neighbor nodes as relay, we assume that the clique nodes are interested in avoiding buffer overflows or bottlenecks at each node of the system. In other words, being Q_i the average queue length of node i, we define $J_i = -\max_{i=1,2,\cdots N} Q_i = J \quad \forall i$. Being K the buffer size of each node (expressed in terms of maximum number of packets that can be accommodated in the buffer), we can express the average queue length of node i as:

$$Q_i = \begin{cases} min\{\frac{\lambda_i}{\mu_i - \lambda_i}, K\} & \mu_i > \lambda_i \\ K & \mu_i \leq \lambda_i \end{cases} \tag{2.2}$$

Since the queue length provides a negative utility (i.e. it represents a cost for the system in terms of pending packets to be delivered out of the clique), nodes have to minimize such a length in order to maximize their payoff.

2.3.6 Station Best Response

An important aspect of our utility function formulation is that such a function is common to all the stations, In fact, since each node has to relay on other nodes for ultimately delivering and receiving its own traffic, there is no point in implementing greedy behaviors that prevent neighbor nodes from accessing the allocated frame slot. Moreover, such a common utility, which represents the clique transport capacity, prevents other malicious behaviors such as the signaling of wrong traffic generation rates λ_i. Being the node induced to collaborate by the need of maximizing a common utility, we also assume that in each notification message about neighbor nodes and traffic rates nodes can also announce their current strategy τ_i[2].

Consider now a tagged node j of the clique. On the basis of the λ_i and τ_i parameters signaled by all the clique nodes $i = 1, 2 \cdots N$ $i \neq j$, and on the basis of its traffic rate λ_j, the tagged node can implement a best response strategy based on the evaluation of the per-node queue lengths as a function of the tagged node strategy τ_j. Figure 2.2 shows an example of such evaluations in a scenario with four nodes in which all the traffic rates are constant and equal to 0.05 packets/slot and $K = 100$. The curves have been obtained assuming that the tagged node is node 1 and that $\tau_2 = 0.15$, τ_3=0.2, and τ_4=0.25. While Q_1 is a not increasing function of τ_1 (since increasing the channel access rate node 1 has more chances to deliver its traffic), all the other queue functions are not decreasing (since neighbor nodes experience higher collision rates as the tagged node increases its access). For minimizing the maximum queue length of the system, node 1 has to play $\tau_1 = \tau_2 = 0.15$ that means that has to equalize its queue to the length of the worst neighbor queue. Being $Q_j(\tau_j)$ not increasing with $Q_j(0) = K$ and $Q_i(\tau_j)$ not decreasing with $Q_i(1) = K$ $\forall i$, that such an equalization it is always possible for at least one strategy τ_j. Such a strategy is unique if the intersection point between Q_j and the highest Q_i curve is in the strictly monotoning range of the curves (as shown in the figure), while it corresponds to a range of possible values when the intersection is on the flat region of the curves (i.e. for $Q_i = K$). In

[2]Such a notification is in principle not necessary, since each node can independently estimate the access probability of other nodes from channel observations.

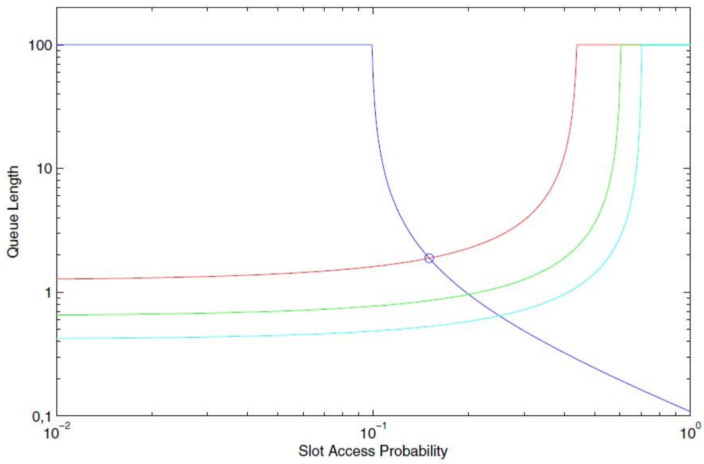

Figure 2.2: Queue Length at each node, as a function of the strategy of a given contending node.

this last case, the tagged node could decide to play the highest τ_j value of the range.

Generalizing the previous considerations to the case of heterogeneous λ_i parameters, we can implement a best response strategy as follows:

$$\tau_j^{br} = \begin{cases} \frac{\lambda_j \tau_x}{\lambda_x - \tau_x(\lambda_x - \lambda_j)} & Q_x < K \\ \frac{\lambda_j}{\prod_{i \neq j}(1-\tau_i)} \frac{K+1}{K} & Q_x = K \end{cases} \tag{2.3}$$

being x the index of the node experiencing the worst congestion (i.e. the longest queue). It can be proved that by repeating such a best response strategy for all the nodes at the reception of the announcement messages, in a finite number of steps the system converges towards a Nash Equilibrium (NE) point, in which all the clique queues are equalized. The equilibrium point is not unique and depends on the initial strategies of the nodes. The details of such analysis are in [82].

2.3.7 Numerical Results

In order to compare the effectiveness of the slot pre-allocations in

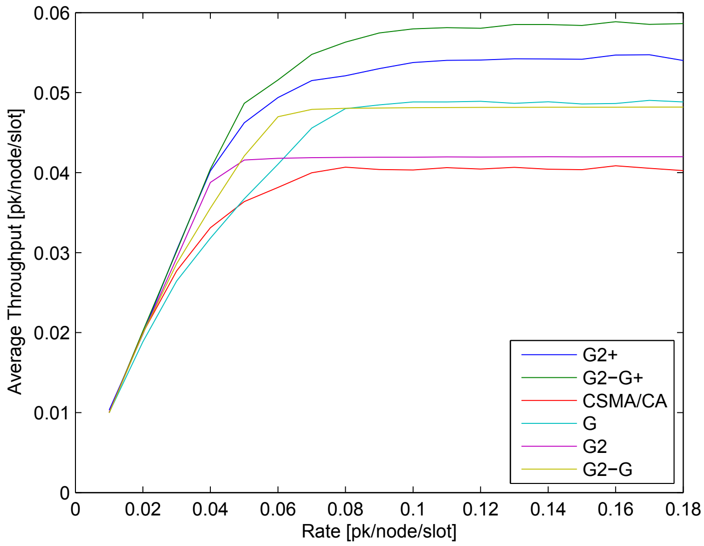

Figure 2.3: Average throughput under the SC coloring scheme, for different incompatibility graphs and comparison with standard CSMA/CA.

improving the CSMA/CA performance in multi-hop networks, we run several simulations, including both the network coloring phase and the data transmission phase. Obviously, the throughput performance perceived in a given network topology are critically affected by the final map of colors and by the node source rates. For the same network topology, such a final map depends on the random color selections and/or on the node initialization choices. Therefore, each run performance can be different and has to be averaged. Note also that in our simulations, we do not consider dynamic node activations and de-activations, thus running the coloring phase only at the beginning of the simulation and maintaining the color map for the rest of the simulation time.

We considered random network topologies of 30 nodes deployed over an area of $10 \cdot 10m^2$, with a transmission range of $3m$. We observed that the CFA scheme requires on average 15 different colors when the incompatibility graph is $H_E = G^2$, while it uses 8 colors only for the graph $H_\emptyset = G^2 - G$. Conversely, the SC coloring scheme resulted in an average number of colors equal to 24 for the $G^2 - G$ case, and 17 for the $G^2 - G$ case. The higher number of adopted colors has two different effects: on one side it increases the frame length, thus resulting in a lower rate of node transmission chances; on the other side it reduces the contention level between 1-hop nodes in case of $G^2 - G$.

After that each node has been colored, we simulated 5000 channel slots. At each slot, three different steps are considered: i) generation of traffic packets, ii) selection of transmitting nodes; iii) verification of transmission outcomes. At the first step, a new packet is generated in the transmission buffer of each node i with a fixed probability $\lambda_i = Rate\ \forall i$. The destination node is uniformly extracted among the neighbors and no buffer size limit is considered. At the second step, the simulator processes all the nodes whose color corresponds to the current slot and extracts uniformly a backoff value for resolving potential contentions. Since the traffic rate of each node is constant, the backoff range of each node is constant too, in order to implement a uniform slot access probability within the after-coloring cliques. All nodes winning the contention are labeled as transmitting. Finally, if the neighbors of the intended receiver are not transmitting, the transmission is considered successful and the packet is removed from the buffer. Otherwise, the packet remains in the buffer until a maximum number of retries (set to 3) has been reached.

Figures 2.3 and 2.4 compare the per-node average throughput measured in our simulations, under the SC and CFA scheme, for different incompatibility graphs (namely, G, G^2 and $G^2 - G$). In both the figures we also plot the CSMA/CA performance. The throughput has been averaged by considering ten different coloring runs of the coloring schemes, referring to the same network topology. From the figures we can draw some interesting observations. First, coloring G can be useless, because the carrier sense functionality is already able to avoid interference among adjacent nodes. For the CFA case, the performance obtained under the G coloring are even worse than the ones obtained with the CSMA/CA protocol, because the slot allocations may synchronize hidden nodes for lower packet generation rates. Second, coloring G^2 can be more efficient (CFA case) or less efficient (SC case) than coloring $G^2 - G$, according to the network topology and to the effectiveness of the col-

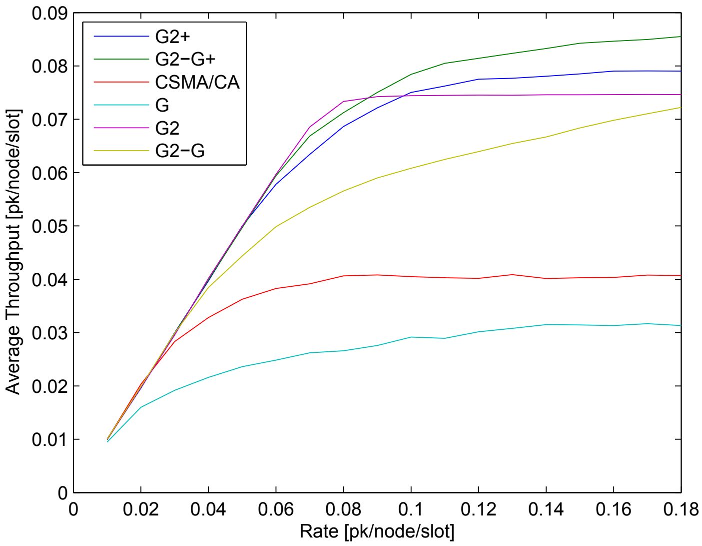

Figure 2.4: Average throughput under the CFA coloring scheme, for different incompatibility graphs and comparison with standard CSMA/CA.

oring scheme in selecting a limited number of colors and/or leaving a limited number of bottlenecks. Third, when additional channel slots are allocated as described in section III-B (the G^2+ and $G^2 - G+$ curves of the figures), the network throughput performance can be further improved.

Finally, to validate the throughput bounds discussed in section III-B, table 2.1 compares the saturation (collision-free) theoretical bounds with the best throughput values measured (on a given topology) under 10 different CFA coloring runs.

2.3.8 Conclusions

Coordination among nodes in ad-hoc networks can significantly improves the transport capacity of the networks, in comparison with simple uncoordinated CSMA/CA protocols. A simple form of coordination can be provided by pre-allocating temporal intervals in which different sets of nodes are allowed to access the shared wireless medium. We have analyzed different solutions introducing such a pre-allocation on the basis of a neighbor discovery protocol

Topology	H_e	x_e	c^e	$\hat{\rho^e}$/r	$E[\rho^e]$/r
1	G^2	16	30	0.0624	0.0625
1	$G^2 - G$	5	12	0.0792	0.0800
2	G^2	13	30	0.0768	0.0769
2	$G^2 - G$	5	11	0.0730	0.0733
3	G^2	15	30	0.0666	0.0667
3	$G^2 - G$	5	13	0.0863	0.0867

Table 2.1: Measurements and estimates of throughput.

and distributed coloring schemes requiring limited signaling overheads. We showed that the performance of these schemes can be critically affected by the considered incompatibility graph, trading off the contention-level experienced by 1-hop neighbors and the orthogonality guaranteed to hidden nodes. We are currently working on coupling our framework with a network coding scheme, able to further improve the transport capacity of each after-coloring clique in the network. Further extensions are also considered for studying different traffic models, based on per-path (multi-hop) traffic flows. Dynamic traffic, changing topologies, mobile nodes and comparison with existing alternative algorithms is also under investigation. Moreover we are investigating the possibility of studying the optimal solution of the posed problems via a game theoretical approach.

2.4 Optimal Resource Allocation in Multi-Hop Networks: Contention vs. Scheduling

In this section we present a solution that partition the network and provide an heterogeneous number of time slots that depends on the total traffic generated in each subset. The slot number allocation is based on an optimization criterion devised to maximize the minimum throughput perceived in the network. In the context described in section 3.4, the main problems to be tackled in this section are the following.

Problem 3. *Determine the best partition of the network i.e.: the best number of subsets in which divide the network and how divide nodes in the subnets.*

Problem 4. *Determine the best number of time slots to grant to each subnet in each frame.*

2.4.1 System Model

The correlation between nodes in a mesh network (either due to spatial reasons, or to traffic routes) has a more important effect on the final network performance than the specific evolution of the backoff counter. Therefore, we propose a system model based on the following simplifying assumptions:

- The time for the backoff countdown and medium sensing is negligible;
- The probability to have the minimum residual backoff counter is uniform for all the contending stations, while the probability to have exactly the same residual backoff expiration time is zero.

According to the first assumption, all the contending stations complete their backoff count-down sequentially (as determined by their backoff counters) but with negligible time intervals between their transmission grants. In other words, the time interval Δ between consecutive transmission grants is assumed equal to 0 when it accounts for consecutive backoff countdown, and equal to P for completing the slot when all the stations are in a transmitting or frozen state. When Δ is equal to P, at the end of the slot all the stations with non-empty queues switch synchronously to a contending state. The second assumption implies that visible stations never collide and that the start of a new slot is a regeneration instant, since all the stations with traffic are in a contending state at the beginning of the slot with the same residual backoff distribution.

2.4.2 Transport Model

Under the previous assumptions and that all the stations are saturated we can evaluate per-node throughput determining transmission events and occurrence probability as seen in [89]. The idea is characterizing the occurrence probability of different *transmission events*. A transmission event is the collection of the state of each node at a generic time slot, where the node state is considered equal to 1 if the node is transmitting during the time slot, and equal to 0 otherwise.

Let A be the set of all the possible values $\boldsymbol{a}$ of transmission events, $a(i)$ the binary transmitting/frozen state of node i in the transmission event $\boldsymbol{a}$, and $p_A(\boldsymbol{a})$ the total probability of event $\boldsymbol{a}$. For a given transmission event $\boldsymbol{a}$, the number of nodes able to receive successfully a frame sent by node i is $S_i(\boldsymbol{a}) = \sum_{j=1}^{n} a(i) g_{ij} \prod_{k=1}^{n}(1 - g_{jk}a(k))$, where we consider that the same frame can be received by all the 1-hop neighbors j for which none of the relative 1-hop neighbors k is active. We consider the time frame divided in N time slots and we grant α_i time slots to the subnet i, so the nodes in this subnet have transmission opportunity only in α_i time slots every N. This implies that for computing the average transmission throughput we have to consider a scale factor α_i/N It follows that the average transmission throughput S_i for each node i (in terms of packets/slot) can be obtained as:

$$\begin{aligned} S_i &= \frac{\alpha_i}{N} \sum_{\boldsymbol{a} \in A} p_A(\boldsymbol{a}) S_i(\boldsymbol{a}) \\ &= \frac{\alpha_i}{N} \sum_{\boldsymbol{a} \in A} p_A(\boldsymbol{a}) \sum_{j=1}^{n} a(i) g_{ij} \prod_{k=1}^{n} (1 - g_{jk} a(k)) \end{aligned} \tag{2.4}$$

2.4.3 Channel Access Model

In order to implement our hybrid TDMA/CSMA access scheme, we organize the channel access time in periodic frames composed by N time slots. Time slots are not exclusively allocated to a single node, but rather are shared among the set of nodes belonging to the same subnet. Contention is still used for arbitrating the channel accesses among the subnet nodes.

We define *partition of a network* the partition of the nodeset V. A partition of V is a set of subsets of V such that:

- All sets in V are pairwise disjoint: $V_i \cap V_j = \emptyset \ \forall i \neq j$

- The union of all the sets forms the whole set $V = \bigcup V_i$
- None of the sets in V is empty: $V_i \neq \emptyset \ \forall i$

Our goal is to find both the best partition of the network and the best resources allocation for improving the network performance in terms of throughput with respect to both pure TDMA and pure CSMA based approaches.

2.4.4 Network Partitioning

Network partitioning substantially affects the network performance. The number of subsets in which we divide nodes decides how much the medium access will be scheduled or with contention. In fact if we consider only one subset, that is the entire network, in all the time slots all nodes will contend according to CSMA, so this is the case of total contention. Conversely if we consider a partition formed by n subsets in a network formed by n nodes, all the subsets are formed by one node only. In this case, each node will transmit without contention and the system is entirely scheduled. All the division in a number between 1 and n gives medium access partially scheduled and partially with contention; our goal is to find the number of subsets with the best performance. After solving the problem on how many subsets need to be selected, we have to decide how to divide nodes in subsets. Network performance may change considerably in different placement of nodes. For example, because hidden nodes degrade CSMA performance, we would like to assign hidden nodes to different subsets, so that they will never collide because they can access the channel in non-overlapping time intervals.

Consider for example a simple network topology with three nodes in a row. With pure CSMA, it is very likely that the edge (non-visible) nodes will achieve an almost zero throughput due to the fact that when the first edge node starts its transmission, the opposite one will decrement its backoff to zero before the end of the ongoing transmission (thus resulting deterministically in continuous collisions). If we consider a *fully connected network* formed by 20 nodes we can observe that applying CSMA/CA the network reach an aggregated throughput of 0.634 pk/slot. If we divide the network

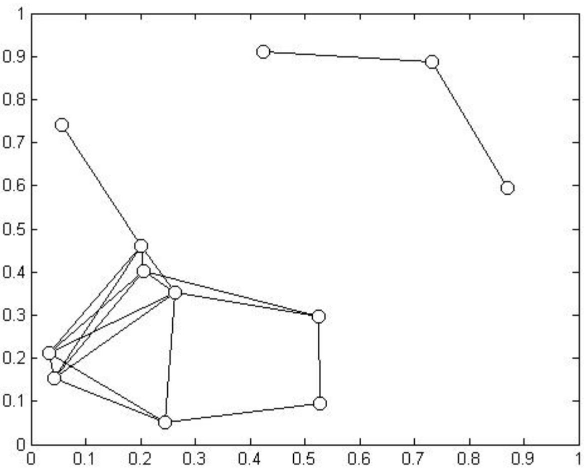

Figure 2.5: Network Topology.

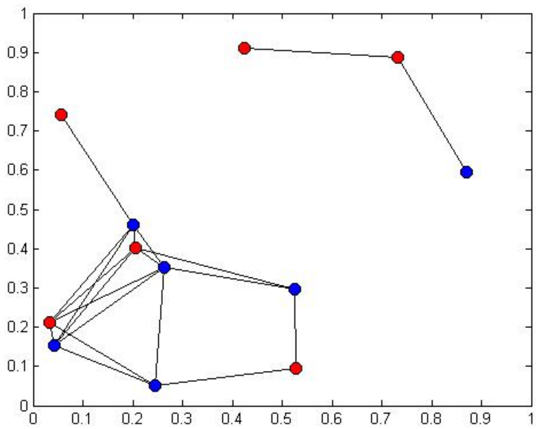

Figure 2.6: Network divided in two random subnets.

in two subnets formed by 10 nodes granting half frame to each subnet the aggregated throughput will be 0.8595 pk/slot, dividing in 4 we obtain 0.9668 pk/slot, in 10 subsets 0.9945. If we consider TDMA we obtain 1 pk/slot. These results show that if we consider a fully connected network the best performance in terms of throughput are obtained scheduling transmissions. Let me now consider a *partially connected network* in which there are hidden nodes. For example consider the network showed in fig. 2.5, formed by 12 nodes randomly distributed over an area of $1m^2$. In this network applying CSMA algorithm we obtain an aggregated throughput of 1.15 pk/slot. Dividing the network in 2 subnets randomly formed by 6 nodes as showed in fig. 2.6 ensuring half frame to each subnet, we obtain an aggregated throughput of 1.82 pk/slot. Dividing the network in 3 subnets randomly formed

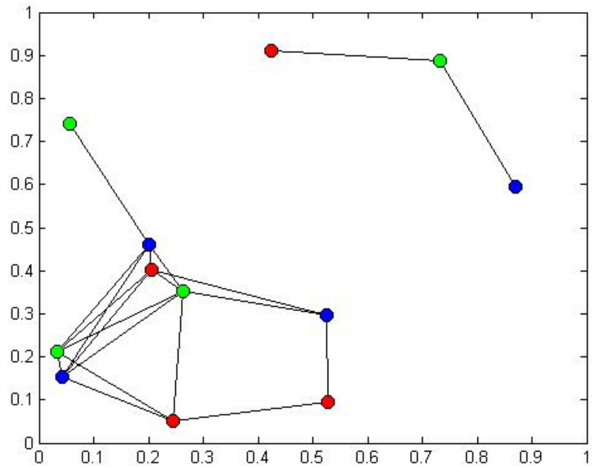

Figure 2.7: Network divided in three random subnets.

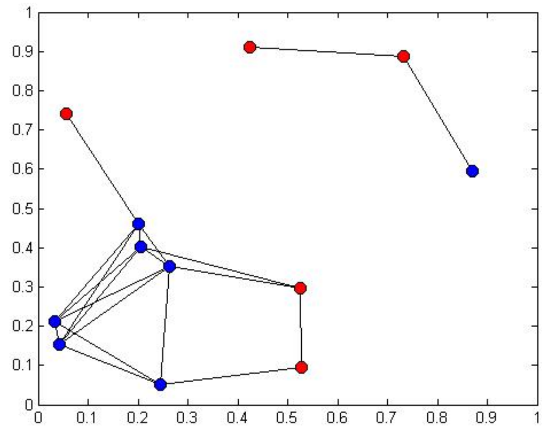

Figure 2.8: Network divided in two chosen subnets.

by 4 nodes as in fig. 2.7 we obtain 1.90 pk/slot. These results might lead to think that also in this case scheduling transmissions improve network's performance, but if we go on in increasing the number of subsets we note that aggregated throughput decrease. In table 2.4.4 are showed the values of

Number of subsets	**aggregated throughput [pk/slot]**
1	1.15
2	1.86
3	1.95
4	1.75
5	1.74
6	1.64
7	1.40
8	1.22
9	1.09
10	1.08
11	1.02
12	1

Table 2.2: Comparison between aggregated throughput dividing the network in subnets.

aggregated throughput dividing the network starting to 1 arriving to 12 subnets. Each value is obtained averaging ten random placement of the nodes in the subnets. The number of subsets in which we divide the nodes represents a different tradeoff between contention and scheduling. A small number of subsets will lead to an higher contention level in each subnet, while a great number of subnets does not allow spatial reuse. We would like to assign nodes that may collide to different subsets, so that they can access the channel in non-overlapping time intervals. Let we now consider the network in fig. 2.5 divided in 2 subnets *conveniently* formed as showed in fig. 2.8 and ensure half of frame to each subnet, in this case we reach an aggregated throughput of 2.1 pk/slot, so, maintaining the same number of subnets, but dividing nodes in a different way we have further improved the performance. This proof that for finding the best performance we have to choose both the number of subsets in which divide the network and in which way dividing nodes. If the partition is limited to two subnets only, being 10 the total number of network nodes, in principle we have 2^{10} possible combinations for organizing the node groups, which are reduced to one half of this number if we consider that throughput performance are invariant to the grouping order. In other words, the performance is the same if nodes 1 and 2 are in subset 1 and nodes 3 and 4 in subset 2 or if nodes 1 and 2 are in subset 2 and 3 and 4 in subset 1. For each possible combination over the total 2^9 possible ones, we can then solve an optimization algorithm in order to decide how many slots allocate to the first and second subnet in a frame.

We can repeat the same reasoning considering the network divided in three subnets e go on. An optimization algorithm that determines the optimal value of subsets in which we can divide the network is NP-Hard [90] [91], and increasing the number of subsets, full exploration becomes unthinkable because of the huge number of possible combinations.

For this reason, we have faced the problem according to an heuristic approach finding some logical rules for dividing the nodes in subnets. If for some nodes is not possible to decide the destination group because the proposed rules cannot be applied, we explore all possible decisions by considering as freedom degree only the decisions about the nodes for which the heuristic logic cannot decide. This allows to dramatically reduce the exploration space in comparison to the whole combination set.

Heuristic decision logic. We base our heuristic decision logic on the following considerations. A first rule to be applied in the subdivision of nodes is trying to put hidden nodes in different subsets, so that they do not interfere. If the number of subnets is not enough to separate all the hidden nodes, we decide the assignment of the pending nodes by considering all the possible decisions. For example, if we have to divide nodes in two subsets and a given node is hidden both to a node in the first group and a node in the second group, the node is considered as a freedom degree and all possible decisions (assignment to the first group or assignment to the second group) need to be explored by means of the throughput model described in section III-A. In general, if the total number of pending nodes is p, we need to consider 2^{p-1} possible combinations.

2.4.5 Frame Decision Optimization

Recalling the network partitions in fig. 2.8 we can underline that the partitions obtaining the best performance has subsets formed by different number of nodes. Partitions do not generally include the same number of nodes, and in generic traffic scenarios, we should also consider the offered load rather than the number of nodes. It follows that the optimal solution is not ensure the same number of slot to each subnet. Slot allocation should depend on traffic demand and throughput achieved in each subnet. In addition to the number of subsets in which we divide the network and to the grouping of nodes in these subnets, we have to decide how many time slots in the frame need to be allocated to each subset. In order to decide how to organize the frame, because of the size of the problem, it is not possible to explore all the combinations for each number of subsets. So, we implement the following optimization algorithm, that we run for each partition of the network.

2.4.6 Optimization problem

The optimization problem is aimed at solving the following issues: i) determine the best number N of subsets in which divide the network, ii) divide the nodes in subnets; iii) obtain the best number of time slots α_i to be granted to each subnet i in each frame.

We recall from eq. 2.4 that the throughput is related to the number N of slots and to the parameter α_i, then our problem can be solved with an algorithm that makes an optimal choice of N and α_i to improve the network throughput.

As possible solution we have considered the maximization of the throughput for each node achieving the worst performance in the network. In each subnet i the node presenting the minimum throughput x_i is chosen. As described in section 2.4.1, the throughput is a portion $\frac{\alpha_i}{T}$ of the throughput calculated with the CSMA algorithm, so we can maximize, varying α_i, the minimum comparing all $\alpha_i * x_i$. The problem becomes a multi-objective integer problem max-min to be solved as follows:

$$
\begin{aligned}
&\max \min_{\alpha_i}(\alpha_1 * x_1, \alpha_2 * x_2, ..., \alpha_n * x_n) \\
&s.t \sum_{i=1,2,..,n} \alpha_i \leq T \\
&\alpha_i \geq 0 \quad i = 1, 2, ..n \\
&\alpha_i \text{integer}
\end{aligned}
\tag{2.5}
$$

where T is the frame duration, and the sum of time slots α_i must be less or equal to T.

This optimization needs to be applied in all possible combinations in case of full exploration of the partitioning possibilities, as we have seen in the previous paragraph.

Because the solution of the above optimization problem which is multi-objective and integer is not viable, we recast the problem introducing another way to assign the frame to the nodes. In this approach we maximize the sum in the network of the minimum throughput of each subnet adding a fairness factor: we grant to each node a throughput grater or equal to a value that we choose as $\frac{1}{n}$ of the sum of minimum throughput. In this way we obtain the following optimization single-objective integer problem:

$$
\begin{aligned}
&\max_{\alpha_i} \sum_{i=1,2,..,n} \alpha_i * x_i \\
&s.t \sum_{i=1,2,..,n} \alpha_i \leq T \\
&0 \leq \alpha_i \leq T \quad i = 1, 2, ..n \\
&\alpha_i \text{integer} \\
&\alpha_i x_i > \frac{1}{n} \sum_{i=1,2,..,n} x_i
\end{aligned}
\tag{2.6}
$$

2.4.7 Numerical results

We have carried out simulations in random networks formed by 25 nodes situated in an area of 1 m^2 with a range of visibility between nodes varying from $0,4$ m to 1 m. Varying the radius of visibility we can see how the *hidden nodes* influence the network performance: a small radius corresponds to a lot of hidden nodes, while a radius equal to 1 corresponds to a completely connected network, without *hidden nodes*. Simulations are carried out with MATLAB software and GUROBI extension for the integer optimization algorithm. Each curve in figure 2.9 is the result of an average over ten simulations.

The algorithm applied for each number of subsets is described schematically as follows:

0 From the knowledge of G and $G^2 - G$, divide nodes in subsets without assigning hidden nodes to the same subset.

i. Check if we have nodes that we can not assign to a subset according to the above rule.

ii. In case there are still not assigned nodes, explore all possible combinations of these nodes in subsets.

iii. For each combination (if all nodes have been divided at the first step, there is only one combination) apply the optimization algorithm.

iv. Check if there exist nodes presenting a throughput equal to zero, and then try all possible combinations repeating from step 2.

v. Evaluate the best configuration solving an optimization problem for each number of subnets in the partition.

In figure 2.9 the simulation results are depicted. In y axis we represent the aggregated throughput, that is the sum of the node's throughput in a subnet. In the figure we represent the maximum value of aggregated throughput for each number of subsets in the partitions. In x axis we see the number of subsets in which the network is divided, starting from 1 subnet, that is the entire network, in which medium access is completely contended between nodes, up to 25 subnets, in which each subnet is formed by one node, corresponding to an entirely scheduled access. Dividing the network in subnets it can be shown that it is possible to obtain a throughput improvement with respect both to an entirely contention access and to an entirely scheduling access. Considering the curve with radius of visibility 0.4 m and dividing the network in 2 subsets we obtain an aggregated throughput of 2.6 pk/node/slot, while with CSMA we obtain 1.35 and with TDMA 0.12. The minimum throughput obtained dividing in 2 subnets is 0.0426, while with CSMA is 0.0063 and with TDMA is 0.04. The maximum throughput obtained dividing in 2 subnets is 0.2, while with CSMA is 0.1 and TDMA is 0.04. So we can see that dividing the network in two subnets we improve all these values.

In table 2.4.7 we show the comparison between the optimal values of throughput obtained with our algorithm and with the throughput obtained, respectively, with CSMA end TDMA for all the considered radius of visibility. These values are obtained as average of ten simulation with random topologies.

We note that the effectiveness of the hybrid CSMA/TDMA scheme improves as the visibility radius becomes smaller, because in these conditions the impact of hidden nodes is more relevant.

		thr aggr	
radius	**opt alg**	**CSMA**	**TDMA**
0.4	2,64	1,35	0,13
0,5	2,06	1,14	0,12
0,6	1,44	0,70	0,12
0,7	1,07	0,59	0,12
0,8	0,95	0,56	0,12
0,9	0,92	0,55	0,12
1	0,81	0,71	0,04

Table 2.3: Comparison between the obtained (max, min, aggregated) optimal values of throughput with the one obtained with, respectively, CSMA end TDMA changing of the radius of visibility.

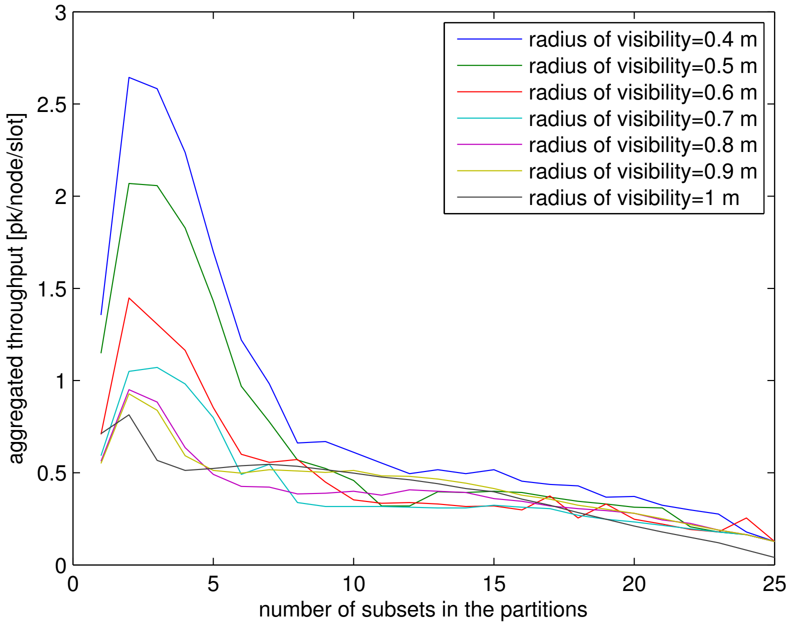

Figure 2.9: Aggregated throughput varying node's subset.

2.4.8 Conclusion

CSMA/CA shows poor performance in presence of hidden nodes. A form of coordination among nodes can improve the network performance. In this paper, we propose a novel algorithm presenting through an integer optimization algorithm an equilibrium between contention and scheduling for allocating optimally resources in multi-hop networks. The idea is to split nodes in subsets transmitting in different time slots. Nodes inside a subnet contend medium according CSMA algorithm. We have showed simulation results proving that this method improves the network performance in terms of throughput both respect to CSMA and TDMA algorithms. We are still working on finding better algorithm for dividing nodes in subnets in order to further improve network throughput.

Chapter 3

Frequency Planning

3.1 Introduction

The flexibility, low cost and ease of deployment offered by WLANs has been a major factor in their widespread deployment and popularity. For designing an efficient WLAN to optimize factors as access point (AP) locations, channel assignment and frequency planning is of primary importance. In this chapter we arrive to analyze the network performance at the physical layer, presenting a novel distributed algorithm for frequency planning. The mechanism proposed is based on a machine learning technique in which each node choose automatically and autonomously the channel. The algorithm does not need to exchange of informations between nodes, and it adapt the planning when conditions, such as topology change. The IEEE 802.11 (a/b/g) based networks operate in the unlicensed ISM band in the 2.4 GHz and 5 GHz frequencies. The 2.4 GHz band is divided into 11 channels with the channel number indicating the center frequency. For example, channel 1 is at 2.412 GHz. The basic 802.11 and the 802.11b extension use the Direct Sequence Spread Spectrum (DSSS) which takes about 22 MHz of bandwidth on each side of the center frequency. However, the channel center frequencies are spaced only 5 MHz apart. Thus, a single channel overlaps with up to 4 successive neighboring channels. As a result, the 2.4 GHz ISM band has only 3 non-overlapping channels, 1, 6 and 11. The overlap among adjacent channels is typically detrimental in nature. For example, a transmission on channel 1 will interfere with transmissions in the vicinity on an adjacent. In recent years the use of WLAN has always been growing, as well as the density of APs, so the availability of only three orthogonal channels has made frequency channel a crucial task. The problem of network planning and frequency reuse in cellular networks is typically modeled as a graph-coloring problem, where each graph vertex represents a different AP, each edge represents potential interferences, and the colors represent the number of non-overlapping channels. Goal of the planning is to cover all the APs in the network, with the minimum number of channels. However, in the case of multi-cellular WLAN, there are some peculiarities which need to be taken into account, especially for the 802.11b PHY layer, which makes available only three orthogonal channels. On one side the 802.11 access protocol is very robust to the interference, thanks to the carrier sense function which prevents the access to the channel whenever a source of interference is active. This means that the carrier sense function makes orthogonal two interfering channels, by operating a time division access, and stations in two interfering cells receive almost the same performance as they were utilizing the same channel. On the other side, in case of misplacing of the interfering cells, the same carrier sense function can very dramatically degrade the performance of some cells of the network. Using orthogonal channels can alleviating contention and interference leading to substantial performance improvement. However, this also gives rise to non-trivial channel coordination issues, and the variability in the achievable data-rates across channels and links make the plan more difficult. Currently, network administrators often manually decide on a static channel assignment for APs based on RF profiles [105]. However, traffic loads in a network tend to vary with time, and consequently, such assignments do not result in the best performance. Computing an optimal schedule, even in a single-channel network, is almost always intractable, due to the need for global information, as well as the computational complexity [92]. For mitigating interference in WLAN a mechanism that could be used is power control. 802.11 APs support power adaptation, each AP can tune the transmission power for communicating with its clients [94, 95, 96], but this approach may introduce throughput starvation due to asymmetric link. Further, although utility based power control methods produce optimal network throughputs, they require frequent message exchange between nodes and also easily produce severe unfairness to individual node's throughput [101] depending on the topological conditions of the network. Another approach is to enable routers with multiradio (MR) multichannel (MC) access. To enable the concurrent transmissions via multiple radios transmitting over orthogonal channels simultaneously, the key problem is the channel scheduling. The benefits of using multiple radios and channels have been theoretically studied in [97, 98, 99] in which is showed the achievable throughput in MR-MC networks. In [99] is proved that the complexity of general channel assignment problems is exponential in the number of wireless links. In [100], the authors propose a channel scheduling algorithm based on a network partitioning. They proposed an algorithm that identifies and protects against interference links that are most critical. This approach is simple and results in polynomial-time problem formulation, but all the links in a partition are fixed to a common channel, so this solution is not flexible and does not reach optimal throughput. Joint scheduling and power control problem has been also proposed [102, 104], but [102] demonstrated to be a NP-complete problem that requires a global knowledge on the network which makes it difficult be applied to ad-hoc networks that do not have a central coordinating node. Moreover the complexity of these algorithms grows exponentially with the number of nodes. [103] proposes an algorithm based on joint scheduling and power control problem employing a simple greedy algorithm that autonomously groups nodes into a number of subgroups for scheduling for a time division multiple access. This algorithm is fully distributed and requires limited message exchange between nodes, but does not obtain optimal results. In [106] is described a scheduling algorithm for MC MR wireless networks that requires information about per-channel queues at all interfering links. This provides a strong motivation for the study of scheduling algorithms that can operate with limited information.

For a small network in a limited area, only manufacturer's information on the coverage range is sufficient to deploy the APs. For a larger network, a more accurate deployment procedure is required to ensure sufficient coverage and network functionality (bit rate, capacity, interference, etc.). Basically there are two approaches. The first is based on a site survey with a lot of

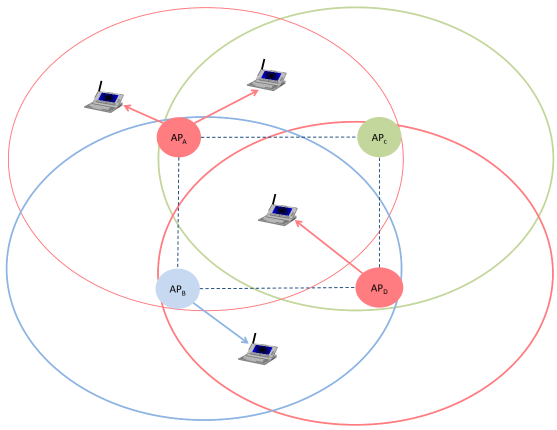

Figure 3.1: An example of frequency planning based on the interference observed by the Access Points.

measurements and experimental decisions. The second method comprises of software planning using propagation models [93].
In this chapter we present a novel distributed protocol based on a machine learning mechanism in which the nodes choose, automatically and autonomously in each time slot, the optimal channel for transmitting through a weighted combination of protocols. Respect to the solutions proposed in literature we obtain optimal results without adding any exchange of informations between nodes and following the network in changing conditions.

3.2 Dynamic Frequency Planning

3.2.1 Motivations

As discussed in the previous section, the problem of frequency planning in WLANs has some peculiarities because of the small number of available channels in the 2.4GHz bands and the high-density of Access Points (APs) usually deployed. In these conditions, it is very likely that multiple interfering cells are configured on the same channel. Although the carrier sense mechanism allows the coexistence of interfering cells, cell capacity can be unevenly allocated in different cells. Moreover, some stations can be significantly impaired by hidden node transmissions or *flow-in-the-middle* problems. These phenomena are difficult to predict, because the APs (which decide about the cell operating channel) can estimate the interference conditions suffered in the cell only locally, while interference can vary drastically for different receiver positions.

Consider for example the scenario depicted in figure 3.1. Four APs are deployed in the same area according to the visibility graph represented by the dashed links (e.g. AP_A can sense AP_B and AP_C transmissions, but not AP_D ones). Assuming that only three channels are available, it is possible that AP_A and AP_D select the same operating channel (e.g. the red one, while AP_B and AP_C are configured on the orthogonal channels blue and green). Under this hypothesis, the station associated to AP_D will experience a very low throughput, because AP_A is hidden to AP_D and its transmissions can collide with AP_D with high probability. Being AP_C cell empty, a better choice could have been to tune AP_D on the green channel. In case AP_C cell becomes congested, performance can be improved by moving AP_D on the blue channel. It follows that it is difficult to find the best possible planning by only considering the interference experienced by the APs. Moreover, since stations can move during their activity and traffic sessions activate/deactivate dynamically, it does not exist a *fixed* frequency planning for which the cell performance is always optimal.

Consider now another example in which four APs are all in radio visibility (e.g. in figure 3.1, AP_D can also sense AP_A, and AP_B can sense AP_C). Since in the 2.4GHz bands only three channels are available, any *fixed* planning will configure two cells on two independent channels and two cells on the same channel. This implies that the cells working on an exclusive channel can achieve the maximum channel capacity, while the cells sharing the same channel can achieve one half of the channel capacity. Resource allocation can be improved under time-varying channel assignments. For example, by periodically switching each AP to a different channel, it is possible to alternatively work on an exclusive channel or on a shared channel in order to allocate 3/4 of the channel capacity in each cell.

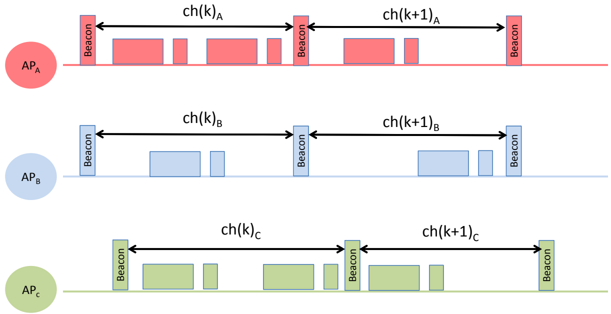

Figure 3.2: An example of dynamic (synchronous and asynchronous) channel switching performed at each beacon interval.

We conclude that a time-varying selection of the operating channel in each cell can be beneficial for optimizing the utilization of the wireless resources and for improving the fairness of resource allocation.

3.2.2 Supporting Channel Switching

Figure 3.2 shows an example of dynamic channel selection. Each AP notifies the decision about the channel switch in the beacon frame, by means of a special information element (IE). A similar information element, called channel switch announcement element, has been already included in the 802.11s extensions, for tuning all the nodes belonging to a mesh network on the same channel, and in the 802.11n extensions for moving from 20MHz to 40MHz channels or vice versa. In principle, a new operating channel can be selected at each beacon interval, while multiple beacon intervals can be considered for minimizing the overhead due to the time required for switching to a novel channel. In the figure, two APs transmit their beacon simultaneously, thus synchronizing their channel switches, while a third AP sends its beacon slightly later. Obviously, this second condition is more realistic, because beacon frames contend for the medium access in each cell and can experience random delays from the expected transmission time. Thus, even synchronizing the expected transmission times, the final transmissions are not generally synchronized. In the following, we will consider both the ideal case of synchronous transmissions and the realistic case of asynchronous transmissions among the cells.

3.3 Learning Scheme

Consider a network formed by n APs potentially interfering in transmissions with nodes associated with them. Our idea is to define a frequency plan through a distributed machine learning mechanism so that the interfering AP will transmit in orthogonal channel without interfere. The adopted mechanism is based on Meta-MAC-Protocol [107], that is a systematic and automatic method to dynamically combine any set of existing MAC protocols into a single higher layer. The mechanism proposed in Meta-MAC-Protocol is to decide if to transmit or not in a time-slot through a weighted combination of MAC protocols. After the transmission, the Meta-MAC evaluate the performance of each protocol discounting the weights of protocols that was wrong. Motivated by this idea, we implement the combination at Physical layer rather than for MAC access. We consider the combination of protocols that decide if transmit or not in a channel. We consider a time-slotted system with three orthogonal channel, for example channel 1, 6 and 11. In each slot every node will choose *automatically* and *autonomously* the optimal channel for transmitting. Each protocol says to transmit in a different channel. As showed in fig.3.3, every node attributes a weight to each protocol, the decision whether or not to transmit in a channel at a given time is reached by appropriately combining the three protocols. Initially each protocol has the same weight and each channel is a good candidate for the transmission; at the end of each slot each node will check the channel performance. If the transmission is not successful the node will penalize the channel for the next slot discounting the weight of the related protocol, decreasing the probability of choosing the same channel in the next slot. The algorithm is run locally at each node and it does not need information from other nodes. We consider two possible variations of the protocol:

- synchronous mode: time slots of nodes are contemporary, a synchronization mechanism is necessary.

- asynchronous mode: time slots of nodes are not contemporary. We assume that the difference between the begin of a slot of each couple of adjacent nodes is minor respect the duration of a time slot.

We consider three protocols:

$$p_1 = [1\ 0\ 0]'$$

$$p_2 = [0\ 1\ 0]'$$

$$p_3 = [0\ 0\ 1]'$$

The component equal to 1 says in which channel the protocol wants to transmit. The decision is computed as a function of the weighted average of the three protocols:

$$d(i) = \frac{\sum_k w(k,i) * p(:,k)}{\sum_k w(k,i)} \tag{3.1}$$

where w(k,i) is the weight of k component of the protocol at node i. Decide the channel: given a random number a the decision is:

- $dt(:,i) = [1\ 0\ 0]'\ if\ a \leq d(1);$
- $dt(:,i) = [0\ 1\ 0]'\ if\ a \leq d(1) + d(2)$
- $dt(:,i) = [0\ 0\ 1]'\ if\ a \leq d(1) + d(2) + d(3)$

At the end of each time slot we evaluate if the decision made in the time slot was right in order to update the weight. At this end we have to define when a decision is correct. We consider two versions of the protocol. In a first version we consider a decision right only if the node transmit without collisions. In a second version, CSMA/C compliant, we consider two cases of right decision: if the node made a successful transmission, or if the node have not transmitted because of carrier sense. Let y_t denote the feedback.

$$\begin{cases} y_t = 1 & \text{if the decision was right;} \\ y_t = 0 & \text{if the decision was wrong;} \end{cases} \tag{3.2}$$

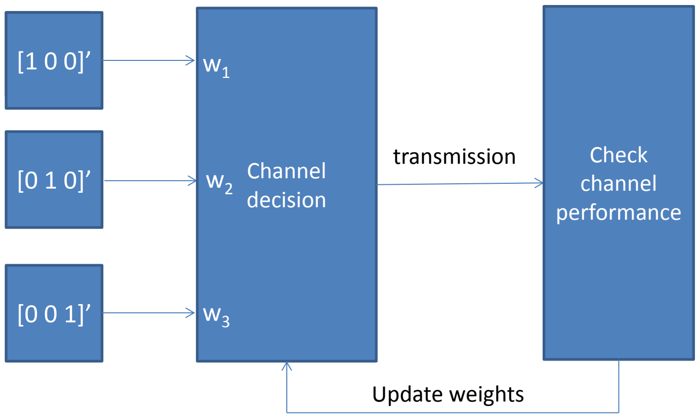

Figure 3.3: Operation of the algorithm.

If the protocol made the right decision: if the node transmitted successfully $z(i) = p(i)$ otherwise $z(i) = [0\ 0\ 0]'$ and update the weight: for each protocol k

$$\delta = \sum_c |z(c) - p(c, k)|$$

for each node i and each protocol k:

$$w(k, i) = w(k, i) * exp(-\eta * \delta)$$

η is a parameter that define the speed of the weight update. The term δ represent the deviation of the protocol k from the correct decision: if this deviation is zero, then the weight of the protocol remains unchanged, otherwise it decreases its weight such that with increasing errors the decrement grows. Due to the normalization in equation 3.1, after each slot, the relative weight of protocols that made the right decision will grows. In this way, the weights essentially reflect the credit history of the component protocols.

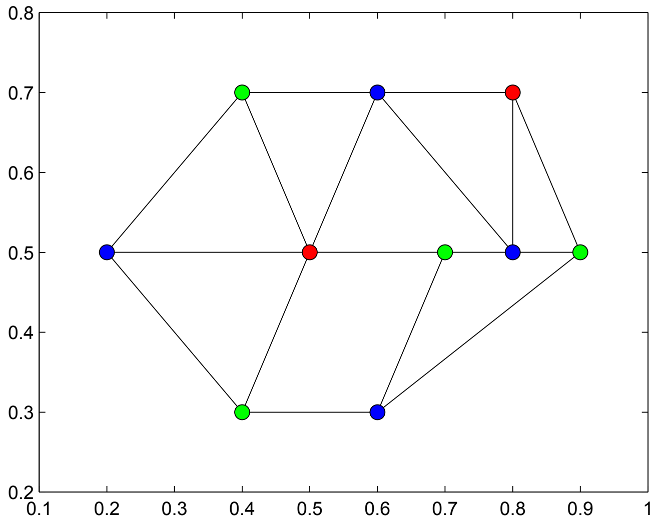

Figure 3.4: Frequency plan in synchronous protocol without Carrier Sense.

Because the weights decrease exponentially but never grows, if a protocol reach weight 0 it can not be choose in next time slots. If network conditions change this can become the right choice. So for maintaining all channel a possible choice (even if improbable) we set a minimum value below which no weight can drop. Renormalization does not change the relative sizes of the weights, and since they are only used in a normalized way in computing the combined value only their relative size matter.

We now summarize the steps of the protocol:

- Initialization: set all weights to 1;
- At the beginning of the slot each node i compute the component decision
$$d(i) = \sum_k w(k,i) * p(:,k)$$
- At the end of the slot (that is at the same time at each node) the nodes evaluate if the protocol made the right decision $z(i)$: if the node transmitted successfully $z(i) = p(i)$ otherwise $z(i) = [0\ 0\ 0]'$ and update

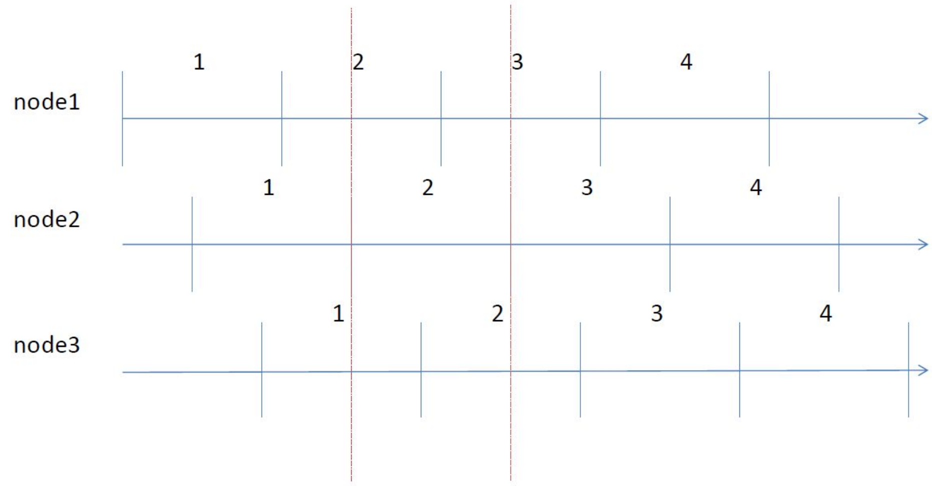

Figure 3.5: time slots in asynchronous mode.

the weight: for each protocol k

$$\delta = \sum_{c} |z(c) - p(c,k)|$$

for each node i and each protocol k:

$$w(k,i) = w(k,i) * exp(-\eta * \delta)$$

For the proof of the convergence of the algorithm we refer to the proof in [107] that is fully applicable also in our version. A key, and not trivial, point in this algorithm is to decide when a protocol made the right decision. In all versions if the node transmits successfully the decision is considered right. While in $p1$ and $p3$ all other decisions are wrong, in $p2$ and $p4$, CSMA compliant, if carrier sense detects that a node in its range of visibility is transmitting in the same channel chosen by node i it decides not to transmit: $dt(:,i) = [0\ 0\ 0]'$. When the node evaluate if the protocol made the right decision $z(i)$ if the node decided to not transmit because of carrier sense the decision is considered right $z(i) = p(i)$.

Another problem that we faced is the synchronization between nodes: if nodes are not synchronized, time slots in different nodes present not synchronized initial and end times instant, and the probability of collisions increases. As we can see in fig. 3.5 in an asynchronous environment, provided node 1 has chosen the frequency channel, node 2 in its time slot 2, can have collisions in both time slot 2 and time slot 3 of node 1, depending on the frequency channel it just decided. Moreover, collisions with node 3 can still happen, depending on the decision of the node 3 itself in its time slot 1 and 2. Generalizing at time slot t node i we have to check possible collisions between:

$d(i, t)$ and:

- $d(j, t)$
- $d(j, t+1) \quad \text{when} \quad j < i$
- $d(j, t-1) \quad \text{when} \quad j > i$

We consider four versions of the protocol:

- $p1$: Syncronous without Carrier sense
- $p2$: Syncronous with Carrier sense
- $p3$: Asyncronous without Carrier sense
- $p4$: Asyncronous with Carrier sense

In the next section we show numerical results obtained applying the four versions of the protocol.

3.4 Numerical Results

In this section we show results in a network formed by ten APs. Simulations are carried out in Matlab. We have considered various networks formed by several nodes starting from 3 up to 30 nodes in different situations. The algorithm in all cases reached the optimal planning. Consider the matrices

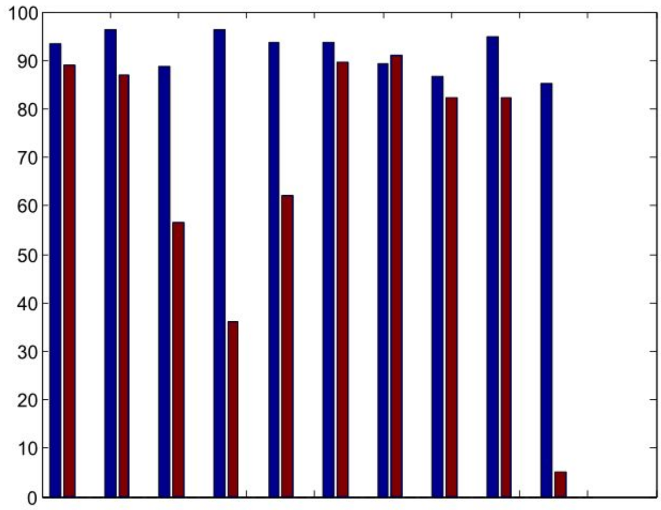

Figure 3.6: Number of packets successfully transmitted in 300 TS from each node comparing results obtained keeping in count or not traffic load in synchronous mode.

G representing interfering conditions:
G=

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0.2 & 0 & 0.1 \\ 0 & 0 & 1 & 0 & 0.6 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.6 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0.7 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0.8 & 1 & 0.2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0.1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Each term G_{ij} represents the probability of interference of transmissions between AP i and AP j and their respective stations. The probabilities showed in G keep in count network load. Clearly we need this matrices only for modeling the collisions that occur in the network. In a real implementation we do not need of information about network topology or interference, in that case we only check if collisions occur.

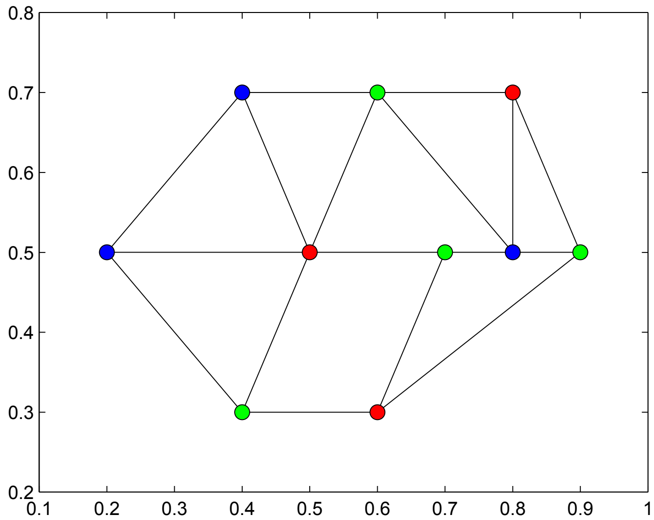

Figure 3.7: Frequency plan in synchronous protocol with Carrier Sense.

We now show results for the four protocols analyzed. Initially nodes choose randomly the channel, passing time the probability of choosing the optimal channel grows, and after some time slots, maintaining network condition stable, each node choose the same channel in all time slots. If the network conditions change for example because of a mobility network, or the switch on of a new AP, the weights will change and the protocol will find a new regime. The parameter η influences how many fast weights will change. Considering $\eta = 0.25$ we reach convergence in about 50 steps in synchronous mode and about 150 steps in asynchronous mode. The number o steps for reaching the convergence does not increase with the network size, and in our experiments maintain these values also in networks formed by 30 nodes. In asynchronous mode we have a longer transitory because as we said in section 3.3 in this case we have a grater number of possible collisions, and to find the optimal solution is more difficult. In fig. 3.4 is showed the frequency plan obtained applying $p1$ protocol in regime's condition. The three different colors represent the three different orthogonal channels. We can see that potentially interfering transmissions are scheduled in different channels. In fig. 3.6 and 3.8 are showed the node's throughput, respectively, in synchronous and asyn-

chronous mode. In blue we represent node's throughput obtained applying the algorithm considering network load, in red considering only potentially interferences. It is evident from numerical example that the applications of the algorithm that considers the network load can considerably improve the performance.
In fig. 3.7 the resulting frequency plan, obtained applying $p3$, has been depicted. We consider a new matrices CS to keep in count the carrier sense. In this example we have considered nodes 1 and 2, and 3 and 4 in radio visibility. In this case we see that nodes 1 and 2 have chosen the same channel, and this could appear an error, because according to G they could interfere, but they are also in radio visibility, so carrier sense could manage transmissions between them. Comparing synchronous and asynchronous mode we note that the first protocol gives better performance in terms of convergence rate and obtained per-node throughput. This result was predictable, as we know that in literature every synchronized algorithm gives better performance respect to its asynchronous version, but in a real environment often is difficult, or not possible, synchronize the nodes, and our asynchronous version still provides good performance. In the next section we show the results obtained applying in a real testbed the results obtained with our algorithm.

3.5 Experiments in a real testbed

3.5.1 Experiment Methodology

For further validate our algorithm we have applied in a real testbed the results obtained with our algorithm. We have considered a network formed by 6 AP with a station associated to each AP. We firstly have evaluate the matrices of interferences at each station, that is a matrices $6 * 6$ in which each row represents a station and each column the interference given by each AP. The value of the cross between a station with its AP is set to 0. We apply $p2$ algorithm for obtained results CSMA compliant. In the testbed we only apply the frequency plan obtained with the algorithm, so at this step using synchronous or asynchronous mode is indifferent. In our experiment scenario

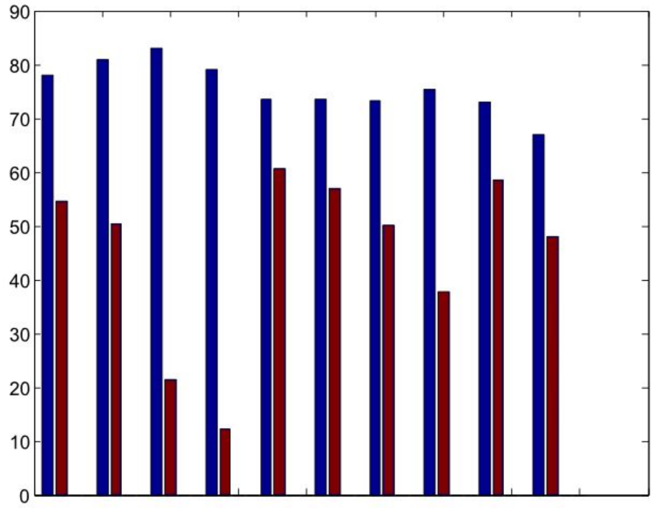

Figure 3.8: Number of packets successfully transmitted in 300 TS from each node comparing results obtained keeping in count or not traffic load in asynchronous mode.

we have designed a multi-AP network were the position/transmission power of each node has been chosen to provide the desired connectivity map. The experiments consists of a saturation at fixed unicast 802.11 data transmission (6 Mbps of modulation rate and 1470B of data payload size). We compared three different channel planning:

- no channel planning, each AP works on the same channel;
- a reasonable channel planning without using the proposed algorithm;
- the channel planning of a converged network that uses our algorithm.

in fig 3.9 the connectivity map of APs is depicted. Each AP is a iMinds lab embedded GNU/Linux Systems equipped with atheros AR900x IEEE 802.11 a/b/g/n COTS NICS. REACT algorithm [78] that is a distributed algorithm for dynamical resource allocation, has been implemented on NIC O.S. drivers using several statistics metrics given by the hardware such as receiving frame counts (RTS,CTS, Data-frame, ACK,...) and busy-time estimation for modeling the network's conditions.

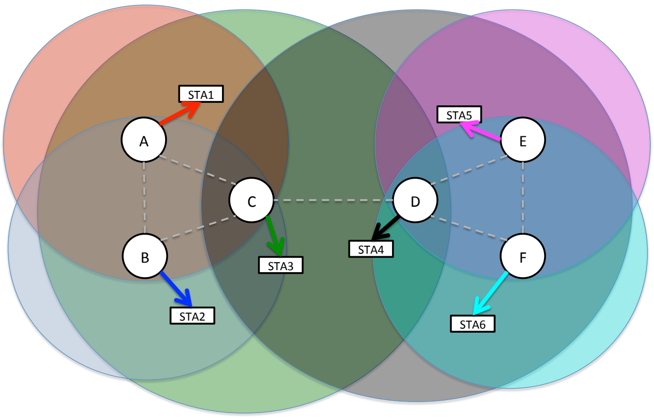

Figure 3.9: Network Topology.

3.5.2 Results

Applying the channel balancing algorithm in a static scenario we obtain a steady state situation where each AP has a fixed channel assignment of three orthogonal channels. In our experiments we denote with ch1, ch2, ch3 these channels.

In the first experiment, no channel planning is provided, all APs transmit

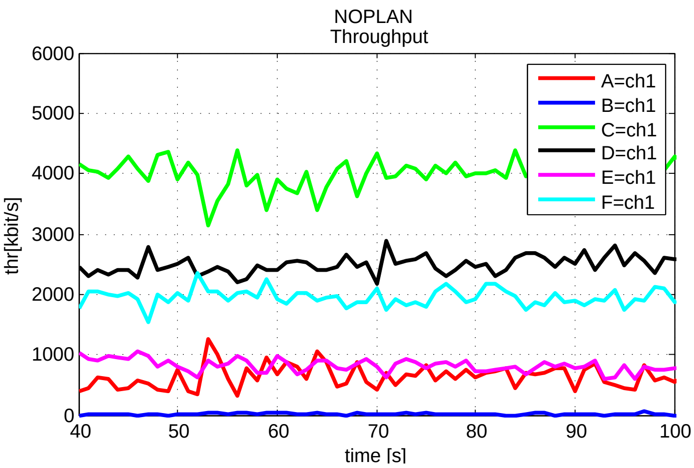

Figure 3.10: Standard DCF transmission.

on the same channel. In fig.3.10 we depict the behavior of a standard transmission throughput. The different performances depend on the position of AP in relation with the position of the stations. All nodes present different throughputs, there is an AP (labeled in the figure as B) that can not transmit and the node that achieves the maximum throughput reaches about 4 Mbit/sec.

In fig.3.11 a reasonable non-automated planning is provided. We can see

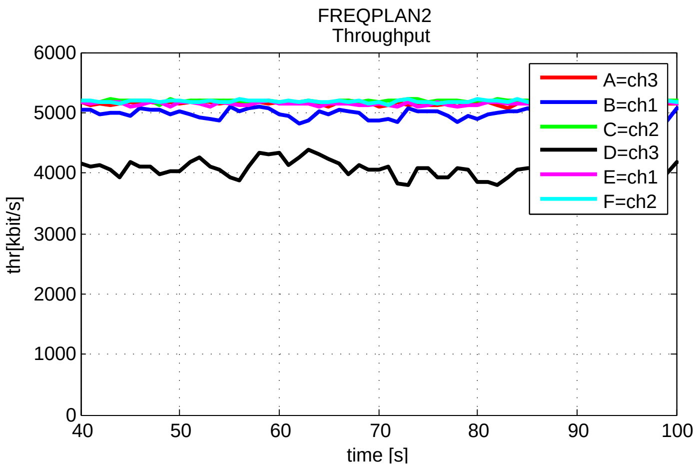

Figure 3.11: Arbitrary Frequency planning.

that 5 APs reach about 5 Mbit/sec, but an AP has a throughput of about 4 Mbit/sec (that is the minimum in this scenario but greater than the maximum in the precedent scenario).

In fig.3.12 we show the results obtained applying the planning obtained with our algorithm. It has been showed that not only we have a fair situation in which all nodes achieve, more or less, the same throughput, but they reach the maximum value obtained applying the precedent non-automated planning.

3.6 Conclusions

In this chapter we have presented a novel protocol based on a distributed machine learning mechanism in which the nodes choose, automatically and

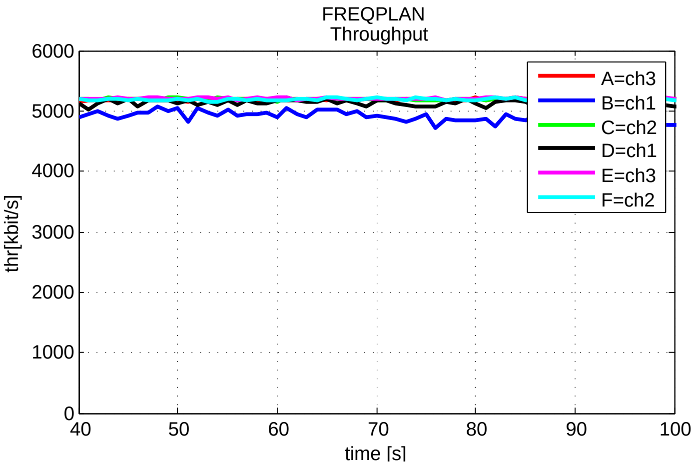

Figure 3.12: Frequency Planning.

autonomously in each time slot, the optimal channel for transmitting through a weighted combination of protocols. The proposed algorithm could works both in synchronous and asynchronous mode. We have showed numerical and experimental results that have both proven the algorithm's effectiveness. While in the actual experiments we have applied the results of the planning, in future works we will implement the algorithm directly in the nodes of the testbed. We expect that results further improve, because an 802.11 APs implementing CSMA/CA knows if a transmitted packet has been successfully transmitted, while for update of the weights in machine learning mechanism we evaluate if a transmission was successfully through a mathematical model of the real environment, that obviously is only an approximation of the real conditions.

Conclusions

Motivated by the increasing importance of WLAN, in this thesis we have approached some performance optimization in WLAN at different layer of the OSI model. At *Network layer*, starting from a Hybrid System modeling the flow of traffic in the network, we propose a Hybrid Linear Varying Parameter algorithm to identify the link quality. To validate the model, numerical results have been presented. In future works, we will implement an on-line version to evaluate the link quality, that could be used as a metric in routing algorithms. At *Data Link*, we have presented two algorithms for MAC access based on a combination between TDMA and CSMA. We divide the nodes in subnets, scheduling transmissions of subnets in different time slots, and performing CSMA behind a subnet. In the first solution a game theoretical study of intra slot contention is introduced, in the second solution we apply an optimization algorithm to find the optimal degree between contention and scheduling. Numerical results show that both the solutions presented improve network performance with respect to both total contented and total scheduled approaches. Finally we analyze the network performance at *Physical* Layer. We have presented a novel protocol based on a distributed machine learning mechanism in which the nodes choose, automatically and autonomously in each time slot, the optimal channel for transmitting through a weighted combination of protocols. We have provided numerical results showing the optimality of our algorithm. We have presented also experimental results obtained applying the offline evaluated planning and we have reached better performance with respect to both pure DCF and a reasonable planning evaluated without the algorithm. In future work we will implement the algorithm in a real testbed.

Bibliography

[1] P. Demestichas, G. Dimitrakopoulos, J. Luo, R. Agusti, E. Mohyledinand, O. Sallent, D. Grandblaise, R. Pintenet, P. Leaves, and K. Moessner. Radio resource management and network planning in a reconfigurability context. In Proc. of IST Mobile and Wireless Summit 2004, June 2004.

[2] J. Mitola. The software radio architecture. IEEE Communications Magazine, 33(5):2638, May 1995.

[3] The GNURadio Software Radio, http://gnuradio.org/trac

[4] USRP. The universal software radio peripheral, http://www.ettus.com

[5] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, G. M. Voelker, Sora: High Performance Software Radio Using General Purpose Multi-core Processors, NSDI 2009.

[6] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, "Wireless MAC Processors: Programming MAC Protocols on Commodity Hardware" IEEE INFOCOM, March 2012.

[7] R. Draves, J. Padhye, and B. Zill, Comparisons of Routing Metrics for Static Multi-Hop Wireless Networks, ACM Annual Conf. Special Interest Group on Data Communication (SIGCOMM), Aug. 2004, pp. 13344.

[8] R. Draves, J. Padhye, and B. Zill, Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, ACM Annual Intl. Conf. Mobile Comp. and Net. (MOBICOM), 2004, pp. 11428.

[9] E. M. Belding-Royer, Multi-level Hierarchies for Scalable ad hoc Rout-

ing, ACM/Kluwer Wireless Networks (WINET), vol. 9, no. 5, Sept. 2003, pp. 46178.

[10] Akyildiz, I.F.; Xudong Wang, "A survey on wireless mesh networks," Communications Magazine, IEEE , vol.43, no.9, pp.S23,S30, Sept. 2005 doi: 10.1109/MCOM.2005.1509968

[11] H. Frey, Scalable Geographic Routing Algorithms for Wireless Ad Hoc Networks, IEEE Network Mag., July/Aug. 2004, pp. 1822.

[12] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximal throughput in multihop radio networks. IEEE Transactions on Automatic Control, 37(12):19361948, December 1992.

[13] A. Dimakis and J. Walrand. Sufficient conditions for stability of longestqueue- first scheduling: Second-order properties using fluid limits. Advances in Applied Probabilities, 38(2):505521, 2006.

[14] C. Joo, X. Lin, and N. B. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. In Proceedings of IEEE INFOCOM, April 2008.

[15] G. Zussman, A. Brzezinski, and E. Modiano. Multihop local pooling for distributed throughput maximization in wireless networks. In Proceedings of IEEE INFOCOM, April 2008.

[16] M. Leconte, J. Ni, and R. Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. In Proceedings of ACM MOBIHOC, May 2009.

[17] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications, 18(3):535547, 2000.

[18] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. In Proceedings 46th Annual Allerton Conference on Communication, Control and Computing, September 2008.

[19] Jian Ni; Bo Tan; Srikant, R., "Q-CSMA: Queue-Length Based

CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," INFOCOM, 2010 Proceedings IEEE , vol., no., pp.1,5, 14-19 March 2010

[20] K.-K. Lam, C.-K. Chau, M. Chen, and S.-C. Liew, Mixing Time and Temporal Starvation of General CSMA Networks with Multiple Frequency Agility, In Proceedings of IEEE ISIT, 2012.

[21] P.-K. Huang and X. Lin, Improving the Delay Performance of CSMA Algorithms: A Virtual Multi-Channel Approach, In Proceedings of IEEE INFOCOM, 2013.

[22] K. Aardal, S. Van Hoesel, A. M. C. A. Koster, C. Mannino, and A.Sassano. Models and solution techniques for frequency assignment problems. 4OR: A Quarterly Journal of Operations Research, 1(4):261317, Dec. 2003.

[23] P. Bjorklund, P. Varbrand, and D. Yuan. Optimal frequency planning in mobile networks with frequency hopping. Computers and Operations Research, 32:169186, Jan. 2005.

[24] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: Questions and answers. In Proc. of Military Communications Conference (MILCOM 2006), Oct. 2006.

[25] A. Eisenblatter. Frequency assignment in GSM networks: Models, heuristics, and lower bounds. PhD thesis, Technische Universitat Berlin, Berlin, Germany, 2001.

[26] A. Eisenblatter, M. Grotschel, and A. M. C. A. Koster. Frequency planning and ramifications of coloring. Discussiones Mathematicae Graph Theory, 22(1):5188, 2002.

[27] W. K. Hale. Frequency assignment: Theory and applications. In Proc. of the IEEE, volume 68, pages 14971514, Dec. 1980.

[28] A. M. C. A. Koster. Frequency Assignment Models and Algorithms. PhD thesis, Maastricht University, 1999.

[29] M. Johansson and L. Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. IEEE Transactions on Wire-

less Communications, 5(2):435445, Feb. 2006.

[30] B. Johansson, P. Soldati, and M. Johansson. Mathematical decomposition techniques for distributed cross-layer optimization of data networks. IEEE Journal on Selected Areas in Communications, 24(8):15351547, Aug. 2006.

[31] J. E. Mitchell. Handbook of Applied Optimization, chapter Branch-and-Cut Algorithms for Combinatorial Optimization Problems, pages 6577. Oxford University Press, 2002.

[32] M. Naghshineh and I. Katzela. Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. IEEE Personal Communications, 3:10 31, June 1996.

[33] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. IEEE Journal on Selected Areas in Communications, 24(8):14521463, Aug.2006.

[34] I. Siomina,Radio Network Planning and Resource Optimization: Mathematical Models and Algorithms for UMTS, WLANs, and Ad Hoc Networks, PHD thesis, Linkoping University, Sweden 2007

[35] D. Lee, D. Yun, J. Shin,S. Yun and Y. Yi, Provable Per-Link Delay-Optimal CSMA for General Wireless Network Topology, Proceedings of IEEE INFOCOM, 2014.

[36] J. Shamma and M. Athans, "Guaranteed properties of gain scheduled control for linear parameter varying plants," *Automatica*, vol. 27, pp. 559–564, 1991.

[37] M. Lovera, M. Verhaegen, and C. Chou, "State-Space Identification of MIMO Linear Parameter Varying Models," in *Proceedings of Mathematical Theory of Networks and Systems*, Padua, Italy, 1998, pp. 839–842.

[38] M. Nemani, R. Ravikanth, and B. Bamieh, "Identification of Linear Parametrically Varying Systems," *Proceedings of the 34'th IEEE Control and Decision Conference*, pp. 2990–2995, 1995.

[39] L. H. Lee and K. Poolla, "Identification of Linear Parameter-Varying

Systems using Nonlinear Programming," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 121, 1996, pp. 71–78.

[40] B. Bamieh and L. Giarré, "Identification for linear parameter varying models," in *Proceedings of the 38th IEEE Control and Decision Conference*, 1999, pp. 1505–1510.

[41] F. Previdi and M. Lovera, "Identification of a class of linear models with nonlinearly varying parameters," in *Proceedings of the V European Control Conference*, Bretthauer Karlsruhe, Germany, 1999.

[42] J. Shamma, "Linearization and Gain scheduling," in *Control Handbook*, W. Levine, Ed. CRC Press, 1996, pp. 388–396.

[43] W. Rugh and J. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[44] P. Lopes dos Santos, T. Azevedo Perdicolis, N. C., J. Ramos, and D. Rivera, *LINEAR PARAMETER-VARYING SYSTEM IDENTIFICATION New Developments and Trends*. World Scientific, 2011.

[45] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[46] A. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. H. G. Niessen, "Comparison of four procedures for the identification of hybrid systems," in *Proceedings of the Int. Conf. on Hybrid Systems: Computation and Control*. Zurich, Switzerland: Springer, 2005, vol. 3414, pp. 354–369.

[47] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[48] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: a tutorial," *European Journal of Control*, vol. 13, pp. 242–260, 2007.

[49] A. Juloski, S. Paoletti, and J. Roll, *Current trends in nonlinear systems and control*. Birkkäuser, 2006, ch. Recent techniques for the

identification of piecewise affine and hybrid systems, pp. 77–97.

[50] M. Tanelli, D. Adagna, and M. Lovera, " Identication of LPV state-space models for automatic web servi e systems," *IEEE Transactions on Control System Technology*, 19:93-103, 2011.

[51] J. Lee, S. Bohacek, J. P. Hespanha, and K. Obraczka., "Modeling communication networks with hybrid systems," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 630–643, 2007.

[52] B. Bamieh and L. Giarré, "Identification of Linear Parameter Varying Models," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 841–853, 2002.

[53] R. Tóth, "Modeling and Identification of Linear Parameter-Varying systems," Lecture Notes in Control and Information Sciences, 403, Berlin Heidelberg, 2010.

[54] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, 1981.

[55] S. Paoletti, A. Garulli, J. Roll, and A. Vicino, "A necessary and sufficient condition for input-output realization of switched affine state space models," in *Proceedings of the 47th IEEE Control and Decision Conference*, 2008, pp. 935–940.

[56] S. Paoletti, J. Roll, A. Garulli, and A. Vicino, "On the input-output representation of piecewise affine state space models," *Automatica*, vol. 55, no. 1, pp. 60–73, 2010.

[57] R. Tóth, F. Felici, P. S. C. Heuberger, and P. M. J. V. den Hof, "Discrete time LPV I/O and state space representations, difference of behavior and pitfalls in interpolation," in *Proceedings of the European Control Conference*, Kos, Greece, 2007, pp. 5418–5425.

[58] H. Abbas, R. Tóth, and H. Werner, "State-space realization of LPV input-output models: practical methods for the user," in *Proceedings of the American Control Conference*, 2010, pp. 3883–3888.

[59] Casella, F. and Lovera, M., " LPV/LFT modelling and identification:

overview, synergies and a case study", in *Proceedings of IEEE Int. Symposium on Computer-Aided Control System Design, Part IEEE Multi-conference on Systems and Control*,2008, pp. 852–857.

[60] Lovera, M., "LPV model identification: overview and perspectives", ftp://ftp.elet.polimi.it/users/Marco.Lovera/PRES/GTIdentMosar.pdf, 2008, available on line.

[61] P. Gupta and P. R. Kumar, The Capacity of Wireless Networks, IEEE Trans. Inform. Theory, 46(2), pp. 388-404, March 2000

[62] A. El Gamal, J. Mammen, B. Prakbkar, D. Shah. Optimal Throughput-Delay Scaling in Wireless Networks Part I: The Fluid Model, IEEE/ACM Transactions on Networking (TON) - Special issue on networking and information theory archive Volume 14 Issue SI, 2568–2592, June 2006.

[63] R. Rozovsky and P. R. Kumar. SEEDEX: A MAC protocol for ad hoc networks. Proceedings of The ACM Symposium on Mobile Ad Hoc Networking & Computing, pp. 67-75, MobiHoc 2001, Long Beach, Oct 4-5, 2001.

[64] K. Oikonomou, I. Stavrakakis. Analysis of Topology-Unaware TDMA MAC Policies for Ad-Hoc Networks Under Diverse Traffic Loads. Mobile Computing and Communications Review, Volume 9, Number 4, 25–38, 2005.

[65] J. Deng, P.K. Varshney, and J. Haas, A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function, Proc. Comm. Networks and Distributed Systems Modeling and Simulation Conf. (CNDS), 2004

[66] L. Giarrè, P. Falugi, R. Badalamenti, "Hybrid LPV Modeling and Identification", in "LINEAR PARAMETER-VARYING SYSTEM IDENTIFICATION New Developments and Trends", P. dos Santos, T. Perdicolis, C. Novara, J. Ramos and D. Rivera, World Scientific, 2011

[67] " Tinnirello, L. Giarrè, R. Badalamenti, F.G. La Rosa, "Utility-Based Resource Allocations in Multi-Hop Wireless Networks", NeTGCooP 2011

[68] " R. Badalamenti, P. Falugi, L. Giarrè, "HLPV Identication in wireless ad hoc networks", Bollettino di Matematica Pura e Applicata,2012

[69] R.Badalamenti, L. Giarrè, I.Tinnirello, Optimal Resource Allocation in Multi-Hop Networks: Contention vs. Scheduling, MED14

[70] P. Marbach and A. Eryilmaz, A backlog-based CSMA mechanism to achieve fairness and throughput-optimality in multihop wireless networks, in Proc. of the Allerton Conference, Monticello, IL, 2008.

[71] J. Ni and R. Srikant, Distributed CSMA/CA algorithms for achieving maximum throughput in wireless networks, in Proc. of the ITA Workshop, San Diego, CA, 2009.

[72] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, Towards utility-optimal random access without message passing, Wiley Journal of Wireless Communications and Mobile Computing, vol. 10, no. 1, pp. 115-128, Jan. 2010.

[73] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. W. Knightly, and M. Chiang, Experimental evaluation of Optimal CSMA, in Proc. of IEEE INFOCOM, Shanghai, China, April 2011

[74] B. Nardelli and E.W. Knightly, *Robustness and Optimality in CSMA Wireless Networks*, Rice University Technical Report, TX, July 2013.

[75] L. Giarré, F.G. La Rosa, R. Pesenti, I. Tinnirello. Coloring-based Resource Allocations in Ad-hoc Wireless Networks. *MedHoc 2011*, Favignana, 2011

[76] I. Chlamtac and S. S. Pinter. Distributed nodes organization algorithm for channel access in a multihop dynamic radio network. IEEE Transactions on Computers, C-36(6):728737, June 1987

[77] C. Zhu and M. S. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. Wireless Networks, 7(4):371384, 2001

[78] Lutz, J.; Colbourn, C.J.; Syrotiuk, V.R., "ATLAS: Adaptive Topology- and Load-Aware Scheduling," Mobile Computing, IEEE Transactions on , vol.13, no.10, pp.2255,2268, Oct. 2014

[79] Lutz, J.; Colbourn, C.J.; Syrotiuk, V.R., "Topological Persistence for

Medium Access Control," Mobile Computing, IEEE Transactions on , vol.12, no.8, pp.1598,1612, Aug. 2013

[80] I. Tinnirello, L. Giarré, R. Pesenti. Decentralized Synchronization for Zigbee wireless sensor networks in Multi-Hop Topology.Proceeding of IFAC NECSYS, Annency, Francre, 257-252,2010.

[81] I. Tinnirello, L. Scalia, F. Campoccia. Improving IEEE 802.11 performance in chain topologies through distributed polling and network coding. Proceedings of the 2009 IEEE international conference on Communications Dresden, Germany, 5392-5397, 2009

[82] I. Tinnirello, L. Giarré. A game theoretical analysis of contention in multihop networks. Technical report, November 2011.

[83] R.L.Brooks On coloring the nodes of a network, 37, 194-97, 1941.

[84] T. Calamoneri. The $L(h, k)$-labelling Problem: an annotated bibliography. *The computer journal*, 49, 5, 585-608, 2006

[85] Luby, M. Removing randomness in parallel without processor penality. Journal of Computer and System Sciences, 47(2), 250-286, 1993.

[86] Johansson, O. Simple distributed $\Delta+1$-coloring of graphs. Information Processing Letters, 70, 229-232, 1999.

[87] I.Finocchi, A.Panconesi, R.Silvestri. An experimental analysis of simple, distributed vertex coloring algorithms. Algorithmica, 41(1), 1-23, 2004. Preliminary version in ACM-SIAM SODA'02.

[88] I. Tinnirello, L. Giarré, G. Neglia. MAC Design for WiFi Infrastructure Networks: A Game-Theoretic Approach. IEEE Transactions on Wireless Communications, Volume 10, Issue 8, August 2011, Pages 2510-2522.

[89] I.Tinnirello, P.Cassará, G. Di Bella. "Performance Analysis in Spatially Correlated IEEE 802.11 Networks", *ICTC*, Jeju Island, Korea 2012.

[90] P. Brucker. "On the complexity of clustering problems". In: R. Henn, B. Korte, W. Oettli, "Optimization and operations research. Lecture notes in economics and mathematical systems", vol 157. Springer, New York, pp 454, 1978.

[91] RM. Karp. Reducibility among combinatorial problems. In: RE Miller, JW. Thatcher, "Complexity of computer computations." Plenum, New York, pp 8504, 1972

[92] V. Bhandari and N. H. Vaidya Scheduling in multichannel wireless networks, Proc. Distrib. Comput. Netw., pp.6 -17 2010

[93] S. Zvanovec, P. Pechac and M. Klepal, Wireless LAN Networks Design: Site Survey or Propagation Modeling?, Radioengineering, Vol. 12, No. 4, p. 42-49, Dec 2003. ISSN 1210-2512

[94] G. J. Foschini, Z. Miljanic, A simple distributed autonomous power control algorithm and its convergence, IEEE Trans. Veh. Tech., vol. 42, pp. 641-646, Apr. 1993.

[95] A. Akella, G. Judd, S. Seshan and P. Steenkiste, Self Management in Chaotic Wireless Deployments, in Proc. of ACM Mobicom, Cologne, Germany, Aug. 2005.

[96] Mhatre, V.P., Papagiannaki, K., Baccelli, F., Interference Mitigation Through Power Control in High Density 802.11 WLANs, INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , vol., no., pp.535,543, 6-12 May 2007

[97] P. Kyasanur and N. H. Vaidya, Capacity of multi-channel wireless networks:Impact of number of channels and interfaces, in Proc. ACM MobiCom, 2005, pp. 43-57.

[98] M. Kodialam and T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in Proc. ACM Mobi- Com, 2005, pp. 73-87.

[99] W. Xie, Y. J. Zhang, M. L. Sichitu, L. Fu, and Y. Yao, Feasibility of optimally assigning channels by exhaustive search in commercial multi-radio wireless mesh networks, Telecommun. Syst., vol. 44, no. 1/2, pp. 171 178, Jun. 2010.

[100] S. Avallone and I. F. Akyildiz, A channel assignment algorithm for multiradio wireless mesh networks, in Proc. ICCCN, 2007, pp. 1034 1039.

[101] Bozidar Radunovic and Jean-Yves Le Boudec. Joint Scheduling, Power Control and Routing in Symmetric, One-dimensional, Multihop Wireless Networks. In WiOpt03: Modeling and Optimization in Mobile,Ad Hoc and Wireless Networks, 2003.

[102] Min Cao, V. Raghunathan, S. Hanly, V. Sharma, and P.R. Kumar. Power control and transmission scheduling for network utility maximization in wireless networks. In Proc. of Decision and Control, Dec. 2007.

[103] Sangho Oh, Gruteser, M., Jiang, D., Qi Chen, Joint power control and scheduling algorithm for Wi-Fi Ad-hoc networks, Wireless Internet Conference (WICON), 2010 The 5th Annual ICST , vol., no., pp.1,9, 1-3 March 2010

[104] M. Johansson and L. Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. Wireless Communications, IEEE Transactions on, 5(2):435445, Feb. 2006.

[105] Jim Geier, Assigning 802.11b access point channels, Wi-Fi Planet.

[106] X. Lin and S. Rasool. A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-hoc Wireless Networks. In Proceedings of IEEE INFOCOM, pages 11181126, May 2007.

[107] Farago, A., Myers, A.D., Syrotiuk, V.R., Zaruba, G.V. Meta-MAC protocols: automatic combination of MAC protocols to optimize performance for unknown conditions IEEE Journal on Selected Areas in Communications, Vol: 18, 9, 1670 - 1681, 2000.

[108] R. Badalamenti, I. Tinnirello and L. Giarrè, F. Giuliano A distributed learning mechanism via load aware adaptive protocol decision for frequency planning in High Density WLANs, Techinical report 2014.