UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

SCUOLA DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE

INDIRIZZO IN SCIENZA E TECNOLOGIA DELL'INFORMAZIONE

XXIII^{o} Ciclo

Network Coding for Cognitive Radio Networks

Dottorando

ALFRED ASTERJADHI

Supervisore:

Chiar.^{mo} Prof. Michele Zorzi

Direttore della Scuola:

Chiar.^{mo} Prof. Matteo Bertocco

Anno Accademico 2010/2011

Contents

List of Acronyms

CRAHN Cognitive Radio Ad Hoc Network 103

CAP Cognitive Access Point 103

CCC Common Control Channel 76

TDD Time Division Duplex 112

CFD Cyclostationary Feature Detection 65

MFD Matched Filter Detection 65

CD Cooperative Detection 77

SNR Signal to Noise Ratio 96

DoS Denial of Service 125

RLNC Random Linear Network Coding 10

TCP/IP Transport Control Protocol/Internet Protocol 135

CLO Cross Layer Optimization 136

ISO/OSI International Organization for Standardization/Open Systems Interconnection 135

NUM Network Utilization Maximization 139

ULLA Universal Link Layer API 155

GENI GEneric Network Interface 155

CAPRI Common Applications Requirements Interface 155

Abstract

Cognitive Radio Networks are a promising technology likely to be deployed in the very near future as a viable solution to the spectrum shortage problems faced by traditional wireless systems. Technological breakthroughs in the field of Software Defined Radios enabled the development of flexible cognitive radio transceivers capable of dynamically changing their transmission parameters in order to efficiently exploit the available wireless resources. This increased capability of cognitive radios to self adapt based on interactions with the surrounding environment makes them the perfect candidates for opportunistic spectrum access in those bands that are assigned to primary users. While these primary users are allowed to access their licensed spectrum resources anytime and anywhere, within the contractual limits imposed by spectrum management authorities, cognitive radios have to scan and identify any unused spectrum in the licensed bands. Most importantly, in order not to interfere with primary users, they have to rapidly vacate the licensed spectrum as soon as the primary user begins to use its legitimate spectrum resources.

The coexistence of cognitive radios with such primary users is very challenging. When considering the natural evolution of cognitive radio networks to more complex systems, the challenges and problems to be faced increase substantially. More specifically, the inherent capability of cognitive radios to base their decisions on their view of the wireless spectrum makes their operation susceptible to a variety of malicious attacks. Hence, in such a challenging environment, mechanisms such as cooperation, learning, and negotiation help cognitive radios make the necessary decisions to ensure reliable communications in a non-interfering manner. We hereby investigate a novel architectural solution for Cognitive Radio Networks that uses network coding for fast control information exchange among cognitive radios, enabling them to maintain coherent and reliable information regarding the status of the wireless environment. This control information is used by cognitive radios to perform cooperative detection of primary users and efficient reuse of the available spectrum resources while guaranteeing robust communication and a prompt reaction to wireless environmental changes.

Sommario

Le Reti Radio Cognitive sono una tecnologia promettente che potrà essere utilizzata in un futuro molto prossimo, come soluzione possibile al problema di spettro limitato riscontrato nelle reti wireless tradizionali. Le recenti innovazioni tecnologiche su Software Defined Radio hanno permesso lo sviluppo di ricetrasmettitori radio sufficientemente flessibili, in grado di modificare dinamicamente i loro parametri di trasmissione, al fine di sfruttare in modo efficiente le risorse radio disponibili. Questa maggiore capacità di autoadattarsi in risposta alle interazioni con l'ambiente circostante rende le radio cognitive i candidati ideali per l'accesso opportunistico nelle bande dello spettro radio utilizzate dagli utenti primari. Poiché gli utenti primari hanno diritto ad accedere alle proprie risorse dello spettro in qualsiasi momento e luogo, entro i limiti contrattuali imposti dalle autorità di gestione dello spettro radio, le radio cognitive devono effettuare la scansione e identificare le porzioni dello spettro non utilizzate dagli utenti primari. Inoltre, al fine di non interferire con tali utenti, esse devono liberare rapidamente lo spettro radio, ogniqualvolta l'utente primario inizi ad utilizzare la propria banda di frequenze.

La coesistenza tra radio cognitive e utenti primari è un obiettivo molto ambizioso. Inoltre, quando si considera la normale evoluzione delle reti radio cognitive a sistemi più complessi, le sfide e i problemi da affrontare aumentano notevolmente. In particolare, la capacità intrinseca delle radio cognitive di basare le proprie decisioni sulla propria visione locale dello spettro radio rende il loro funzionamento sensibile a molte tipologie di attacco. Quindi, in tale ambiente, meccanismi come la cooperazione, l'apprendimento e la negoziazione sono di aiuto alle radio cognitive nel prendere le decisioni necessarie a garantire le proprie comunicazioni senza interferire con quelle degli utenti primari. In questa tesi si presenta un'architettura innovativa per le Reti Radio Cognitive che utilizza la tecnica di *network coding* per lo scambio di informazioni di controllo tra i nodi, riguardo lo stato dell'ambiente radio, in modo rapido, coerente e affidabile. Questa informazione viene successivamente utilizzata dalle radio cognitive per eseguire il rilevamento cooperativo degli utenti primari e il riutilizzo efficiente dello spettro a disposizione, garantendo una comunicazione robusta e una reazione tempestiva ai cambiamenti dello stato delle risorse radio.

1 Introduction

TRADITIONALLY wireless networks have been operating based on fixed spectrum assignment policies. According to these policies, licensees are granted the rights for exclusive use of frequency bands on a long term basis over vast geographical areas. Because of this static allocation of the available spectrum resources, several portions of the licensed bands are unused or underused at many times and/or locations [1]. On the other hand, several recent technologies - such as IEEE 802.11, Bluetooth, ZigBee, and to some extent WiMAX - that operate in the Industrial, Scientific and Medical (ISM) unlicensed bands, have experienced a huge success and proliferation. As a consequence, the wireless spectrum they are accessing - especially the 2.4 GHz ISM band - has become overcrowded. In an effort to provide further spectrum resources for these technologies, as well as to allow potential development of alternative and innovative ones, it has recently been proposed to allow unlicensed devices, called secondary users, to access those licensed spectrum resources that are unused or sporadically used by their legitimate owners, called primary users. This approach is normally referred to as Dynamic Spectrum Access and the technology that enables secondary users to find and opportunistically exploit unused or underused spectrum bands is called Cognitive Radio [2].

The concepts of Dynamic Spectrum Access and Cognitive Radio have attracted significant attention by the research community since the recent achievements in the field of Software Defined Radios [3]. These achievements provided the required technological background for the realization of low-power Cognitive Radio transceivers which are able to change their transmitter parameters (operating frequency, modulation, transmission power and communication technology) as a response to changes in the wireless environment.

Cognitive Radio Networks have consequently emerged as viable architectural solutions to alleviate the spectrum shortage problem faced by traditional wireless networks [4, 5] by exploiting the existing wireless spectrum opportunistically. However, when designing such solutions it is necessary to consider that, besides the strict requirements imposed by the opportunistic coexistence with Primary Users, Cognitive Radios may also have to deal with other malicious/selfish (adversary) Cognitive Radios that aim at denying/gaining access to the available spectrum resources with no regard to fairness or other behavioral etiquettes. This is possible because the same Software Defined Radio technology can enable adversary Cognitive Radios to significantly modify the perception that legacy Cognitive Radios have of the surrounding environment, resulting in suboptimal or interruption of operation for Cognitive Radio Networks.

Hence, in order to opportunistically access the licensed spectrum in a non interfering manner and, at the same time, guarantee their own communications in the face of malicious attacks, Cognitive Radios must rely on mechanisms such as cooperation, learning and negotiation. By observing the wireless environment, exchanging information, and evaluating different actions, Cognitive Radios can take the appropriate countermeasures to guarantee the continuity of their communications and the integrity of Primary Users' activity.

To tackle these issues we propose an architectural solution for Cognitive Radio Networks which uses network coding techniques for reliable control information exchange and enables Cognitive Radios to maintain up-to-date information regarding the network status and promptly react to wireless environmental changes. Its main features are: 1) a robust neighbor discovery algorithm able to guarantee fast and reliable network deployment; 2) a robust control channel for prompt control information exchange; 3) efficient cooperative detection of Primary Users' activity; 4) distributed allocation of the spectrum resources to Cognitive Radios for both single hop and multi hop Cognitive Radio Networks; 5) a spectrum aware cluster formation protocol that allows spectrum reuse and network scalability.

We approach the problem systematically, first identifying the opportunities offered by network coding in practical settings and subsequently solving the main problems that need to be faced in Cognitive Radio Networks.

In Figure 1.1 we show the structural organization of this work, with a particular focus on the interdependence between different chapters. Our discussion begins in Chapter 2 which introduces network coding and all the necessary components to design a practical dissemination scheme which will be used throughout this thesis.

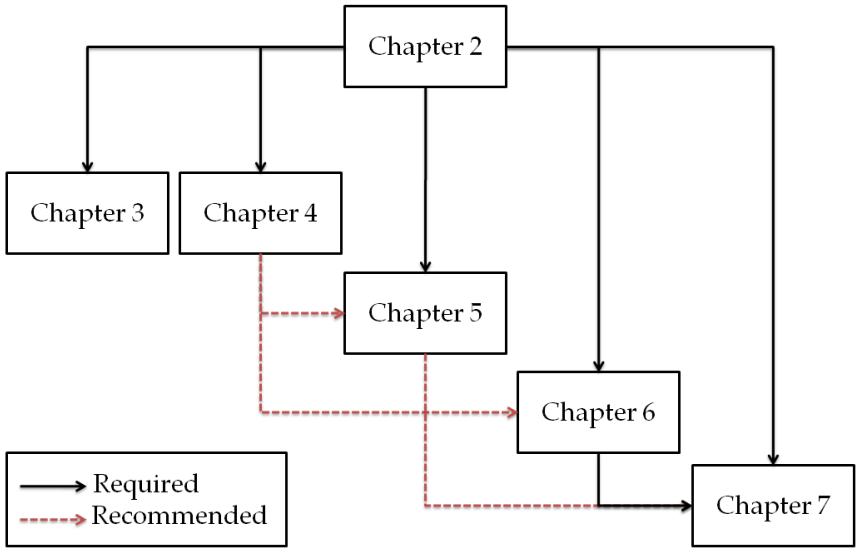


Figure 1.1. *Structural representation of the thesis.*

To fully understand the benefits of network coding, in Chapter 3 we investigate practical network coding based algorithms for data broadcasting in single channel wireless ad hoc networks. In these networks, deadlock situations may occur causing the delivery process to stop before all nodes[1] in the network are able to gather and decode all the required packets. To tackle this problem we analyze the performance of a proactive mechanism (called proactive network coding) which is able to detect when nodes need additional packets in order to decode the information. We conclude our investigation of wireless ad hoc networks by studying the performance of network coding in multi-rate wireless technologies, such as the IEEE 802.11g standard, and proposing a distributed heuristic approach for the selection of data rates which guarantees reliable and fast data delivery.

Chapter 4 investigates the problem of data broadcasting in multi channel networks where normal nodes coexist with adversary nodes that launch different attacks in an attempt to interrupt the dissemination process. In this context we evaluate the performance of different Medium Access Control and Broadcasting protocols. The analysis led to the identification of the necessary conditions for an efficient use of network coding in a multi channel scenario which is the starting point for the development of Cognitive Radio Networks.

In Chapter 5 we propose a Jamming Evasive Network coding Neighbor discovery Algorithm which ensures complete neighbor discovery for a Cognitive Radio Network even in

[1]Cognitive Radios (CRs) and nodes will be used interchangeably throughout this thesis.

presence of jammers[2]. Using network coding for efficient data dissemination and random channel hopping for jammers' avoidance, we design a neighbor discovery scheme which is fully distributed, asynchronous and very robust to jamming attacks. In addition, it does not need to know in advance the number of nodes in the network and ensures fast neighbor discovery even in the case when all nodes are required to simultaneously terminate the neighbor discovery process.

Chapter 6 introduces the architecture of the proposed Dynamic Spectrum Access scheme (NC^4-DSA) that enables Cognitive Radios to opportunistically and efficiently access the channels available for communications in a single hop scenario. It addresses the following important aspects of opportunistic spectrum access: 1) implementation of a control channel, 2) multi channel medium access control, 3) Primary Users detection, and 4) secondary reuse of spectrum unused by Primary Users. The scheme is completely distributed, does not need dedicated spectrum resources for control purposes, and exploits a cooperative detection strategy to identify unused spectrum. Due to these aspects, our scheme represents a significant improvement with respect to existing Dynamic Spectrum Access solutions. In order to assess its performance we carry out an evaluation study for different scenarios and system parameters, showing that the proposed scheme is feasible, capable of providing satisfactory performance, and suitable for implementation in real systems.

We successively discuss the extension of the proposed scheme for Cognitive Radio Ad Hoc Networks in Chapter 7 where the increase in network size poses additional design challenges. We preliminarily focus on the implementation of the control channel where the increase in network size directly impacts the efficiency of the network coding based control channel and the resource allocation algorithms that are the core of our proposal. We show that for limited size Cognitive Radio Ad Hoc Networks, NC^4-DSA is still capable of providing very good performance in terms of control channel functionality and Cognitive Radio's achievable goodput. However, in order to solve network size related issues with the control channel we propose Clustered NC^4-DSA (CNC^4-DSA) which limits the decoding complexity and improves dissemination efficiency by establishing a control channel for each cluster in the network. Resource allocation and spatial frequency reuse are provided by a graph-coloring algorithm that significantly reduces both intra- and inter-cluster interference while avoiding to operate on those spectrum bands that are used by legitimate Primary Users in the area. We note that the proposed scheme does not put limitations on the protocol used

[2]Jammer and adversary will be used interchangeably throughout this thesis.

for the division of the network in clusters as long as it guarantees connectivity and limits the cluster size. However, given that network coding is capable of fully utilizing the available channels for control information dissemination, it is advisable that nodes with similar spectrum availability be members of the same cluster. Hence, we propose a novel spectrum aware cluster formation protocol that organizes the network taking into consideration the spectrum availability. This way it is possible to increase the number of intra-cluster parallel transmissions and reduce the time required for control information dissemination among Cognitive Radios of the same cluster. We conclude by analyzing security issues related to the deployment of our proposal and compare it with existing architectures in terms of drawbacks and benefits.

As a concluding note, in this thesis we propose a novel Dynamic Spectrum Access architecture for Cognitive Radio Networks which enables Cognitive Radios to operate in a completely distributed and autonomous way. It provides an efficient reuse of the licensed spectrum resources while guaranteeing not to interfere with the Primary Users' communications. Its featured control channel allows the timely dissemination of control information among all nodes in the network allowing the implementation of efficient routing, resource allocation and network maintenance, etc. These services are vital for the opportunistic utilization of wireless spectrum resources which are shared with a wide variety of ever growing wireless devices.

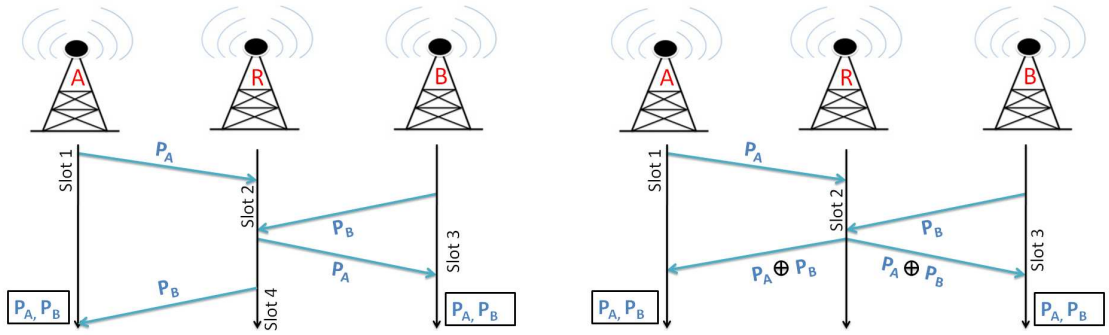
2

Network Coding Basics

Network coding is a recently introduced paradigm for data dissemination in wireless networks able to increase throughput, reduce delay, and enhance robustness. In contrast to traditional store and forward approaches, it provides a store, code and forward technique where each node stores all the incoming packets in an internal buffer and transmits their linear combinations, where combining is performed over some finite Galois Field. This technique allows for increased throughput efficiency as well as scalability and robustness due to its inherent capability of approaching the network capacity in practical settings.

NETWORK coding was first introduced in their seminal work by Ahlswede et al. [6]. We can define it as a particular in-network data processing technique that exploits the characteristics of the wireless medium, in particular the broadcast communication channel, in order to increase the achievable throughput of wireless networks.

In contrast to the *store* and *forward* paradigm, network coding implements a more complex *store*, *encode*, and *forward* approach where each node stores the incoming packets in its own buffer, and successively sends a combination of the stored data. In order to successfully decode, e.g., n packets, a node must collect at least n independent combinations of the original packets. This way it can provide high throughput gains in multicast or broadcast networks. More specifically, network coding can typically achieve higher transmission rates



(a) Store and Forward

(b) Store, Encode and Forward

Figure 2.1. *Example of data dissemination in a wireless ad hoc network: traditional store and forward vs. network coding.*

than separate unicast transmissions when information sources transmit to multiple destinations or to all nodes in the network.

2.1 A First Example

In Figure 2.1 we show an example of network coding in a simple two hop wireless network and compare it to the traditional store and forward approach. The network consists of two nodes A and B and a relay node R. In this example, nodes A and B are interested in exchanging with each other the data packets P_A and P_B, respectively. The distance between A and B is such that it is not possible for them to directly exchange their packets, i.e., they are not within each other's transmission range. Hence, in order to exchange information they have to relay their transmissions through node R.

Focusing on Figure 2.1, we see that for both the traditional (*store and forward*) approach and network coding (*store, encode and forward*), during the first two time slots nodes A and B forward their packets to the relay node R. Once the relay node R has received both packets P_A and P_B it can use the traditional approach (see Figure 2.1(a)). According to it, the relay node R forwards the packets in subsequent time slots (P_A is transmitted in time slot 3 and P_B in time slot 4) allowing to deliver the information to both destination nodes A and B in 4 time slots. On the other side, if network coding is used (see Figure 2.1(b)), the relay node R transmits a XOR-ed version of packets P_A and P_B to both nodes A and B. At this point, node

A (B) can decode packet P_B (P_A) by simply subtracting its own packet from the received one. This way, it is possible for the nodes to receive the packets in 3 time slots instead of 4 time slots, as required by the traditional approach. This example shows how network coding is particularly effective whenever there are overlapping data flows as it can exploit both the broadcast nature of the wireless channel and the coding process to simultaneously deliver different packets to multiple users.

2.2 Practical Network Coding

In this section we describe the principles that stand behind a practical implementation of network coding in distributed wireless networks [7].

Let us consider a system that acts as information relay in a generic network. Moreover, let the information flowing in the network be represented by the source packets $x_i \in \{x_1, \ldots, x_n\}$. Traditionally, in order to deliver a source packet to some destination nodes the relay node simply forwards it as is. With network coding instead, the relay node is enabled to combine a number of packets it has received into an encoded packet which is then forwarded to the destination nodes.

Assume that each source packet consists of b bits. We note that if the source packets have different sizes, the shorter ones can be padded with trailing 0s to have the same size across all source packets. A source packet, x_i, can be interpreted as a vector $\underline{x}_i$ over some finite Galois Field, GF(2^q). Given that the packet size is b bits its vector representation will have $\lceil b/q \rceil$ elements each of them of q bits. With linear network coding [8], outgoing packets $\underline{y}$ are linear combinations of the source packets, where the operations of addition (+) and multiplication ($\cdot$) are performed over the field GF(2^q). That is:

$$\underline{y} = \sum_{j=1}^{n} g_j \cdot \underline{x}_j, \tag{2.1}$$

where the combination coefficients can be grouped to create a vector $\underline{g} = [g_1, \ldots, g_n]$ which is called the *global encoding vector*. This vector is needed by the receiving nodes to decode the information contained in the encoded packet as we will explain later.

Returning to our example of Figure 2.1(b), we have that GF(2^1) = {0, 1}, i.e., the elements of the vector $\underline{x}_i$ are one bit each and its length is equal to the packet size. Hence, the linear combination sent by R in time slot 3 (after receiving $\underline{x}_1 = P_A$ and $\underline{x}_2 = P_B$) is $\underline{y}$ = $\underline{x}_1$ + $\underline{x}_2$ where the summation occurs for every symbol position, i.e., in this case bitwise xor.

We note that the procedure of encoding can be performed recursively, i.e., by linearly combining already encoded packets. Assume a node i has received and stored a set of encoded packets $\mathcal{C} = \{\underline{z}_1, \ldots, \underline{z}_m\}$ and their corresponding global encoding vectors $\mathcal{G} = \{\underline{g}_1, \ldots, \underline{g}_m\}$. This node may generate a new encoded packet by picking a set of coefficients $\xi_1, \ldots, \xi_m$ and computing the linear combination $\underline{y}_i = \sum_{j=1}^{m} \xi_j \underline{z}_j$. In this case the global encoding vector g_i associated with packet y_i is given by $g_i = \sum_{j=1}^{m} \xi_j \cdot g_{i,j}$.

As for the decoding procedure we note that a node i that wants to retrieve the source packets needs to solve the system:

$$\begin{bmatrix} \underline{z}_1 \\ \vdots \\ \underline{z}_m \end{bmatrix} = \begin{bmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & \ddots & \vdots \\ g_{m,1} & \cdots & g_{m,n} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_n \end{bmatrix} = G \begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_n \end{bmatrix} \tag{2.2}$$

This is a linear system with m equations and n unknowns. Hence, node i needs $m \geq n$ to have a chance to solve the system and recover the source packets. That is, the number of received packets must be at least as large as the number of source packets generated in the network. However, we note that this condition is necessary but not sufficient as some of the received encoded packets may be linearly dependent with each other. An efficient network coding scheme is able to encode the packets in such a manner that the probability of receiving linearly dependent packets is very low. A simple way of achieving this is by having nodes choose the coefficients uniformly at random over the field GF(2^q). In the case of this random linear network coding this probability is related to the Galois field size [9]. Simulation results [7, 10] show that this probability becomes negligible even for small field sizes such as for example $q = 8$. Moreover, Random Linear Network Coding allows nodes to operate in a completely independent and decentralized manner which is appropriate for operation in wireless networks. Hence, throughout this thesis we will use Random Linear Network Coding (RLNC) and will refer to it as network coding (NC).

In our model we have that each node $i \in \{1, \ldots, n\}$ in the network generates a single source packet x_i and is interested in receiving the source packets generated by all the other nodes. It stores the encoded packets it receives as well as its own packet, row by row, in a *decoding matrix*. Initially, the decoding matrix contains only the source packet generated by node i. Successively, when an encoded packet is received it is appended as the last row to the decoding matrix. The matrix is then transformed to a triangular matrix using Gaussian elimination. An encoded packet that increases the rank of the matrix after its reception is

called an *innovative packet*. If the decoding matrix, at some point, has a row of the form $e_j = [0, \ldots, 1, \ldots, 0]$ where 1 is located at the jth position, the node knows that the source packet x_j associated to that position can be recovered. In the general case, using Gaussian elimination, a submatrix of the decoding matrix may become upper triangular, i.e., there exists a subset of the encoded packets received so far that allows decoding of some source packets. In this case it is possible to prematurely decode that particular subset of source packets. If this event occurs before the decoding matrix has full rank (equal to n) then we have early decoding. Otherwise, the node can decode all the information contained in the decoding matrix (buffer) at the latest after receiving n linearly independent combinations of the source packets. We note that the size of these matrices has to be limited mainly due to the complexity of solving a linear system with n unknowns. For this reason, packets are usually grouped together into so-called generations, and only packets of the same generation can be mixed together [7]. The same considerations hold for the size of the Galois field which, together with the dimension of the generations, allows to reduce both memory requirements and computational complexity.

2.3 Network Coding for Data Dissemination

Network coding allows for increased throughput efficiency as well as scalability and robustness [11]. These benefits arise in the case of multicasting [12, 13] as well as for other network configurations, such as multiple unicast communications [14, 15]. Moreover, they are not restricted to error-free communication networks, but can also be obtained in ad hoc networks [16–18], peer-to-peer systems [19], and optical networks.

Important theoretical results are known, see, e.g., [6, 12, 20–22], and research is now moving towards the exploitation of network coding in practical communication protocols. To this end, the work in [12] is of considerable importance as it demonstrates that random linear network coding is able to reach network capacity in practical settings. This is very important as linear random coding is lightweight and inherently localized and, as such, can be exploited by communication protocols at low overhead.

Related work on communication protocols for wireless networks can be found in [7, 14, 15, 17, 18, 23, 24]. [7] was the first contribution to present a practical and distributed solution exploiting random linear network coding. The authors focused on how the coding matrix as well as the information related to the random combination of packets in some finite Galois

Field GF(2^q) can be shared by different nodes at low overhead. This is a crucial aspect for network coding algorithms to work in multi-hop radio networks.

COPE [14] applies network coding to unicast flows in wireless networks. The authors of the paper experimentally show that significant gains, in terms of maximum throughput, are possible even in the case of unicast transmissions and even when network coding is implemented through simple XORing of packets within a single-hop neighborhood (rather than forwarding encoded packets over multiple hops). [15] presents BFLY, a localized network coding protocol which recognizes butterfly structures in the network to exploit the coding opportunities they represent. This protocol builds on COPE and also encodes packets through XOR operations, but it additionally allows the transmission of encoded packets over multiple hops. The work in [14, 15] presents practical communication schemes, where [14] quantifies the achievable gains from coding when an actual Medium Access Control (MAC) layer is used. Our focus is different in terms of network scenario, as we consider an all-to-all communication paradigm and encode packets in GF(2^q) with $q = 8$, considering more general coding rules.

In [17], the authors investigate the interaction between MAC and network coding in wireless multi-hop networks, and propose distributed and opportunistic scheduling rules for the combination of packets in the presence of time-varying fading links. They also look at the impact of MAC schedules. However, this topic is treated differently from what we do here as packets at the relay nodes are XORed and possible deadlocks in the data dissemination are not investigated.

The authors of [24] study the interaction of network coding and MAC, devising suitable conflict-free transmission schedules (for a given connectivity graph) and related off- and on-line algorithms for wireless multi-hop networks. However, their strategies entail some coordination among nodes which incurs additional communication overhead with respect to random linear network coding.

Reference [18] studies broadcasting scenarios and introduces a class of lightweight reactive and distributed network coding protocols based on random linear network coding, proving the superiority of these schemes over flooding [25] and epidemic routing [26].

3

Broadcasting in Single Channel Wireless Networks

Practical dissemination algorithms can exploit network coding for reliable data broadcasting in single channel wireless ad hoc networks. The efficient design of such algorithms must take into account issues related to the use of network coding in realistic wireless environments. Identifying their impact on the network performance allows to achieve substantial benefits by designing heuristic and proactive mechanisms that optimize network operation under these conditions.

NETWORK coding based algorithms [18] are suitable for data broadcasting in single channel wireless ad hoc networks. According to these algorithms, whenever an innovative packet is received at a given node, it generates with probability ρ a new packet through network coding and broadcasts it over the wireless channel. For the MAC protocol we consider several variants of Carrier Sense Multiple Access (CSMA). We initially focus on the interaction between MAC and network coding over different wireless network configurations in order to capture the effects of each protocol component and quantify the performance degradation due to packet collisions and

The material presented in this chapter has been published in [27].

random transmission schedules [28]. Subsequently, the discussion is extended to a Proactive Network Coding (ProNC) dissemination mechanism [29] for which we show that it outperforms the reactive network coding schemes proposed in [18].

The results that we discuss in this chapter can be considered as the natural continuation of [18] and the extension of [28,29], which we complement by investigating the performance degradation due to actual MAC schemes and analyze the proposed solution to deadlocks in the dissemination of the information which may occur in certain topologies. The performance evaluation that we carry out in this chapter is based on simulation results obtained using network simulator 2 (ns2); the relevant simulation code can be downloaded from [30].

The remainder of this chapter is organized as follows. In Section 3.1 we describe the problem and the network model used along this chapter. In Section 3.2 we continue describing several IEEE 802.11 MAC variants and in Section 3.3 we discuss different random packet combination strategies based on network coding. In Section 3.4 we present simulation results to quantify their impact on the performance of data broadcasting. Section 3.5 presents a lightweight and distributed mechanism to perform network coding more efficiently and show via simulations that it can significantly outperform the data dissemination schemes of [18]. Other results are given for multi-rate environments in Section 3.6: on the one hand, high data rates are good as they shorten packet transmission times, thereby reducing the collision probability; on the other hand, however, in a multi hop scenario high data rates mean that packets have to travel more hops to reach the destination. Thus, determining suitable data rates for each node, so as to obtain good tradeoffs between packet delivery delay and delivery ratio, is a challenging and interesting problem for which we propose a heuristic solution in Section 3.6.

3.1 Problem Description and Network Model

Wireless ad hoc networks are severely constrained by interference and channel impairments, especially in the case of broadcast communication. The use of traditional access mechanisms such as CSMA-like protocols, when multiple nodes transmit, may suffer from a high number of collisions and dropped packets. Two main factors are to be taken into account when using network coding in conjunction with an actual MAC, namely 1) *collisions* and 2) *packet scheduling*. Both collisions and scheduling are the direct consequence of the random (CSMA-like) channel access that we adopt in this study. Collisions impact the

performance as fewer packets are collected; as a consequence it takes longer to obtain full rank decoding matrices at the receivers. Packet scheduling refers to the way in which different nodes take turns in transmitting, which is dictated by the MAC rules. The transmission order is important when network coding is used at higher layers as it influences the way encoded packets are created, i.e., which packets are mixed together.

Traffic pattern: each node $i \in \{1, \ldots, n\}$ inserts into the network a single source packet x_i and wants to collect all the other inserted packets. x_is are generated either *randomly* or *deterministically*. In the former case, each node inserts its source packet by independently picking the insertion time uniformly in a fixed length interval of $\Delta_1 = 100$ ms. In the latter case, we can assume to have a simple application that inserts source packets sequentially in each node. Subsequent insertions, at different nodes, are separated by fixed time intervals of $\Delta_2 = 1$ s. For this value of Δ_2, with the considered scenario (e.g., transmission times and network size), the collision probability is negligible for both source packets and subsequent transmissions elicited by network coding. This is useful to assess the performance of the dissemination schemes when used with an ideal MAC.

Network topologies: we start our investigation with circular and grid reference scenarios and then consider random topologies. We ensure that all random topologies used in the simulations are connected. To this end, we do a simple breadth-first-search of the underlying connectivity graph and check if all nodes are visited (a standard procedure to check for connectivity). The topology is valid (i.e., it is used in the simulations) if a single connected cluster exists and is discarded otherwise.

Physical (PHY) layer: we have implemented an extended version of the ns2 PHY layer for IEEE 802.11b/g which includes Packet Error Rate (PER) calculations accounting for modulation, channel effects, and multi-user interference. In detail, the Signal to Interference plus Noise Ratio (SINR) is evaluated for each receiving node and for each packet taking into consideration the interference generated by nearby transmitters. PERs are obtained from pre-calculated packet error rate curves. For the channel, we use the standard ns2 two ray ground propagation model. More details can be found in [31].

3.2 MAC Protocols

We consider four different MAC protocols based on CSMA, which is currently the most widely used MAC mechanism in wireless ad hoc networks.

IEEE 802.11b: Is is considered as the baseline MAC. We adopt the basic medium access provided by IEEE 802.11b that, in the broadcast mode, does not use any Acknowledgment (ACK) mechanism. Hence, no retransmission occurs if the packet is lost.

IEEE 802.11b PB: IEEE 802.11b with Pseudo Broadcast (PB) [14] is an extension of the basic IEEE 802.11b, where an ACK mechanism is implemented. A given node randomly picks a neighbor and sends a packet to it via unicast by including its address in the packet header. All other neighbors are in promiscuous mode and can overhear/decode the transmission. However, only the neighbor which is the intended receiver of the unicast sends back an ACK. This is done according to the basic IEEE 802.11b unicast communication mode (without Request To Send/Clear To Send (RTS/CTS)). The packet is retransmitted, after a backoff period, in case there is no ACK from the intended receiver. Using this mechanism, only collisions at the addressed receiver can be detected, while collisions occurring at any of the other neighbors are ignored. Moreover, it does not solve the hidden terminal problem.

IEEE 802.11b PB RTS/CTS: To further improve the performance we consider the previous scheme with an additional RTS/CTS handshake. These control messages are introduced to alleviate the hidden node problem. The Clear To Send (CTS) is only transmitted by the node addressed in the packet header.

Ideal MAC: In this case transmitted packets are only affected by the delay, $\Delta = \ell_p + \Delta_{\text{prop}} \simeq \ell_p$, where Δ_{prop} is the propagation delay and ℓ_p is the packet transmission time, i.e., we assume to have an omniscient entity which regulates the transmissions to completely avoid interference and collisions. Hence, as a node sends a packet, all its neighbors successfully receive the message after the (fixed) delay Δ. Note that this MAC still schedules packet transmissions according to CSMA and is thus non-ideal from a packet scheduling point of view when network coding is used at the upper layers. This idealized scheme is exploited to pinpoint the effect of packet errors on the performance of network coding.

3.3 Reactive Network Coding

In the following we discuss three different packet combination strategies based on NC, where the combination coefficients are scalars randomly picked in GF(2^8). The first two schemes are inspired by the work in [18], while the last one is discussed in [28]. All the presented schemes are characterized by the *forwarding factor* which is defined as follows.

Definition 3.1. *Forwarding factor, ρ: the ratio between the average number of packets transmitted and the average number of innovative packets received per node.*

We recall that a received packet is innovative whenever it increases the rank of the decoding matrix [18]. For the schemes below ρ is decided a priori and equal for all nodes.

3.3.1 Probabilistic network coding

Each node sends a random linear combination of the packets in its buffer. When receiving an innovative packet a new combination is transmitted with probability ρ whereas nothing is transmitted with probability $1-\rho$. For example if the forwarding factor is $\rho = 0.5$ it means that a node on average sends a new packet every two innovative packets received. From [32] we know that $\rho = 0.5$ would theoretically (circular topology, ideal scheduling, and no collisions) assure a packet delivery ratio of 1 when the number of neighbors is 2 (the packet delivery ratio is defined in Section 3.4).

3.3.2 Semi-deterministic network coding

In this case, for a given forwarding factor ρ, each node sends out a new combination after having received exactly $\lceil 1/\rho \rceil$ innovative packets. As an example, $\rho = 0.5$ means that each node deterministically transmits a new combination every two received innovative packets. The forwarding factor, in this case, is not related to a probability but is rather used as a threshold on the number of incoming packets.

3.3.3 Timed network coding

The two previous schemes have two major drawbacks. The first is that they are particularly sensitive to packet losses, e.g., due to collisions. In fact, if one of the transmitted packets is lost, the propagation of the information through the network could be interrupted. The second drawback is that both probabilistic and semi-deterministic network coding suffer from some inefficiencies when there is a small number of packets to combine. In such cases, new combinations are created from a small set of packets and, for this reason, are often not innovative. To alleviate these problems, we introduce a *timing strategy* into the first scheme.

For each received innovative packet a timer is activated. When the timer has expired, the node decides to send out a new random combination with probability ρ. The timer, τ, is a uniform random variable in $[0, \tau_{\max}]$. This timing approach has two advantages. With the

introduction of a waiting interval before transmission, nodes have the chance of collecting other innovative packets and send out richer combinations. Moreover, the reduction of the number of transmissions and the random characteristic of the timer help in decreasing the collision probability at the MAC layer.

The drawback of the timed scheme is the introduction of a short delay due to the timer. Hence, the timer value shall be chosen so as to achieve a good trade-off between extra delay and performance improvements. In IEEE 802.11b, this value has to be large enough to allow for the collection of more than one packet, which translates to selecting $\tau_{\max} \approx 10 - 30$ ms. We picked $\tau_{\max} = 20$ ms. Note that in general $\tau_{\max}$ depends on network density and flow demands.

3.4 Performance Evaluation

In this section we discuss the most relevant results we obtained via ns2 simulations. All presented schemes are evaluated taking into account the *random* and the *deterministic* traffic patterns. We test the algorithms varying ρ from 0.1 to 1 and the number of nodes in the network, n from 4 to 64, and we consider a data rate of $R = 1$ Mbps. For all MACs we consider packet lengths of $L = PHY + MAC + NC(n) + P$, where PHY and MAC are the physical and the MAC headers, respectively, with $PHY = 192$ bits, $MAC = 224$ bits. $NC(n)$ is the network coding header that depends on the number of nodes n and is of size $NC(n) = 8(18 + n)$ bits. P is the payload size that in our case is 64 bits. Due to the inefficiencies of 802.11 medium access and the additional network coding overhead, using a small packet size represents a lower bound on throughput performance. The relative performance differences between the protocols remain unchanged when using larger packet sizes. We organize our performance analysis in two parts: we first focus on the impact of different MAC protocols on network coding and we subsequently evaluate the effect of different packet combination strategies. Our performance metrics are:

Definition 3.2. *Packet Delivery Ratio (PDR): the ratio between the number of successfully received (and decoded) packets and the number of packets a node is interested in.*

Definition 3.3. *Packet Delivery Delay (PDD): the average time between the first transmission of a packet and its reception and successful decoding at the destination nodes. It is only computed for correctly received packets.*

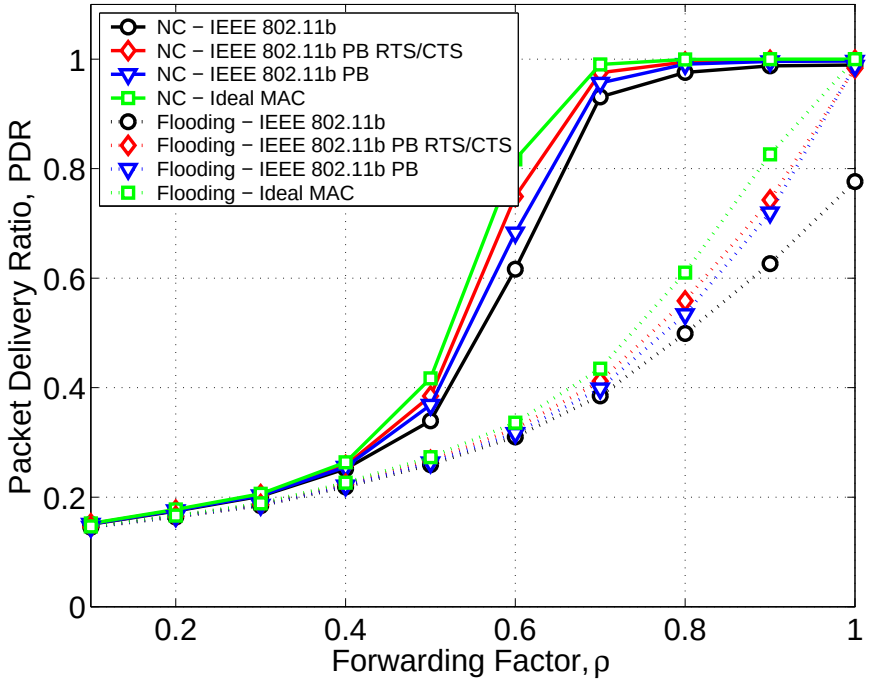


Figure 3.1. *PDR: Performance comparison of Probabilistic NC and Probabilistic Flooding for different MAC protocols in circular networks with* $n = 16$.

Definition 3.4. *Protocol Overhead (OH): the ratio between the number of transmitted packets at the MAC layer and the number of successfully decoded packets.*

3.4.1 Impact of MAC protocols

In Figure 3.1 we compare probabilistic network coding (solid lines) against probabilistic flooding (dotted lines) in terms of PDR in a circular network topology for n=16. Network coding outperforms probabilistic flooding for all values of ρ. Gains are more pronounced when ρ is close to one and n is large (results for different values of n are not shown here as they are similar to those in [18]). As observed in [32], for this topology a PDR$= 1$ is theoretically achievable with ρ slightly larger than 0.5. This is obtained through a proper centralized coordination of the nodes' transmissions which maximizes the probability of sending innovative packets at each transmission attempt. However, this performance level is never reached in practice and the actual PDR depends on the number of nodes. Looking at Figure 3.1 for $\rho = 0.6$, IEEE 802.11b achieves PDR ≈ 0.6, whereas an ideal MAC achieves PDR ≈ 0.8, which corresponds to a decrease in performance of about 25%. Note that our ideal MAC does not provide full reliability as it still schedules transmissions according to CSMA

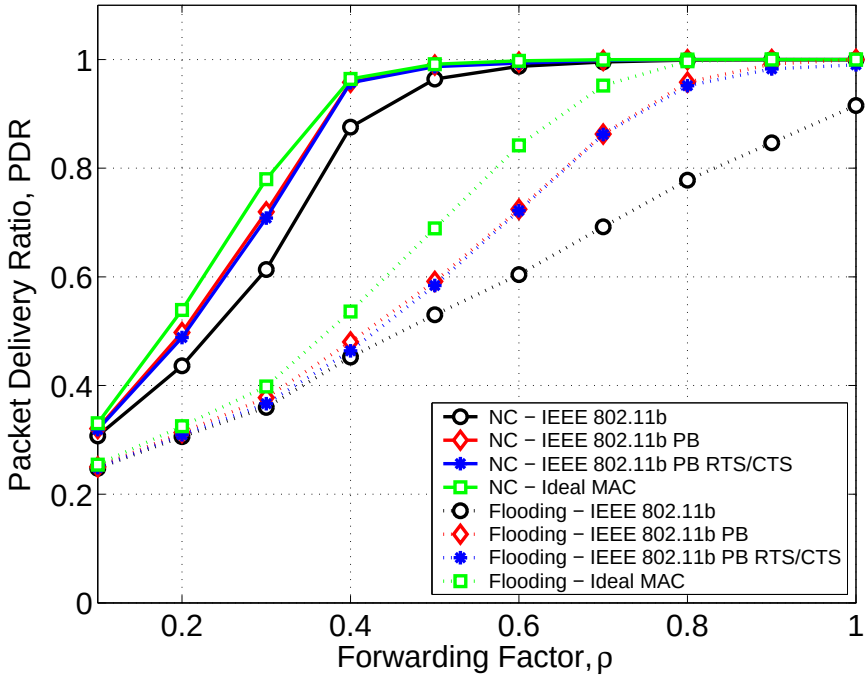
and does not use the optimal coordination strategy of [32]. The effectiveness of pseudo broadcast (IEEE 802.11 PB in the figure) and pseudo broadcast with RTS/CTS (IEEE 802.11 PB RTS/CTS) is also clear, though the improvements are not as large as expected. The observed decrease in performance is due to the use of an actual MAC layer (IEEE 802.11b in this case) and to the sub–optimality of random scheduling, which indicates the importance of these issues for the design of practical schemes.

Figure 3.2 shows results for a different setting where $n = 16$ nodes are placed over a grid. As expected, the achieved performance is better than in the circular case due to the higher number of neighbors per node (4 instead of 2), which favors packet mixing and dissemination. Also in this scenario, the presence of realistic MAC layers significantly reduces the PDR metric for a given ρ (see Figure 3.2(a)). As expected, the schemes implementing collision avoidance policies (i.e., IEEE 802.11b PB and IEEE 802.11 PB RTS/CTS) improve PDR but also increase the protocol overhead. This is due to the MAC retransmissions in case of collisions and to the control traffic (i.e., ACK, Request To Send (RTS) and CTS packets). In addition, we note that when we compare the performance of probabilistic network coding and flooding against ρ, we have a fair comparison as, given a specific ρ and a fixed MAC protocol, both network coding and flooding lead to very similar protocol overhead (see Figure 3.2(b)). Pseudo broadcast and pseudo broadcast with RTS/CTS are effective in decreasing the number of collisions. However, using these additional techniques to recover from packet loss leads to longer delays, as can be seen from Figure 3.2(c). The PDD increase is about one order of magnitude in the worst case (PB RTS/CTS). We also note that the PDD of network coding stabilizes for increasing ρ while it continues to increase for flooding. The reason for this is that with flooding, a higher number of redundant packets is received early on, delaying the reception of innovative packets. For network coding, packets' combination prevents this from happening and most packets received are innovative even for high ρ.

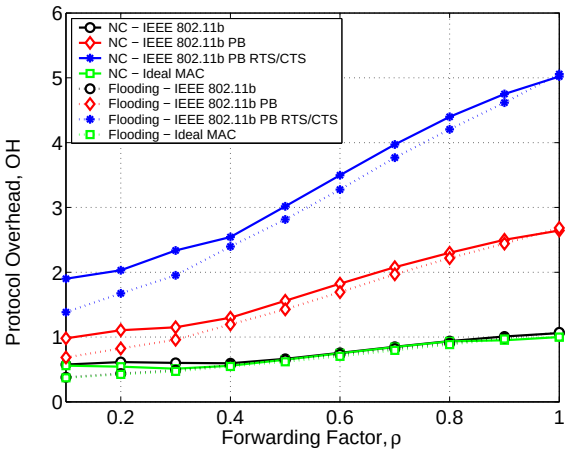
To sum up, we observe that the presence of actual MAC protocols reduces the performance in terms of PDR. In addition, collision avoidance policies give little improvement in terms of PDR, while leading to poor protocol overhead and delay performance.

3.4.2 Impact of packet combination strategies

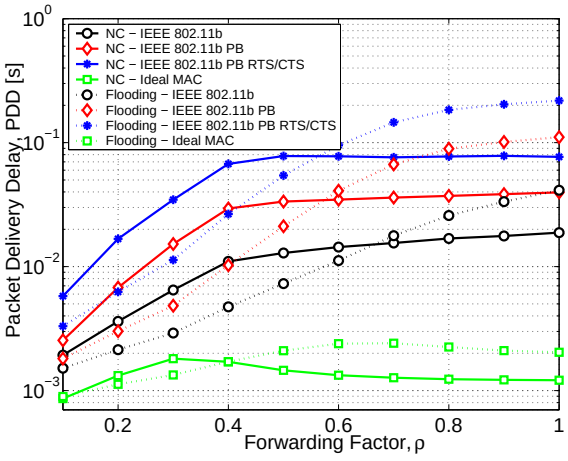
Figure 3.3 shows the PDR performance for a circular network with $n = 16$ for various packet combination strategies for an IEEE 802.11b MAC protocol. The semi-deterministic schemes (dotted lines) show a phase change, where PDR remains constant up to $\rho^* = 0.4$



(a) Packet Delivery Ratio, PDR



(b) Packet Overhead, OH



(c) Packet Delivery Delay, PDD

Figure 3.2. *Performance comparison of Probabilistic NC and Probabilistic Flooding for different MAC protocols in grid networks with* $n = 16$.

and then suddenly increases for higher ρ. This does not occur for probabilistic network coding (solid lines) whose curves are smooth. This reflects the threshold based transmission policy of semi-deterministic network coding. The exact value of the shifting point ρ^* depends on the number of neighbors. For circular networks, where each node has exactly two neighbors, $\rho < 0.5$ ($\lceil 1/\rho \rceil > 2$) never suffices to trigger the transmission of a new combination as the initial number of innovative packets is equal to two. This flaw is not present

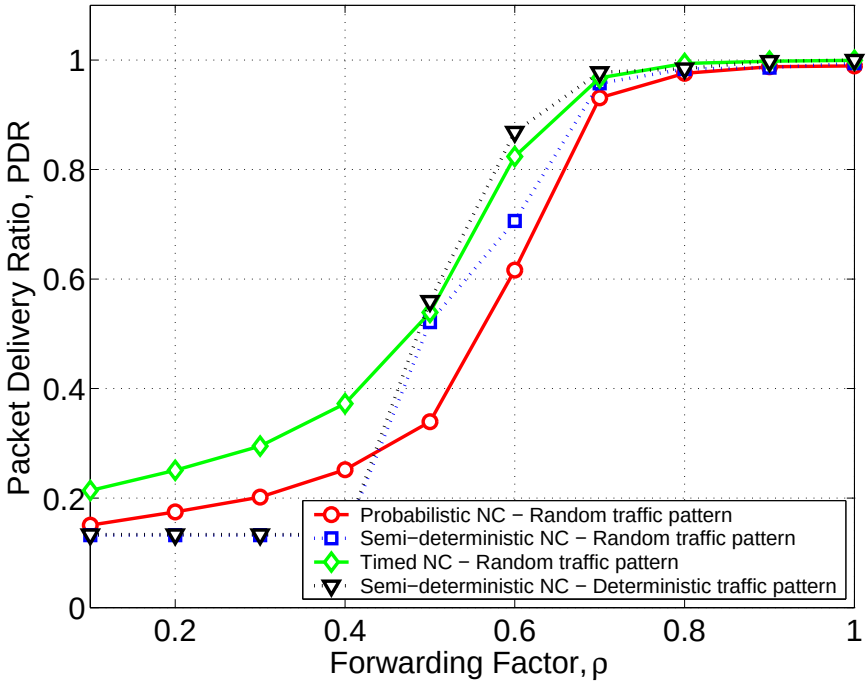


Figure 3.3. *Performance comparison of different combination strategies in circular networks with $n = 16$ and IEEE 802.11 MAC.*

in probabilistic and timed network coding, where sending rules are based on probabilities rather than on hard thresholds. Timed network coding outperforms the semi-deterministic scheme with deterministic traffic pattern for $\rho \leq \rho^*$ and performs very close to this method for larger forwarding factors.

In addition, the timed strategy performs better than both semi-deterministic and probabilistic network coding with random scheduling. For $\rho = 0.5$ probabilistic network coding with random scheduling achieves PDR ≈ 0.35, whereas timed network coding leads to PDR ≈ 0.55 which corresponds to an improvement of about 57%. We observed that the timed strategy introduces an additional delay. Also, there are some expected differences between ideal and actual MAC. For IEEE 802.11b, the PDD increase is reasonably small (approximately equal to the average value of the timer) and is similar to that introduced by the pseudo broadcast algorithms. Hence, the timed combination provides higher benefits than pseudo broadcast in terms of PDR, leading to similar extra delays. For this reason, the timed scheme may make sense when the goal is to maximize the PDR (throughput) while accepting some PDD degradation. This extra delay appears, however, well tolerable (less than 5% increase over the delay without the timed strategy). The results of probabilistic and timed network coding for random topologies are plotted in Figure 3.4 (similar results hold for grid

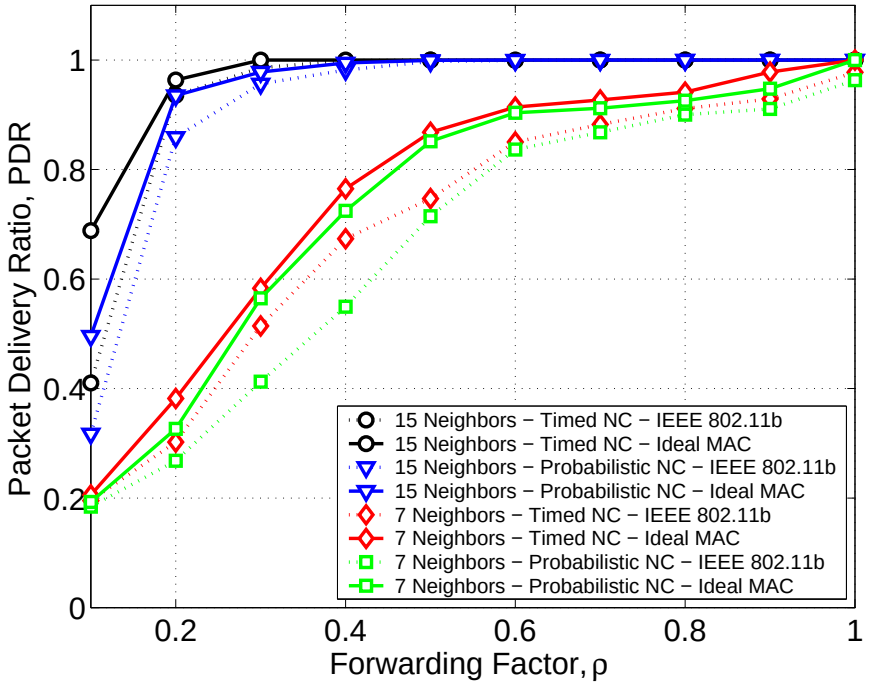


Figure 3.4. *PDR: Performance comparison of Probabilistic NC and Timed NC for ideal and IEEE 802.11b MAC for random networks.*

networks). Note that in these networks the gain becomes larger for decreasing ρ; in fact, when fewer packets are transmitted timed network coding more effectively exploits coding opportunities.

From the above results we see that there exists a gap between the predicted theoretical performance and the results obtained in realistic environments. This gap is due to the random access mechanism of IEEE 802.11, which leads to collisions, and also to its packet scheduling, which does not allow the combination of packets in the optimal order.

3.5 Proactive Network Coding

The schemes considered up to now are *reactive* protocols, i.e., nodes participate in the dissemination of data only when they receive innovative information. If this does not occur, the dissemination is interrupted even though nodes may still have innovative information to send. This fact is an inherent characteristic of the reactive approach. In this section we describe a network coding data dissemination scheme based on a *proactive* approach (referred to in the following as ProNC) to address this problem. Even though our focus here is on scenarios where data is to be exchanged among all the users of a wireless ad hoc network,

the rationale behind ProNC also applies to different settings. This scheme is completely distributed and self-adaptable and requires very limited network knowledge, which can be easily acquired by overhearing the exchanged data.

In the previous section we have seen that reactive schemes are likely to suffer from the presence of interference and collisions in realistic radio environments. The main problem of reactive schemes is that new random combinations are generated and transmitted only when innovative (i.e., linearly independent) information is received. Innovative packets may however be lost due to packet collisions, thus interrupting data propagation. Even worse, the insertion of innovative information into a given network area often causes all nodes in the area to attempt their new transmissions simultaneously which further increases the collision probability.

In reactive probabilistic network coding, nodes send out new combinations based on a forwarding factor ρ, which depends on their number of neighbors [18]. Setting ρ inversely proportional to the number of neighbors has the desirable effect that the number of innovative packets per area is independent of the local node density. We observe that there are particular topologies where this strategy does not work. As an example, think of the case where a given node t has a large number of neighbors and one of them, say node r, has only t as its neighbor. Due to its high number of neighbors (small ρ), t sends out a small number of packets and, in turn, r is unlikely to be able to decode all the wanted information (as it did not receive enough independent combinations from t). In contrast, ProNC does not require the reception of innovative information to continue data dissemination (so it is more robust to interference and collisions), and its performance does not depend on the forwarding factor ρ. It is based on two important components:

- a set of conditions to stop transmissions when all source packets have been delivered to all nodes, i.e., Stopping Conditions (SCs),
- a strategy to set the frequency at which new random packet combinations are to be sent so as to avoid network congestion. In the rest of the section we refer to this strategy as *Rate Adaptation* mechanism.

Basic rules for ProNC: each node can be in one of two different states: *active* and *inactive*. The basic idea of the proactive approach is that an *active* node periodically sends out a new packet combination to its neighbors, while an *inactive* node does not transmit. To switch from one state to the other, a node considers the following set of rules:

R1 *A node becomes active upon receiving the first innovative packet.* This means that a data dissemination phase is started and the node has to contribute to it.

R2 *A node becomes inactive when the SC is verified.* In this case further transmissions from this node are no longer useful for its neighbors and should be suppressed to avoid unnecessary overhead.

R3 *A node becomes active again when the SC no longer holds.* This last rule is particularly important as it allows propagation of new information into an area where all nodes are currently inactive.

Note that while a node is inactive, it can still receive packets from its neighbors. This information is used to assess whether the stopping condition still holds.

Stopping Conditions: there are different ways to define the SCs for Proactive Network Coding. They depend, in general, on the amount of information that each node has to collect in order to decide whether to suspend its transmissions. Our main aim is to keep the transmission overhead as low as possible. We identify two simple cases in which a node has to suspend its transmission: 1) In the first case, all neighbors of a node t have decoded all the packets they require and thus no further transmissions by t are necessary. 2) The second is when the subspace spanned by the information vectors (i.e., packets) available at node t is contained in the subspace spanned by the information vectors at each of the node's neighbors. In this case, t's packets will not be innovative for any of its neighbors and the node should suspend its transmission.

Based on these observations, we describe two different conditions which are referred to as Strong Stopping Conditions (SSCs) and Weak Stopping Conditions (WSCs). They define two different proactive schemes. According to the SSC, nodes send out beacons (Strong Stopping Messages (SSMs)) to their neighbors when they have decoded all the packets they are interested in. Each node collects SSMs from its neighborhood. When a node receives an SSM from each of its known neighbors, the SSC is verified and transmissions are stopped. We refer to this scheme as Strong Proactive Network Coding (Strong ProNC) as it requires strong assumptions on the data traffic. In order to send out SSMs, each node needs to know in advance how many packets it wants to collect. This fact implies that each node has full knowledge about the amount (and type) of data flowing over the network. Note that the collection of this information may not be feasible in practice.

The second strategy we describe is based on the WSC. During data propagation, each node sends out beacons (Weak Stopping Messages (WSMs)) containing a *decoding field* which is set to 1 if it can decode all packets in its buffer and to 0 otherwise. In addition, beacons contain a *rank field* specifying the rank of the nodes' decoding matrices. According to the WSC each node suspends its transmissions when all its neighbors can decode all the packets in their buffers and their matrices have the same (full) rank. We refer to this second strategy as Weak Proactive Network Coding (Weak ProNC) because it does not require any knowledge about the data traffic and has a limited overhead. However, Weak ProNC is suboptimal as there are some situations in which the rank alone does not capture the exact decoding status at different nodes. For instance, it might happen that all neighbors of a node can decode all the packets in their buffers and they all have the same rank but the decoded information is different.

3.5.1 Rate adaptation heuristics

We define τ as the time elapsed between the completion of the transmission of a packet by the PHY and the instant when the next packet is made available for transmission at the MAC, i.e., the idle time of the node. Note that τ is (roughly) inversely proportional to the transmission rate of the nodes. In what follows, we present an approximate model to find the value of τ that maximizes the amount of information that is transferred over the channel as a function of the system parameters. Note that the PHY layer data rate is kept constant. Methods to change the PHY rate are investigated in Section 3.6.

Impact of MAC layer dynamics: in what follows we derive the relationship between the value of $\tau_{\text{avg}} = E[\tau]$ that maximizes the throughput, referred to as $\tau^{\star}_{\text{avg}}$, and the number of neighbors at any given node, n_v. We note that from the network coding point of view, the value of $\tau^{\star}_{\text{avg}}$ will be slightly greater in order to allow that a node receive as much as possible innovative packets prior to transmission. This way it maximizes the probability that the generated packet will be innovative for the receiving nodes. We consider the packet transmission process in a given neighborhood of n_v+1 nodes making the following assumptions: A1) we neglect the channel propagation delay as for the considered system parameters it has a negligible impact on the throughput performance, A2) we assume that all packets involved in a collision are lost and A3) we assume that any transmitted packet is always successfully received by all nodes in the neighborhood unless it collides with another transmission. To obtain a rate adaptation heuristic we model the IEEE 802.11 broadcast communication pro-

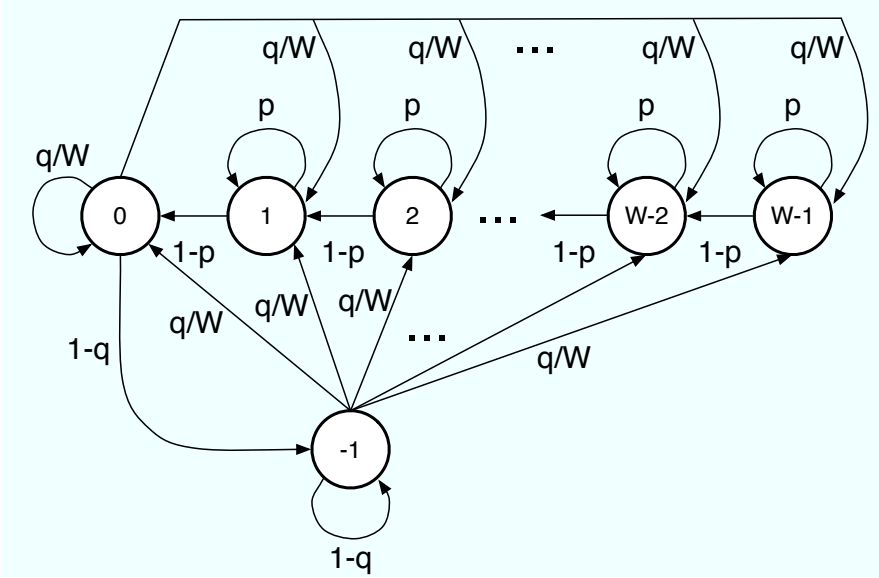


Figure 3.5. *Markov chain tracking the evolution of the IEEE 802.11 broadcast transmission process. For n_v nodes in a neighborhood, p is the probability that at least one of the remaining n_v nodes transmits when the target node is in slot $s = 1, \ldots, W-1$. $1-q$ is the probability that the transmission buffer is empty after the transmission of a given packet. W is the backoff window size of IEEE 802.11.*

cess. As in [33], the evolution of the transmission backoff counter is tracked using a suitable Markov chain. However, in our case the backoff window size W is always constant as packets are neither retransmitted nor acknowledged. This implies that the broadcast backoff process of any of the $n_v + 1$ nodes can be modeled through the one dimensional Markov chain of Figure 3.5 (a two dimensional chain was used in [33] for the IEEE 802.11 unicast case to take retransmissions into account). In addition, in order to model the transmission process in nonsaturated traffic conditions, we adopt the technique of [34] where idle transmission times (due to empty transmission queues) are modeled through the addition of the further state -1. With reference to Figure 3.5 our model works as follows. When a target node has a packet to transmit, it starts the backoff process by randomly selecting a backoff counter value from 0 to $W-1$ and then starting to decrement the counter until state 0 is reached. State $s = 0, 1, \ldots, W-1$ represents the current backoff counter value. The transition from state s to state $s-1$ occurs with probability $1-p$ after a backoff slot time (of fixed duration σ), while with probability p the process remains in state s. p is the probability that at least one of the remaining n_v nodes transmit when the target node is in state s. If this occurs, the node momentarily stops counting down its backoff timer. The current packet is finally transmitted when the backoff process reaches state 0 (*transmission state*). Upon the completion of

the packet transmission two events can occur: E1) with probability q the transmission queue is non-empty and a new backoff timer is uniformly selected at random in $\{0, 1, \ldots, W-1\}$: the probability that the system moves from state 0 to any state $s = 0, 1, \ldots, W-1$ is thus q/W. E2) With probability $1-q$ the transmission queue is empty and in this case the process moves to the idle state -1, where it remains until a new packet arrives (at which point the chain evolution is the same as in E1).

Let π_s, $s = -1, 0, 1, \ldots, W-1$ be the steady-state probabilities of the above Markov chain. Our goal is to find π_0 (the *transmission probability*) and relate it to τ_{avg}. From the chain regularities, and by computing recursively through the chain from right to left, we obtain:

$$\begin{aligned} \pi_{W-s} &= \frac{sq(\pi_0 + \pi_{-1})}{W(1-p)} \ , \ s = 1, 2, \ldots, W-1 \ , \\ \pi_0 &= (\pi_0 + \pi_{-1})q \ . \end{aligned} \tag{3.1}$$

From the normalization condition $\sum_{s=-1}^{W-1} \pi_s = 1$ we find:

$$\pi_0 + \pi_{-1} = \frac{1}{1 + \frac{(W-1)q}{2(1-p)}} \ , \tag{3.2}$$

from which we finally obtain π_0 as:

$$\pi_0 = \frac{q}{1 + \frac{(W-1)q}{2(1-p)}} \ . \tag{3.3}$$

The probability p that at least one of the remaining n_v nodes transmits when the target node is in slot $s = 1, \ldots, W-1$ is found as: $p = 1 - (1-\pi_0)^{n_v} \stackrel{def}{=} f_1(\pi_0)$. We additionally define the probability P_t that at least one node is transmitting in a given slot:

$$P_t = 1 - (1-\pi_0)^{n_v+1} \ , \tag{3.4}$$

and P_s as the probability that only one node is in the transmission state, conditioned on the fact that at least one node is transmitting. P_s is obtained as:

$$P_s = \frac{(n_v+1)\pi_0(1-\pi_0)^{n_v}}{P_t} = \frac{(n_v+1)\pi_0(1-\pi_0)^{n_v}}{1-(1-\pi_0)^{n_v+1}} \ . \tag{3.5}$$

We are now ready to calculate the normalized throughput S as (see [33]):

$$S = \frac{E[\text{payload bits successfully transmitted in a slot}]}{E[\text{slot length}]} \ . \tag{3.6}$$

Since a successful transmission occurs in a slot with probability $P_t P_s$, the average number of payload bits successfully transmitted in a slot time is $P_t P_s P$, where P is the payload size.

The average length of a slot can be obtained by considering the following three mutually exclusive cases: C1) with probability $1 - P_t$, none of the nodes transmit in the slot. The duration of an empty backoff slot is $\sigma = 20$ μs. C2) with probability $P_t P_s$, only one node transmits in the slot. The slot duration in this case is T_s, which is the transmission time of a packet, given by the sum of the time spent transmitting the physical header (T_{PHY}), the MAC header (T_{MAC}), the network coding header ($T_{\text{NC}}(n) = NC(n)/R$), the payload ($T_P = P/R$), plus the Distributed Inter Frame Space (DIFS) time, T_{DIFS}:

$$T_s = T_{\text{PHY}} + T_{\text{MAC}} + T_{\text{NC}}(n) + T_P + T_{\text{DIFS}} \; . \tag{3.7}$$

C3) With probability $P_t(1 - P_s)$, multiple nodes transmit in a slot time, leading to a collision event. The duration of this slot is also T_s because packets are not acknowledged and, therefore, the transmission period for successful and collided packets is the same. Hence, we have $E[\text{slot length}] = \sigma(1 - P_t) + P_t P_s T_s + P_t(1 - P_s)T_s$. These facts together with Equation 3.6 give:

$$S = \frac{P_t P_s P}{\sigma(1 - P_t) + P_t P_s T_s + P_t(1 - P_s)T_s} = \frac{P_s(P/\sigma)}{(1 - P_t)/P_t + T_s/\sigma} \; . \tag{3.8}$$

Note that the maximum throughput is achieved when the following function is maximized:

$$g(\pi_0) = \frac{P_s}{(1 - P_t)/P_t + T_s/\sigma} = \frac{(n_v + 1)\pi_0(1 - \pi_0)^{n_v}}{T_s/\sigma - (1 - \pi_0)^{n_v+1}(T_s/\sigma - 1)} \; . \tag{3.9}$$

The optimal transmission probability $\pi_0^\star$ can be found as (see calculations in Section VI of [33]):

$$\pi_0^\star = \underset{\pi_0}{\text{argmax}}\, g(\pi_0) = \frac{\sqrt{[(n_v + 1) + 2n_v(T_s/\sigma - 1)]/(n_v + 1)} - 1}{n_v(T_s/\sigma - 1)} \; . \tag{3.10}$$

The optimal $\tau_{\text{avg}}^\star$, is obtained from $\pi_0^\star$ as:

1. Express q as $q = 1 - e^{-E[\text{slot length}]/\tau_{\text{avg}}}$, which as shown in [34] provides a good approximation of the queue behavior in the unsaturated case. Inverting this relation gives:

$$\tau_{\text{avg}} = -\frac{E[\text{slot length}]}{\log(1 - q)} = -\frac{\sigma(1 - P_t) + P_t T_s}{\log(1 - q)} \; . \tag{3.11}$$

2. Obtain $\pi_0^\star$ from Equation 3.10.

3. Invert Equation 3.3 to find p as a function of π_0 as:

$$p = 1 - \frac{\pi_0(W - 1)}{2(1 - \pi_0/q)} \overset{def}{=} f_2(\pi_0, q) \tag{3.12}$$

and find $q^\star$ as the solution of $f_1(\pi_0^\star) - f_2(\pi_0^\star, q) = 0$, which leads to:

$$q^\star = \frac{2\pi_0^\star(1-\pi_0^\star)^{n_v}}{2(1-\pi_0^\star)^{n_v} + \pi_0^\star(1-W)} \,. \tag{3.13}$$

4. Obtain $\tau_{\text{avg}}^\star$ from Equation 3.11 setting $q \leftarrow q^\star$ and expanding $E[\text{slot length}]$ using Equation 3.4:

$$\tau_{\text{avg}}^\star = \frac{T_s - (1-\pi_0^\star)^{n_v+1}(T_s - \sigma)}{\log\left(\frac{2(1-\pi_0^\star)^{n_v} + \pi_0^\star(1-W)}{2(1-\pi_0^\star)^{n_v+1} + \pi_0^\star(1-W)}\right)}. \tag{3.14}$$

We observe that our model is accurate for sufficiently large n_v, i.e., $n_v \geq 4$. For smaller values it is however inaccurate because of several approximations made in the analysis, i.e., the expression of q [34], the independence of the busy channel probability p among subsequent access slots [33] and to the fact that in our derivation of the steady state probabilities we neglect the semi-Markov character of the process, i.e., that states 0 and $s \neq 0$ have different durations (a common simplification for the analysis of IEEE 802.11 throughput [33, 34]).

Implementation notes: in our implementation we pick $\tau_{\text{avg}} = \tau_{\text{avg}}^\star$ selecting τ uniformly in $[0, 2\tau_{\text{avg}}]$, which gives $E[\tau] = \tau_{\text{avg}}$. We obtain $\tau_{\text{avg}}^\star$ as a function of n_v from Equation 3.14 using $T_{\text{PHY}} = 192$ μs, $T_{\text{MAC}} = 224$ μs, $T_{\text{DIFS}} = 50$ μs, $\sigma = 20$ μs, $R = 1$ Mbps and $W = 32$ slots, which are used for IEEE 802.11b broadcast with a rate of 1 Mbps. We find that the relationship between the two is well approximated by a linear function, as predicted by the simulation results of [18, 28]. A good approximation is in fact given by the following heuristic: $\tau_{\text{avg}}^\star \simeq \kappa n_v T_s$, where $\kappa = 0.7$. We note that τ_{avg} corresponds to the average amount of time spent in state -1, i.e., to the time elapsed between the completion of the transmission of a packet by the PHY and the instant when a new packet is made available by ProNC. The inter-packet transmission time is greater than τ_{avg} as it also includes the time spent in backoff. Moreover, ProNC requires the estimation of the number of neighbors at each node which can be simply achieved by monitoring the source addresses of incoming packets. Note that both the stopping conditions and the packet rate adaptation mechanism depend on the node density. Most importantly, the number of nodes estimated in this way is smaller than that of the previous analysis. In fact, this number of neighbors only accounts for the nodes within transmission range, whereas n_v should include all nodes in the collision domain, whose range is always greater. Hence, the actual inter-packet transmission time τ'_{avg}, which accounts for all these facts, is $\tau'_{\text{avg}} = \kappa' n_v T_s$ where $\kappa' = 6$. We thus use this linear heuristic which gives good results across all simulations.

In addition, Stopping Messages (SMs) are included within data packets at the cost of a few extra bits. For SSM, we need one additional bit, whereas for WSM we need a bit to represent the decoding status and a byte to communicate the rank of the local decoding matrix. A single byte often suffices in practice, i.e., the number of nodes in the network that generate source packets is lower than or equal to 256. Coding over more source packets would imply the inversion of large matrices at the receiver which is impractical and difficult to obtain in realtime. In both cases, the additional overhead is acceptable. On the downside, when a node becomes inactive it must send out at least one SM to communicate its change of status and this packet may be useless for coding purposes.

We note that piggybacking control information within data packets has the beneficial effect of keeping channel congestion low. In addition, the added control information (SSMs and WSMs, rank, decoding status) is used to increase the efficiency of network coding schemes which, in turn, can further reduce the number of transmissions for a target performance level. These benefits are quantitatively verified below.

3.5.2 Simulation results

Next, we compare ProNC against the reactive probabilistic schemes proposed in [18]. The results that follow are for topologies where nodes are randomly placed within a fixed area in such a way that the topology is always connected, possibly through multi-hop paths. We consider several average node densities by varying the average number of neighbors, $n_v \in \{2, 4, 8, 16, 32\}$. For the MAC, we adopt the basic IEEE802.11b broadcast mode, accounting for channel errors and collisions. In the remainder of this chapter we will consider the probabilistic network coding of Section 3.3.1 (from now on referred to as "Reactive NC") as well as the adaptive network coding (referred to as "Adaptive NC") scheme of [18] where ρ is picked independently at each node as $\rho = c/(n_v + 1)$, where c is a suitable constant equal for all nodes [18]. In Figure 3.6 we show the tradeoff between failure probability, 1-PDR and PDD. We note that ProNC performs better in terms of data recovery; 1-PDR is at least one order of magnitude smaller for ProNC when $n_v \in \{8, 16\}$. For small n_v, i.e., $n_v \in \{2, 4\}$, we often obtain pathological topologies leading to deadlocks of the data dissemination when reactive protocols are used. ProNC efficiently deals with these topologies and alleviates the deadlock problem by offering better performance in terms of 1-PDR. On the downside, in these cases the dissemination of data in ProNC takes slightly longer due to the waiting periods of proactive schemes (see variable τ). The tradeoff concerning the protocol overhead

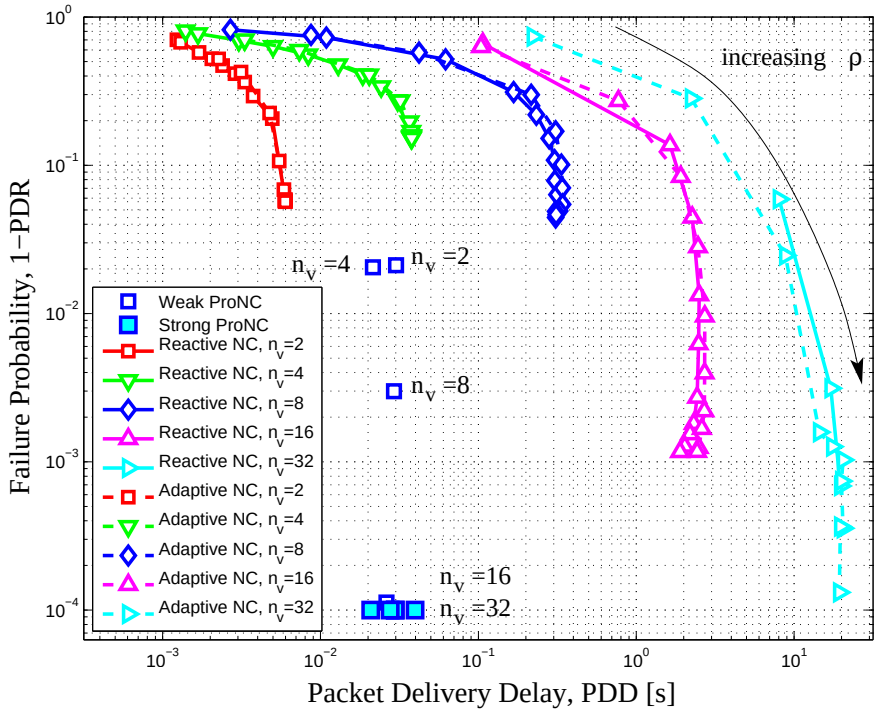


Figure 3.6. *Failure probability, 1-PDR vs. PDD: comparison between proactive and reactive schemes. The curves shown for reactive schemes are obtained for different values of $n_v \in \{2, 4, 8, 16, 32\}$, varying ρ as the independent parameter.*

is shown in Figure 3.7: similarly to reactive schemes, the overhead of ProNC increases with decreasing n_v. This is because network coding is more efficient when the node density is high. Also, the overhead of ProNC is usually smaller than that of reactive schemes, while it always outperforms reactive solutions in terms of data recovery performance. For both graphs Weak ProNC performs slightly worse than Strong ProNC in terms of PDR, whereas it performs better in terms of OH performance for the same n_v. The difference in performance is more significant at small densities, i.e., where deadlocks are more likely to occur. As demonstrated in [29], the OH performance of all schemes at high densities approaches that of an idealized scheme, having the minimum possible overhead: this reflects the fact that network coding works better when there are more coding opportunities. To summarize, both Strong ProNC and Weak ProNC show satisfactory performance in actual network settings. Weak ProNC is a distributed and self-adaptable dissemination protocol which does not require any knowledge about the traffic and only requires a few local interactions among nodes to work properly. Further improvements of Weak ProNC in terms of PDR are possible through the extension of the communication of control messages over multiple hops.

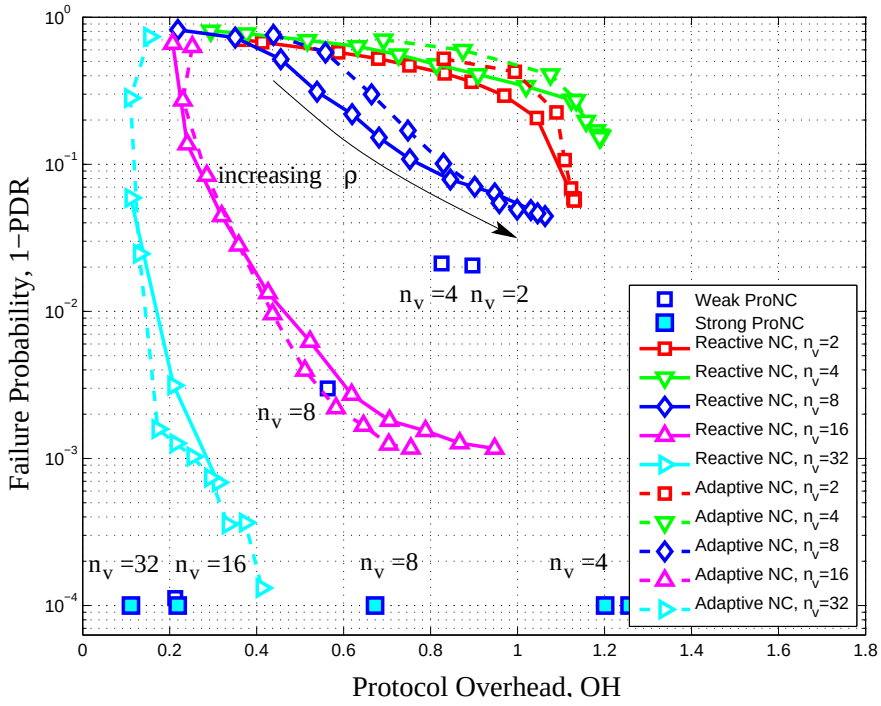


Figure 3.7. *Failure probability, 1-PDR vs. OH: comparison between proactive and reactive schemes. The curves shown for reactive schemes are obtained for different values of $n_v \in \{2, 4, 8, 16, 32\}$, varying ρ as the independent parameter.*

3.6 Extension to Multi-rate Ad Hoc Networks

When nodes have multi-rate transmission capabilities, it is necessary to pick a suitable PHY layer data rate, besides the selection of the forwarding factor, ρ. This is not an easy task. When nodes transmit at low data rates their coverage area is larger and in this case packets travel long hops. Conversely, transmitting with higher data rates leads to shorter transmission delays and shorter hops. Whichever is best depends on several factors such as network density and topology. The analysis in, e.g., [35] can be used to calculate the involved delay-throughput tradeoffs for different rates. The objective of the following paragraphs is to obtain good tradeoffs between PDR and PDD in IEEE 802.11g-based multi-rate scenarios when using reactive network coding.

As in Section 3.5.2, we consider a random wireless network where nodes want to disseminate and retrieve information through multi-rate network coding. We investigate the cooperation between IEEE 802.11g PHY/MAC and reactive network coding in this random scenario. We present results obtained through ns2 simulations with varying forwarding factor ρ for reactive network coding. As per our discussion above, ρ is the probability of

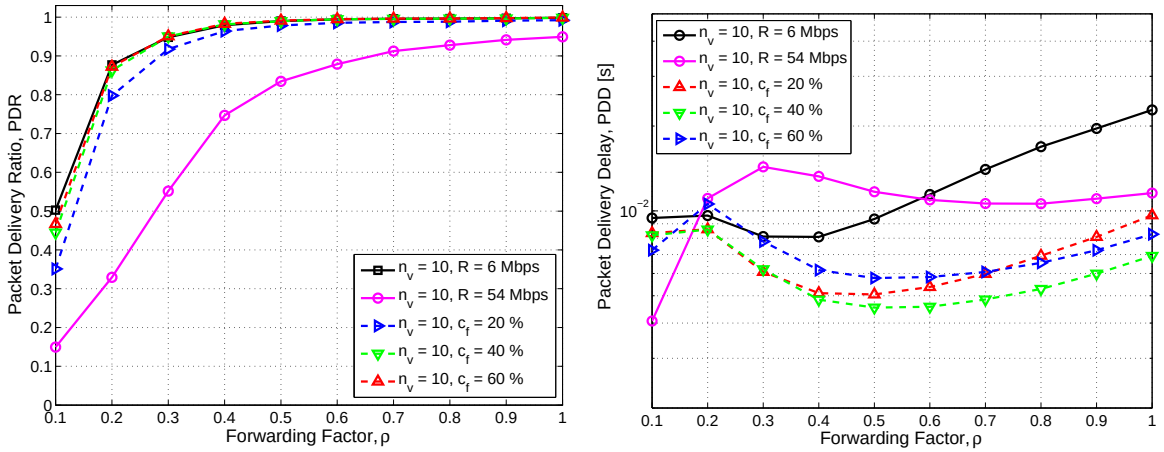
sending a new combination whenever a node receives an innovative packet and directly determines the amount of traffic which flows through the wireless network. Note that ProNC can be applied to a multi-rate transmission scenario as well and, for this case, we found similar advantages as those presented in Section 3.5.2.

3.6.1 Rate adaptation heuristic

In the following, we present a data rate adaptation heuristic which tries to achieve, at the same time, a short PDD and a high PDR. We assume that wireless nodes initially have no knowledge about the network status, i.e., they are completely unaware of location and number of neighbors as well as the transmission opportunities in their neighborhood. In order to acquire this knowledge the algorithm uses internal variables, at the MAC layer, to store auxiliary pieces of information such as node addresses, SINR, as well as data rates.

Initially, all nodes begin their transmissions with the lowest available rate (which is 6 Mbps for IEEE 802.11g). This is the best choice in terms of neighbor discovery as it allows nodes to collect information from a larger area. The subsequent reception of packets permits the gathering of useful information which will determine the data rate at this node, as we explain next. Each time a node, say node r, receives a new packet from one of its neighbors t, it extracts the following information: 1) `id`: the address of the sending node, 2) L: the size of the received packet, 3) γ_t: the instantaneous SINR associated with this packet reception. Upon reception, an average SINR, $\overline{\gamma}_t$, is updated for each neighbor t according to a discrete time first order low pass filter as: $\overline{\gamma}_t = \alpha\gamma_t + (1-\alpha)\overline{\gamma}_t$, where $\alpha \in (0,1)$ is the smoothing factor. For our simulations we picked $\alpha = 0.5$. In this way we take into account the variations in the received SINRs, while trying to capture its average value. The packet transmission length L and $\overline{\gamma}_t$ are thus used to estimate the PER for neighbor t for all transmission rates.

In this way, each node estimates the status of its neighbors in terms of associated SINR. Once node r has this information it updates its internal variables. Subsequently, it considers the fraction c_f of the nodes in its neighboring set that have the highest $\overline{\gamma}$s. c_f is referred to here as *coverage factor*. The data rate at a given node r is thus selected such that all of these nodes will receive packets from node r with a small packet error probability, i.e., smaller than a given threshold P_{th}. In the following results we selected $P_{th} = 0.03$ as it gave good results across all our experiments.



(a) Packet Delivery Ratio, PDR

(b) Packet Delivery Delay, PDD

Figure 3.8. *Dissemination performance for a multi-rate ad hoc network as a function of ρ.*

3.6.2 Simulation results

Figure 3.8(a) shows PDR as a function of ρ for different coverage factors c_f. For this figure, nodes have on average $n_v = 10$ neighbors, solid curves represent reactive network coding with fixed data rate (only the extreme cases of 6 and 54 Mbps are plotted), whereas dotted curves represent the reliability when nodes are allowed to adapt their data rate based on the above heuristic. First of all, we observe that transmitting with the highest rate of 54 Mbps performs the worst. This is because in such a case the error probability is quite high and this affects the overall performance. As expected, a fixed rate of 6 Mbps leads to the best reliability performance. However, setting $c_f = 40\%$ gets very close to this performance while leading to a shorter PDD, as we can see from Figure 3.8(b). From this plot we can further appreciate the benefits of adapting the data rate: overall, for a coverage factor of $c_f = 40\%$ we get the shortest delays by performing, in terms of reliability, almost as well as the fixed rate scheme with 6 Mbps. For the non monotonic behavior of the curves of Figure 3.8, note that when ρ is low, e.g., $\rho = 0.1$, the dissemination of innovative information terminates early (in this case deadlocks are frequent) and a large number of nodes are unable to decode all packets (PDR is low, see Figure 3.8(a)). An increasing ρ allows the dissemination process to continue longer, leading to fewer deadlocks, thus the PDD increases with a corresponding increase in PDR. A further increase of ρ at first allows a quicker dissemination of innovative information (shorter delays), and afterwards the PDD increases again due to the increased traffic load (collisions). Figure 3.9 shows tradeoff results representing the failure probability

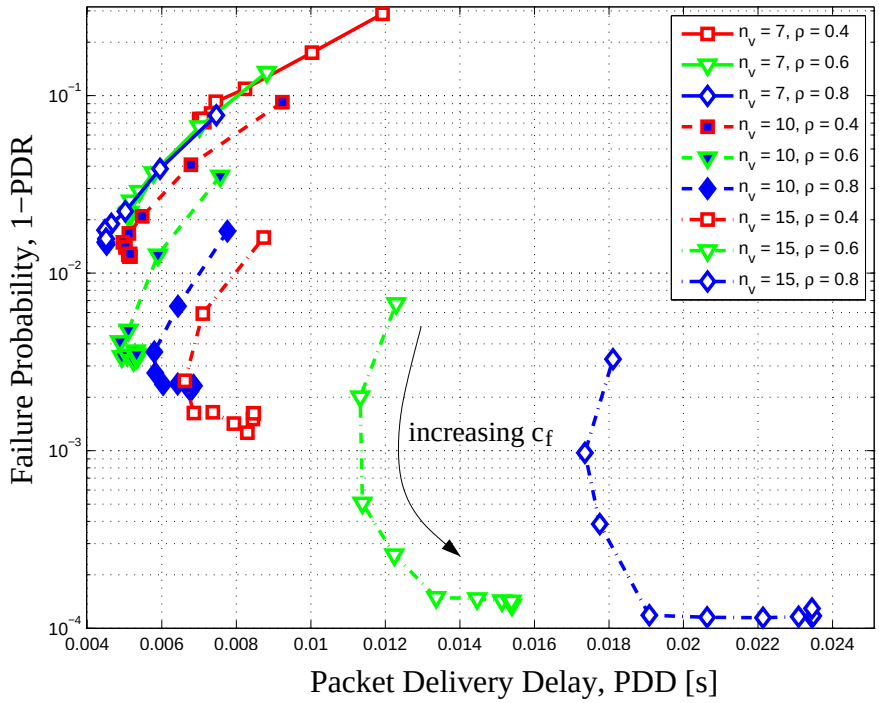


Figure 3.9. *Tradeoff between Failure Probability 1-PDR and PDD for a multi-rate ad hoc network with different network densities. Different curves correspond to different forwarding factors ρ. Each curve is plotted for varying c_f.*

1-PDR as a function of PDD for different number of neighbors $n_v \in \{7, 10, 15\}$. Simulation curves are plotted for different values of ρ and each curve is obtained by varying c_f. Good schemes are those lying on the bottom left of the plot (i.e., having short PDD and high PDR). From this graph we see that for each (n_v, ρ) pair there exists a suitable coverage factor c_f which minimizes the PDD while achieving good performance in term of 1-PDR. In practice, c_f in the range $[30, 50]\%$ are good choices as they give good reliability performance while ensuring short delays for all reasonable network sizes. As usual, various tradeoffs can be obtained for different forwarding factors ρ: high ρ values always lead to good PDR performance at the cost of additional delay, while a small ρ is a good choice in terms of delay performance at the cost of an increased failure probability.

4

Broadcasting in Multi Channel Wireless Networks

Multi channel wireless networks enable multiple parallel transmissions on orthogonal frequency bands, leading to a more efficient utilization of spectrum resources than their single channel counterparts. The use of multiple channels provides increased throughput and robustness to interference generated by other users. As a consequence, the use of multiple parallel channels is expected to bring significant benefits to wireless ad hoc, sensor and cognitive radio networks. In such a multi channel system, nodes need to coordinate in order to efficiently share the available wireless resources. Hence, it is important to design a robust dissemination protocol for broadcasting to enable nodes in exchanging the required information for coordination.

ONE of the challenges in designing and operating multi channel wireless networks is the coordination of the nodes operating in the system. Some strategy or control rules must be in place in order for nodes to exchange control information. One important building block in this exchange of information is the ability of nodes to broadcast information to all other users in their neighborhood. Applications for this type

The material presented in this chapter has been published in [36].

of broadcast include dissemination of routing information, information about availability of spectrum, or neighbor discovery.

One first approach is to select a single channel for broadcasting purposes and use one of the techniques discussed in Chapter 3. This approach suffers from several drawbacks, including: (i) this strategy eliminates the possibility that information broadcast benefits from the use of multiple channels if the single chosen channel becomes congested; (ii) this solution cannot be used in opportunistic cognitive networking scenarios where the availability of a given channel cannot be guaranteed a priori [37]; (iii) if the chosen channel experiences high levels of interference, the system performance may degrade; and (iv) it is easy for an adversary to jam a single control channel.

A second approach is to simply designate a fixed number of channels, c, to be used for broadcasting purposes. With this solution, nodes in the system must move between the c channels to disseminate their information to all other nodes over time. While this solution may be more robust than using a single control channel, it still suffers from some drawbacks: (i) if c is chosen to be too large, random encounters of nodes on a common channel may become too infrequent, thus requiring a long time for information to be received by all nodes; (ii) if c is chosen to be too small, the channels may become congested; or (iii) it may be possible for a small number of adversaries to effectively jam a small number of control channels. In fact, as we show below, using simple mechanisms for disseminating information over c channels can often lead to very high dissemination delays.

This motivates our problem: the design of a robust dissemination protocol for broadcasting information within a neighborhood in a multi channel wireless system, and tuning the parameters in the system for optimal performance. We consider cases in which there are no adversaries, and cases in which there are adversaries with various capabilities of jamming the channels. Hence: 1) We provide a framework to analyze the performance of network coding and other protocols for broadcast; 2) We show that the broadcasting protocols require access to only a subset of the available channels to achieve minimum dissemination delay under different types of attacks; 3) We derive such optimum number of channels for all broadcasting protocols under investigation; and 4) We explore the impact of incorrect estimation of the number of adversaries on the dissemination delay.

The rest of this chapter is organized as follows. Section 4.1 discusses existing adversary avoidance techniques. Section 4.2 introduces the network model and the adversary models, and discusses the broadcasting protocols. In Section 4.3 we model the dissemination process

as a coupon collector's problem and find the optimum number of channels that minimizes the dissemination delay in different adversary scenarios. In Section 4.4 we present simulation results validating our analysis and discuss the impact of incorrect estimation of the number of adversaries.

4.1 Existing Adversary Avoidance Techniques

Traditionally, Spread Spectrum-based techniques, such as Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), have been used to mitigate adversary attacks [38, 39]. Intuitively, such robustness is achieved by the transmitter which spreads the information (represented as a narrow band signal) in a wide spectrum band where it is practically impossible for the adversary to disrupt the communication in all its frequency components. In order for the receiver(s) to correctly retrieve such information, it is required that all the communication participants share a common key that identifies how the spreading has been performed [38]. This common key has to be kept secret among the participants as its knowledge is sufficient for a malicious adversary to disrupt any communication attempt [39].

In order to eliminate the dependency on such pre-shared keys there have been several proposals which introduce the concept of Uncoordinated Spread Spectrum (USS) techniques [40–47]. These were first introduced in the context of neighbor discovery [41] and successively used as an efficient method for establishing a common secret key between two nodes [42–44]. Once the secret key is exchanged between these nodes, it can be used for future FHSS communications until the adversary succeeds in identifying it.

Given their robustness to adversary attacks, variants of USS have been used as a basis to provide data broadcasting [40, 45–47] in wireless networks. The key idea is to use a pool of publicly available spreading codes which are going to be used by the nodes every time they have to transmit packets. Hence, by selecting a random spreading code among those available it is possible to reduce the impact of the adversary attacks [46]. In order to further improve the robustness against reactive adversaries, Liu et al. [45] use the correlation of unpredictable spreading codes to encode each bit of data, making it very difficult for adversaries to find the correct spreading code.

We note that these protocols perform broadcasting by allowing nodes to repeatedly transmit multiple copies of their packets. Hence, by using redundancy in the packet trans-

mission it is possible for the receivers to retrieve all packets. However, as shown in [46], increasing the dimension of the set of the spreading codes (which can be interpreted as an increase of the number of channels nodes can select to transmit) also increases the time required for the nodes to retrieve all packets, i.e., the dissemination delay. Such increase becomes essential in those scenarios where all the nodes in the network are interested in exchanging their packets. In this case, even under optimum operation (no adversaries and single spreading code [46]) the dissemination delay grows as $O(n \log n)$ [48] where n is the number of nodes.

4.2 Model, Protocols and Definitions

Let the network consist of a set of nodes $\mathcal{N} = \{1, \ldots, n\}$, assumed to be within mutual transmission range. Let the number of available non-overlapping channels be c. Each node is equipped with a single half duplex transceiver. We further assume a time slotted system for communications where nodes are synchronized at the slot level. Hence, in a given slot, if a packet is successfully transmitted in a channel it will be received by all co-channel nodes, i.e., nodes whose radios are tuned to that particular channel.

We present details on the adversary model in Section 4.2.1. Broadcasting protocols are examined in Section 4.2.2 and Section 4.2.3 summarizes the performance metrics.

4.2.1 Adversary model

The wireless medium is subject to attacks launched by adversaries which transmit jamming signals, thereby destroying any communication in a given channel and slot. We consider two types of adversaries - *random* and *colluding*. Random adversaries hop independently and randomly over all channels. Colluding adversaries are more effective as they negotiate which set of channels to access to avoid the occurrence of multiple jammers in the same channel.

Let m_r and m_c be the number of random and colluding adversaries, respectively. The probability that a channel is jammed in a given slot by random adversaries or a colluding adversary is:

$$\mu^r = 1 - \left(1 - \frac{1}{c}\right)^{m_r} \text{ and } \mu^c = \begin{cases} \frac{m_c}{c} & \text{if } m_c \leq c, \\ 1 & \text{if } m_c > c, \end{cases} \tag{4.1}$$

respectively. Assuming that the random adversaries and the colluding adversaries operate independently, the probability that a channel is not jammed is:

$$\lambda \quad = \quad (1-\mu^r)\cdot(1-\mu^c). \tag{4.2}$$

4.2.2 Broadcasting protocols

As in the previous chapter, we focus on the broadcasting problem where each node $r \in \mathcal{N}$ generates a source packet x_r and aims at gathering the packets of all the other nodes in the network. The problem can be generalized for any number of source packets generated by only a subset of the nodes accounting for the time required to disseminate such packets to the rest of the nodes.

In order to compare 1) Network Coding NC(2^q) - node transmits random linear combinations over GF(2^q) of all the received packets, we also consider the following broadcasting protocols:

2. Random Message Selection (RMS) - node t randomly transmits one of the packets it has received so far (including its own).

3. SeLF message replication (SLF) - node t always transmits only its own packet, x_t.

4. ConcATenation (CAT) - This is an idealized scheme where every node t concatenates and transmits all the packets it has received so far.

We note that, unlike the first three schemes which are practically implementable and can be actually used (examples of protocols using such schemes for packet forwarding can be found both along this thesis and in other works such as e.g. [41, 46, 48, 49]), CAT is considered here for the purpose of providing a performance bound only. This is because a practical implementation of CAT would require variable-length transmissions, whose duration depends on the size of the buffer. A study of this more realistic implementation involves several difficulties, including issues related to slot synchronization among nodes and to an accurate analysis of jamming attacks (whose attacking efficiency is higher as the duration of the transmission increases). However, our purpose in considering CAT is to provide an optimal benchmark scheme to evaluate a performance bound rather than proposing an implementable scheme. Hence, in order to keep the evaluation simple by avoiding these difficulties, we assume somewhat unrealistically that the transmission time of the whole buffer

content is always equal to a single packet duration, which in terms of dissemination delay does better than any practically implementable scheme.

4.2.3 Definitions and performance metrics

We hereby introduce the metrics which will be used for the performance evaluation of the broadcasting protocols.

Definition 4.1. *Dissemination Delay, D: the average number of slots required for a generic node to receive and decode all the source packets.*

Definition 4.2. *Optimum Number of Channels, $c^\star$: the number of channels nodes have to access to minimize D given the number of random and colluding adversaries, i.e.:*

$$c^\star = \underset{c\in\{1,\cdots,C\}}{\arg\min}\ \mathrm{D}. \tag{4.3}$$

where C is the maximum number of available channels.

Definition 4.3. *Operational Well, $\Omega(d)$: the interval of the values of c delimited by:*

$$\begin{cases} c^- = \underset{c\le c^\star}{\arg\min}\ |D - D^\star \cdot 10^{\frac{d}{20}}| \\ c^+ = \underset{c\ge c^\star}{\arg\min}\ |D - D^\star \cdot 10^{\frac{d}{20}}|. \end{cases} \tag{4.4}$$

where $D^\star$ is the minimum dissemination delay.

In words, the operational well defines the admissible number of channels nodes can operate in to ensure that the obtainable dissemination delay degradation is less than or equal to $d/2$ dB. As an example, if nodes pick c inside the operational well $\Omega(1)$, this means that the dissemination process, on average, will be at most 12% slower than in the optimum case when nodes pick precisely $c^\star$.

Definition 4.4. *Dissemination Delay Deterioration, Φ: the ratio between the dissemination delay obtained by the protocols in the same operating conditions given two protocols, A and B, i.e.:*

$$\Phi = \frac{\mathrm{D}_A}{\mathrm{D}_B}. \tag{4.5}$$

Given that the CAT scheme represents a lower bound in terms of dissemination delay, in the following all the above mentioned schemes will be compared to such lower bound, i.e., B = CAT unless otherwise specified.

4.3 Analysis

In this section, we briefly describe the coupon collector's problem and analyze the two MAC protocols that can be employed in a multi channel scenario. We subsequently use the obtained results to determine the dissemination delay for NC(2^q), RMS, SLF (CAT performance is obtained via simulations) using the coupon collector's approach. Finally, we derive the optimum number of channels that nodes need to access to minimize the dissemination delay for each broadcasting protocol.

4.3.1 The coupon collector's problem

The coupon collector's problem has been extensively studied and applied in several fields to solve practical problems [50]. In its classical formulation this problem is as follows. A collector wishes to collect the complete set of n distinct coupons. Each of these coupons is hidden inside breakfast cereal boxes and he is equally likely to find any of the n coupons in a cereal box. The collector purchases one box of cereals at a time and collects the coupons until all the n distinct coupons have been gathered. The problem is to find the total number of cereal boxes Y_n that the collector has to buy in order to collect all the n coupons which is:

$$E[Y_n] = n \sum_{i=1}^{n} \frac{1}{i} = nH(n). \tag{4.6}$$

where $H(n)$ denotes the n-th Harmonic number.

4.3.2 MAC protocols

CSMA-based MAC protocols have been employed to provide a distributed and dynamic access mechanism. In contrast, Slotted ALOHA-based protocols provide a simple protocol to access the medium. We are interested in determining the receiving probability for a CSMA variant and for Slotted ALOHA.

Definition 4.5. *The receiving probability ρ_r is defined as the probability that a given node successfully receives a packet in a given slot and any channel.*

Slotted CSMA (SCSMA): nodes sense the carrier prior to transmission in order to avoid collisions. This can be achieved by allocating a back-off window at the beginning of every slot. Hence, assuming perfect operation, at most one node may transmit in a slot while the

remaining nodes in the same channel listen. The probability that a given channel is busy, i.e., there is at least one node in the channel, is:

$$\nu = 1 - \left(1 - \frac{1}{c}\right)^n. \tag{4.7}$$

where $1/c$ is the probability that a node selects a specific channel.

We note that, in a slot, there are on average $c\nu$ nodes transmitting in the system. Such transmissions can be either useless (the transmitter is the only node in the channel) or useful (there are other nodes which are listening in the channel). Given that in the system there are only transmitters and receivers, i.e., no collisions, the average number of receivers is $n - c\nu$. A given node correctly receives a packet in a slot if it is one of the receiving nodes. Hence, the receiving probability can be calculated as:

$$\rho_r = \frac{n - c\nu}{n} = 1 - \frac{c\nu}{n}. \tag{4.8}$$

Slotted ALOHA (SALOHA): as for the previous MAC, time is divided in slots and nodes follow a random hopping pattern to access the available channels. A node transmits with probability p_t and listens otherwise. Accordingly, in a given channel and slot a packet can be correctly received if there is a single transmission. Unlike SCSMA, nodes in SALOHA do not sense the channel in order to avoid collisions. Hence, in a given slot a node transmits with probability $p_t = 1/n$. In order to determine the receiving probability for SALOHA we note the following. The probability that in the same channel as that selected by the receiving node, there are $k - 1$ other nodes (i.e., overall k nodes) is:

$$p(k) = \binom{n-1}{k-1}\left(\frac{1}{c}\right)^{k-1}\left(1 - \frac{1}{c}\right)^{n-k}. \tag{4.9}$$

In this configuration, a given node correctly receives a packet if (i) it does not transmit, and (ii) only one of the $k - 1$ remaining nodes in the channel transmits. Averaging over all configurations the receiving probability is:

$$\rho_r = \sum_{k=2}^{n} p(k)(k-1)p_t(1-p_t)^{k-1} = \frac{(n-1)p_t(1-p_t)}{c}\left(1 - \frac{p_t}{c}\right)^{n-2}. \tag{4.10}$$

4.3.3 Broadcasting protocols

In this section we derive the dissemination delay that a node experiences to retrieve all the source packets generated in the network using different broadcasting protocols.

Let $\mathcal{S}^\star$ be the subspace generated by applying a linear transformation to the subspace of all the source packets. Depending on the broadcasting protocol, the applied linear transformation is going to be different. Moreover, let $\mathcal{S}^t, \mathcal{S}^r \subseteq \mathcal{S}^\star$ be the subspaces observed in a given slot by the transmitter t and the receiver r, respectively.

Definition 4.6. *A packet $\underline{\ell} \in \mathcal{S}^t$ is said to be innovative for node r if its reception increases the dimension of the subspace $\mathcal{S}^r$.*

We divide time into epochs where epoch i begins when node r receives the ith *innovative* packet and ends when it receives the $i+1$th *innovative* packet. We denote the duration of epoch i by T_i. In order to calculate the average delay for each broadcasting protocol we proceed as follows. Assume that node r has already received i innovative packets, i.e., it is in epoch i. We note that, in slot j, node r correctly receives a packet $\underline{\ell}$ if it is in a channel where only one node $t \neq r$ is transmitting. This occurs with probability ρ_r (calculated from Equation 4.8 or Equation 4.10 according to which MAC protocol nodes are using). Moreover, the transmission is successful only if the channel is jammer free, which occurs with probability λ (calculated according to Equation 4.2). The probability that a packet received in slot j of epoch i is innovative can be expressed as:

$$\beta_{i,j} = \sum_{t \in \mathcal{N} \setminus \{r\}} \Pr(t) \cdot \sum_{\underline{\ell} \in \mathcal{S}^t_{i,j}} \Pr(\underline{\ell}) \cdot \Pr[\underline{\ell} \notin \mathcal{S}^r_{i,j}]. \tag{4.11}$$

where $\mathcal{S}^r_{i,j}$, $\mathcal{S}^t_{i,j}$ are the subspaces observed at slot j of epoch i by the receiver r and the transmitter t, $\Pr(t)$ is the probability of picking transmitter t, and $\Pr(\underline{\ell})$ is the probability of picking packet $\underline{\ell}$ in $\mathcal{S}^t_{i,j}$.

Assuming for simplicity $\beta_{i,j} = \beta_i$ (i.e.the innovative probability does not change within epoch i), the overall probability of receiving an innovative packet in any slot j of epoch i is:

$$\pi_i = \lambda \rho_r \beta_i. \tag{4.12}$$

The probability that such an innovative packet is received at the jth slot of epoch i is $\pi_{i,j} = \pi_i(1-\pi_i)^{j-1}$, $j = 1, 2, \ldots$. Hence, the expected duration of epoch i can be calculated as:

$$E[T_i] = \sum_{j=1}^{+\infty} j\pi_{i,j} = \sum_{j=1}^{+\infty} j\pi_i(1-\pi_i)^{j-1} = \frac{1}{\pi_i}. \tag{4.13}$$

At this point we can calculate the average time required to receive up to the kth innovative packet by summing the contributions of each epoch:

$$\mathrm{D}_k = \sum_{i=1}^{k-1} E[T_i] = \sum_{i=1}^{k-1} \frac{1}{\pi_i} = \sum_{i=1}^{k-1} \frac{1}{\lambda \rho_r \beta_i} \tag{4.14}$$

which, for $k = n$, gives the average dissemination delay:

$$\mathrm{D} = \frac{1}{\lambda\rho_r}\sum_{i=1}^{n-1}\frac{1}{\beta_i} = \frac{\Sigma^n}{\lambda\rho_r}. \quad (4.15)$$

where $\Sigma^n = \sum_{i=1}^{n-1}\frac{1}{\beta_i}$.

Network Coding (NC(2^q)): according to network coding, a node t forwards random linear combinations of all the received packets. In this case $\mathcal{S}^\star$ is the Galois subspace spanned by all the coded packets (equivalently code vectors) generated in the network whose cardinality is $\|\mathcal{S}^\star\| = 2^{qn}$. Let $\mathcal{S}_i^t \subseteq \mathcal{S}^\star$ be the subspace spanned by the code vectors available at the transmitting node t and $\mathcal{S}_i^r \subseteq \mathcal{S}^\star$ the subspace spanned by the code vectors available at the receiving node r in epoch i (i.e., its cardinality is $\|\mathcal{S}_i^r\| = 2^{qi}$). According to Equation 4.11 we express the probability that packet $\underline{\ell}$ is innovative as:

$$\begin{aligned}\beta_i &= \sum_{t\in\mathcal{N}\setminus\{r\}}\Pr(t)\cdot\sum_{\underline{\ell}\in\mathcal{S}_i^t}\Pr(\underline{\ell})\cdot\Pr[\underline{\ell}\notin\mathcal{S}_i^r] \\ &= \sum_{\underline{\ell}\in\mathcal{S}_i^t}\Pr(\underline{\ell})\cdot\Pr[\underline{\ell}\notin\mathcal{S}_i^r] \approx \frac{2^{q(n-1)}-2^{q(i-1)}}{2^{q(n-1)}}. \quad (4.16)\end{aligned}$$

where we assume that the buffers' statistics of all transmitters are the same. Moreover, in the last approximation, we assume that the buffers' content of the transmitter and the receiver are statistically independent. This approximation is very good when the number of channels is large (increasing the number of channels leads to an increase in packet diversity inside the nodes' buffers). However, in those cases where the number of channels is low it is very difficult to derive an analytical expression for Equation 4.16. In order to precisely quantify β_i we obtain it via simulations (β_i^{NC}). Hence, the dissemination delay becomes:

$$\mathrm{D} = \frac{1}{\lambda\rho_r}\sum_{i=1}^{n-1}\frac{1}{\beta_i^{\mathrm{NC}}}. \quad (4.17)$$

We note that as long as the node has not received all n innovative packets it will not be able to retrieve the information contained in them, i.e., the n source packets. However, partial decoding may still be possible depending on the Galois Field (GF) size used to encode the packets. We leave this topic as future work.

Random Message Selection (RMS): according to this scheme, a node forwards a randomly selected packet from the set of all the innovative packets it has received so far, including its own packet. In contrast to NC(2^q), with RMS (and SLF) nodes are enabled to transmit source packets only, instead of their linear combinations. Hence, $\mathcal{S}$, from now on, instead

of a subspace will denote a subset of packets. Accordingly, $\mathcal{S}_i^t$ for a transmitter t consists of all the innovative packets it has received up to epoch i. The probability of a packet being innovative can be expressed as:

$$\beta_i = \sum_{\underline{\ell} \in \mathcal{S}_i^t} \Pr(\underline{\ell}) \cdot \Pr[\underline{\ell} \notin \mathcal{S}_i^r] = \sum_{\underline{\ell} \in \mathcal{S}_i^t} \frac{1}{i} \Pr[\underline{\ell} \notin \mathcal{S}_i^r]. \quad (4.18)$$

As in NC(2^q), also for RMS, $\Pr[\underline{\ell} \notin \mathcal{S}_i^r]$ highly depends on the correlation factor between the transmitter's and receiver's buffers. Hence, also in this case we obtain β_i via simulation (β_i^{RMS}) which leads to:

$$\mathrm{D} = \frac{1}{\lambda \rho_r} \sum_{i=1}^{n-1} \frac{1}{\beta_i^{\text{RMS}}}. \quad (4.19)$$

SeLF message replication (SLF): a node t forwards only its own packet, $\mathcal{S}_t = \{t\}, \forall t \in \{1, \ldots, n\}$. Hence, from the point of view of the receiving node r, the events of receiving the $(i+1)$th innovative packet (end of epoch i) and meeting a node t that it has not met before are the same. In this case $\beta_i = (n-i)/(n-1)$ which gives:

$$\mathrm{D} = \frac{1}{\lambda \rho_r} \sum_{i=1}^{n-1} \frac{1}{\beta_i} = \frac{(n-1) \cdot H(n-1)}{\lambda \rho_r}. \quad (4.20)$$

where $H(n-1)$ denotes the $(n-1)$-th Harmonic number.

4.3.4 Optimum channel selection

We derive the optimum number of channels, $c^\star$, that nodes have to access in order to achieve minimum dissemination delay in different network configurations. Intuitively, if the number of channels used is too small, nodes will have to wait for a long time before they can access the channel to disseminate their information and adversaries will have a better chance to jam the channels on which transmissions are taking place. If the number of channels used is too large, nodes may not be on the same channel concurrently at most times, once again increasing dissemination delay. Thus, the correct determination of $c^\star$ is important to the overall performance of the system.

In the previous subsection we obtained the dissemination delay as a function of the number of nodes, n, number of channels c and adversaries, m_r and m_c, operating in the area for all the broadcasting protocols under examination. According to the general expression Equation 4.15 we have that three factors contribute to the dissemination delay:

- Σ^n - the cumulative sum of the inverse of the innovative probability, which quantifies the capability that a given broadcasting protocol has to deliver innovative information with each transmitted packet,
- ρ_r - the receiving probability, which depends on the MAC protocol that nodes are using to access the medium,
- λ - the probability of a channel not being jammed, which depends exclusively on the adversary attacks.

Hence, for a given number of nodes n that are using a certain MAC and broadcasting protocol to disseminate packets in the presence of adversary attacks (m_r random and m_c colluding adversaries), we have that:

$$c^\star = \underset{c\in\{1,\cdots,C\}}{\arg\min}\ \mathrm{D}(n, c, m_r, m_c). \tag{4.21}$$

We note that, from the point of view of the optimum channel selection, adversaries can behave in two different ways:

- *oblivious* - the adversaries are unaware of the optimization process that is undertaken by the nodes and their corresponding decision regarding $c^\star$. In this scenario the adversaries, both random and colluding, continue to randomly hop over all channels, c.
- *aware* - the adversaries are aware of the selected $c^\star$ from the nodes. This is achievable by having the adversaries extrapolate any control information nodes exchange along with their packets. Hence, they can maximize the effectiveness of their attack by accessing the set of channels specified along with $c^\star$.

As can be noticed, in the long term it is important to differentiate among the two distinct behaviors as they significantly impact the dissemination delay performance. That is, solving the minimization problem must take into account such distinction. More precisely, for oblivious adversaries we have that:

$$c^\star = \underset{c\in\{1,\cdots,C\}}{\arg\min}\ \frac{\Sigma^n}{\rho_r}, \tag{4.22}$$

as in this case λ will remain unchanged after $c^\star$ has been selected. Hence for oblivious adversaries, from the point of view of determining $c^\star$, the problem is in principle the same as for the case of no adversaries in the area. This is because the packet losses due to the attacks

of these adversaries in a given channel do not depend on the actual subset of channels that nodes select to access.

For aware adversaries, we must account for the fact that, after the optimum channel selection, these adversaries will access the same set of channels, hence increasing significantly the impact of their attacks. Hence:

$$c^{\star} = \underset{c\in\{1,\cdots,C\}}{\arg\min} \frac{\Sigma^n}{\lambda\rho_r}. \tag{4.23}$$

In the following, when discussing adversary attacks, we will assume them to be aware of the nodes' decisions. Results for oblivious adversaries are qualitatively the same as in the case of adversary-free networks.

4.4 Performance Evaluation

In this section we discuss the performance of the broadcasting protocols in different settings. The performance evaluation that we carry out is based on simulations using Matlab© where all the results are averaged over 500 trials. We first discuss the performance of broadcasting in a multi channel network with a pre-configured number of channels. We then obtain the optimum number of channels and the respective operational well for all the protocols.

4.4.1 Dissemination delay in multi channel networks

In Figure 4.1 we plot the dissemination delay versus the number of channels for the case of no adversaries. As expected, using SCSMA instead of SALOHA as the underlying MAC protocol significantly reduces the delay for all protocols. This is due to the inherent collision avoidance mechanism of SCSMA which eliminates any packet loss due to simultaneous access to the same channel. In other words, the receiving probability ρ_r is significantly higher for SCSMA than for SALOHA.

Focusing on SCSMA (the discussion is equivalent for SALOHA), we notice that the broadcasting protocols exhibit different behaviors depending on the number of channels. While for SLF increasing the number of channels always leads to an increase of the dissemination delay, the remaining protocols (including NC(2^8)) initially reduce the dissemination delay until they reach a global minimum for a given number of channels $c^{\star}$. This reduction is due to the fact that, for these protocols (NC(2^8), RMS and CAT) the increase in number of

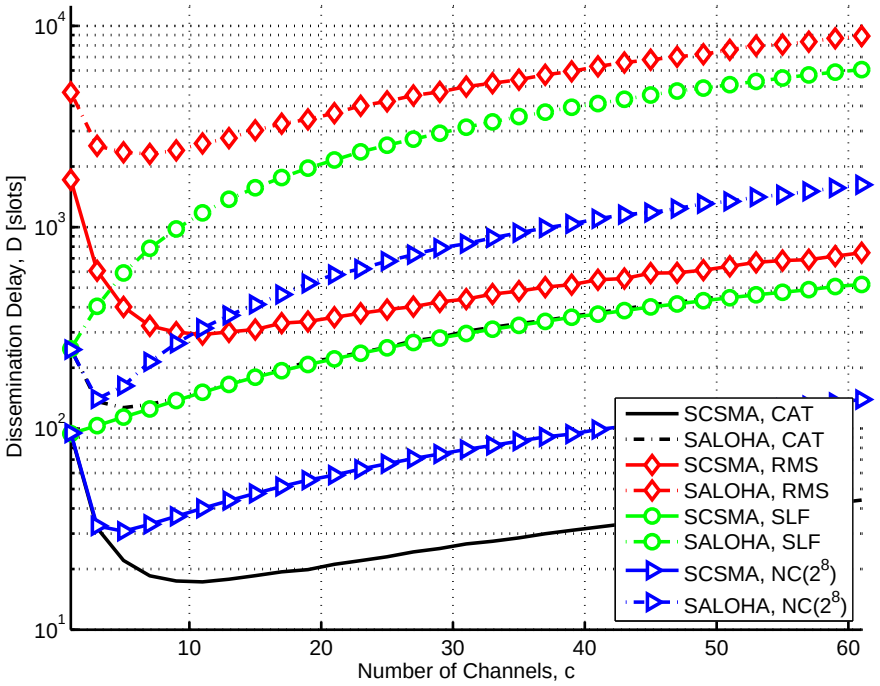


Figure 4.1. *Dissemination delay vs. number of channels for $n = 25$ ($m_r = 0$, $m_c = 0$), and different broadcasting and MAC protocols.*

channels, even though it reduces the receiving probability ρ_r, also increases the circulation of innovative packets in the network (Σ^n decreases faster than ρ_r in Equation 4.15). The dissemination delay starts to increase again once the number of channels is large enough to induce a considerable reduction of the receiving probability, ρ_r which can no longer be compensated by a reduction of Σ^n.

The impact of adversary attacks is shown in Figure 4.2. The network consists of $n = 25$ nodes operating with different numbers of channels. We plot results in two different cases: adversaries are either $m_r = 10$ (Random) or $m_c = 10$ (Colluding). We note that the performance degradation due to the adversary attacks is maximum when nodes access a limited number of channels. Moreover, due to cooperation, colluding adversaries are able to totally disrupt packet exchange between nodes for $c \leq 10$ (the jamming probability is $\mu^c = 1$). As for the random adversaries, they can have a similar impact only when the number of channels is significantly lower ($c \leq 2$). As expected, increasing the number of channels reduces the impact of attacks because hopping in a broader spectrum reduces the jamming probability of both random and colluding adversaries. Moreover, the jamming probability tends to be the same for both types of adversaries when the number of channels

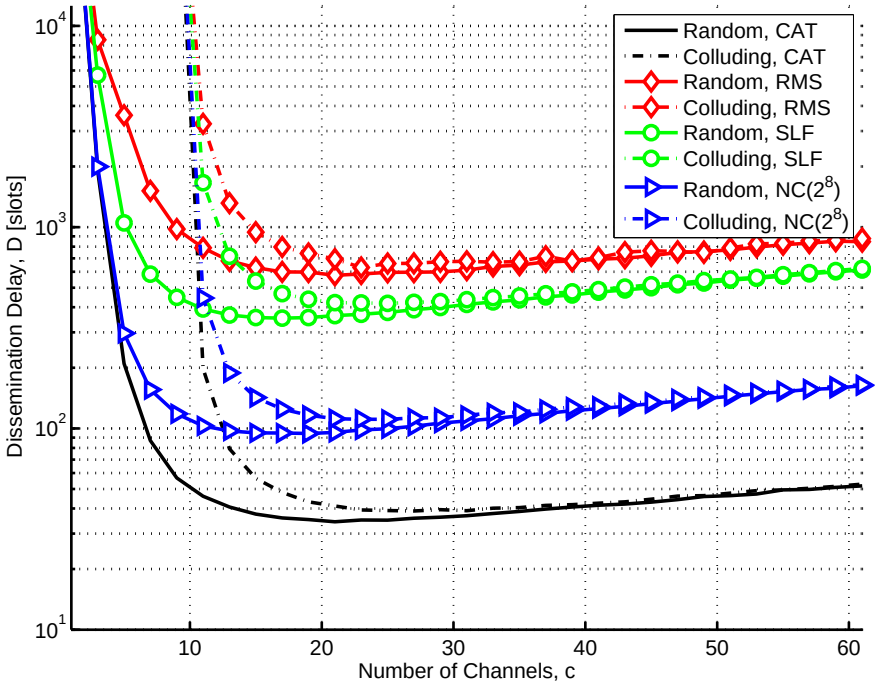


Figure 4.2. *Dissemination Delay vs. number of channels for $n = 25$, for SCSMA with different broadcasting protocols and adversary attacks ($m^r = 10, m^c = 0$ for Random and $m^r = 0, m^c = 10$ for Colluding).*

increases $\lim_{c\to+\infty} \mu^r = \mu^c$ (e.g., for $c = 45$, $\mu^r = 0.78$ and $\mu^c = 0.8$).

However, the distinction between colluding and random adversaries remains very important when it comes to the calculation of $c^\star$. As an example, for the NC(2^8) scheme and random adversary attacks, it is sufficient for the nodes to access $c = 15$ channels to achieve minimum dissemination delay, whereas if the same adversaries are cooperating the delay degrades with a factor of $\mu^c/\mu^r \approx 1.5$, i.e., the dissemination delay increases by 50% due to adversaries' cooperation. Continuing with our example, if nodes are aware of the behavior of the adversaries they can reduce the dissemination delay by increasing the number of channels from 15 to 23. In the opposite case, (colluding adversaries stop cooperating) the delay degrades by approximately 10%, which is less than the previous case. Hence, as a rule of thumb, if nodes do not know what the behavior of the adversaries will be, they can simply assume them to be colluding. This way the delay deterioration due to a possible misinterpretation of the adversary behavior is limited.

We conclude by emphasizing the importance of identifying the correct number of channels nodes have to access to minimize the dissemination delay for NC(2^q) as well as for the other protocols. In the following subsections we focus on network operation in such

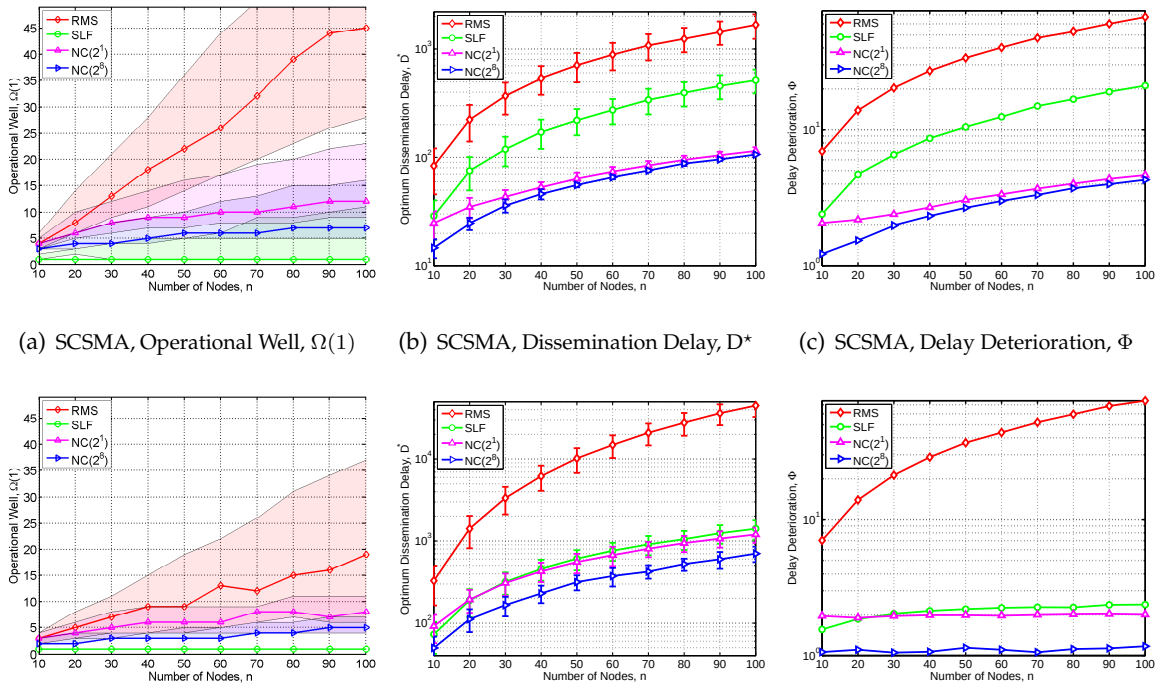
optimum settings and on the impact that different parameters have on the dissemination delay.

4.4.2 Optimum operation in adversary-free networks

We hereby discuss several performance metrics of interest in an adversary-free scenario. We emphasize that, according to the discussion in Section 4.3.4, these results qualitatively hold in the case of oblivious adversaries as well. In the following we are interested in the operational well, optimum number of channels, and delay deterioration with respect to CAT for the different broadcasting protocols. We show these results in Figure 4.3. More precisely, in Figure 4.3(a) and Figure 4.3(d) we show the operational well $\Omega(1)$, and the optimum number of channels $c^\star$ that nodes have to access to ensure minimum dissemination delay for SCSMA and SALOHA, respectively. In the case of SLF, for both MAC protocols, the minimum dissemination delay is obtained for $c^\star = 1$ independently of the number of nodes, n. Moreover, the operational well $\Omega(1)$ coincides with $c^\star$ in the case of SALOHA (Figure 4.3(d)). Instead, it expands with increasing number of nodes, n, when nodes use SCSMA as the MAC protocol, as can be observed in Figure 4.3(a). Hence, the SLF scheme has some degree of robustness with respect to misestimation or behavioral modification of the adversaries only if it is used in conjunction with SCSMA.

We note that a wide operational well is useful as it quantifies the degree of freedom that nodes have when choosing $c^\star$. As long as $c^\star$ is picked inside the operational well (e.g., $\Omega(1)$) the delay deterioration will be limited (no more than 12% worse than the optimum case). The operational well expands as the number of nodes increases for all remaining broadcasting protocols, and is very wide for RMS. The same holds for $c^\star$. However, as can be observed in Figure 4.3(b) and Figure 4.3(e), with RMS the obtained dissemination delay $D^\star$ is prohibitively large.

Focusing on NC(2^q) we notice that it exhibits a trade-off behavior between the RMS and SLF protocols in terms of the optimum number of channels: $c^\star$'s increase with n, in the case of NC(2^q), is more limited compared to its increase for RMS. With reference to Figure 4.3(a) and Figure 4.3(d), it can be seen that such an increase is even steeper in the case of SCSMA. In addition, we note that NC(2^8) needs fewer channels to achieve minimum dissemination delay compared to NC(2^1). This is because, even for a low number of channels, NC(2^8) is able to ensure that each received packet is innovative with high probability. Hence, it tends to select fewer channels in order to take advantage of the higher receiving probability.



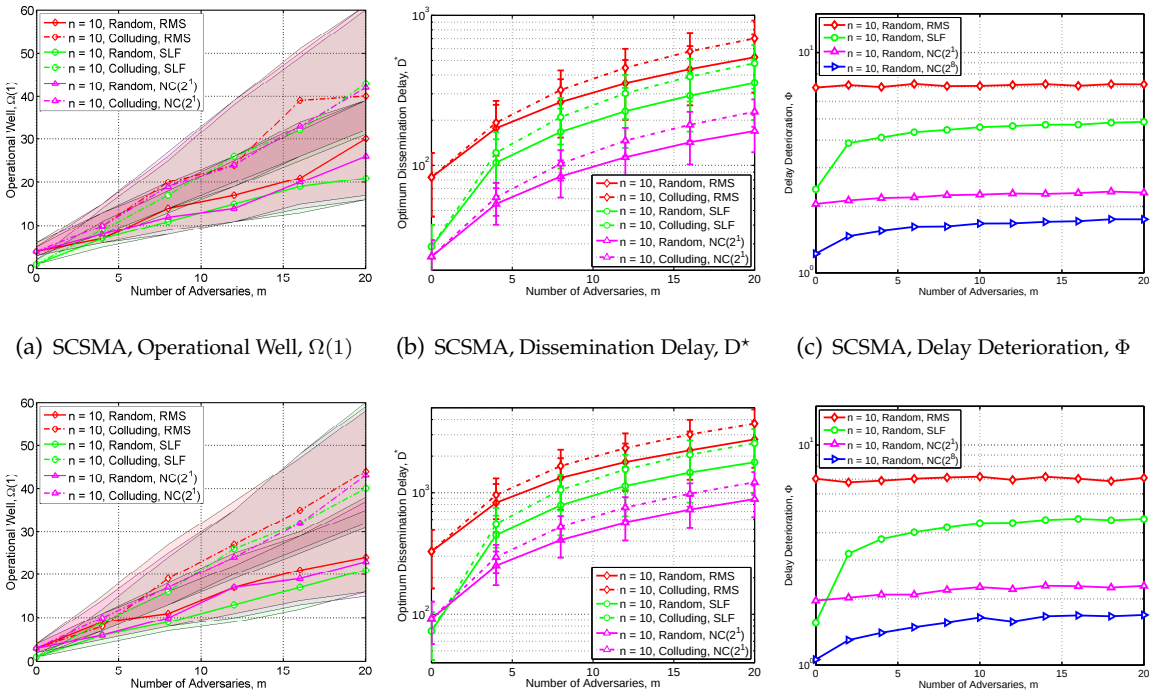
(a) SCSMA, Operational Well, $\Omega(1)$ (b) SCSMA, Dissemination Delay, D* (c) SCSMA, Delay Deterioration, Φ

(d) SALOHA, Operational Well, $\Omega(1)$ (e) SALOHA, Dissemination Delay, D* (f) SALOHA, Delay Deterioration, Φ

Figure 4.3. *Performance metrics for data broadcasting in adversary-free networks vs. number of nodes, n. For different broadcasting protocols we show the operational well and optimum number of channels, the optimum dissemination delay, and the delay deterioration in the case of SCSMA (upper figures) and SALOHA (lower figures).*

The situation is slightly different when the field size is low as nodes require an increase of the number of parallel transmissions in the network to further reduce Σ^n. More parallel transmissions require more channels to bring more packet diversity to the nodes' buffers, i.e., to increase the probability that a transmitted packet is innovative. Hence, due to their characteristics, NC-like protocols tend to be more reception oriented protocols, unlike RMS. A similar behavior can also be seen in the SALOHA case, Figure 4.3(d), even though the optimum number of channels is lowered to counteract the impact that collisions have on the receiving probability and consequently on the dissemination performance.

As to the dissemination delay, we note that for a large number of nodes, both versions of NC behave similarly in the case of SCSMA (Figure 4.3(b)), while with few nodes high field sizes (NC(2^8)) bring more diversity in packet combinations. On the other side, in the case of SALOHA (Figure 4.3(e)), high field sizes lead to a dissemination delay performance which is similar to the ideal CAT scheme, while NC(2^1) tends to have a dissemination performance



(a) SCSMA, Operational Well, $\Omega(1)$ (b) SCSMA, Dissemination Delay, D* (c) SCSMA, Delay Deterioration, Φ

(d) SALOHA, Operational Well, $\Omega(1)$ (e) SALOHA, Dissemination Delay, D* (f) SALOHA, Delay Deterioration, Φ

Figure 4.4. *Performance metrics for data broadcasting in a single hop network with $n = 10$ nodes, under various adversary attacks. For different broadcasting protocols we show the operational well and optimum number of channels, the optimum dissemination delay and its deterioration with respect to CAT for SCSMA (upper figures) and SALOHA (lower figures) as a function of the number of adversaries.*

slightly better than SLF. This can be observed when comparing the delay degradation Φ of NC(2^q) and SLF with respect to CAT in Figure 4.3(f). Moreover we note that, even though SLF and NC(2^1) perform similarly in terms of dissemination delay, NC(2^1) remains the preferred choice when operating in an area where adversaries may appear. This is because by accessing more channels for data dissemination NC(2^1) is more robust to adversary attacks than SLF which must access a single channel to ensure minimum dissemination delay.

4.4.3 Optimum operation under adversary attacks

We continue our performance evaluation in a scenario where nodes are subject to continuous attacks. The adversaries in this case are *aware* of the optimization procedure. That is, they concentrate their attacks on the set of channels that nodes are accessing for data broadcasting, i.e., they know $c^\star$ and the corresponding subset of channels identified by it.

We show simulation results for the case of SALOHA in Figure 4.4. The behavior of

the broadcasting protocols with SCSMA is similar. This is because when the number of nodes is low (in these settings $n = 10$), it can be seen in Figure 4.4(a) and Figure 4.4(d) that the performance metrics under investigation are similar. In addition, when increasing the number of adversaries (either Random or Colluding) in the area, the main factor that changes in Equation 4.23 is λ, which equally affects the choice of $c^\star$ and also the operational well.

With reference to Figure 4.4(a) we note that, for all broadcasting protocols, the optimum number of channels increases for increasing number of adversaries. The same also holds for the operational well. As mentioned in the previous subsection, colluding adversaries have a greater impact on the delay degradation with respect to random adversaries. As can be seen in Figure 4.4(a), under the same conditions, a given number of colluding adversaries force nodes to pick a $c^\star$ which is greater than the case when the same adversaries are random. The same considerations also hold for the dissemination delay; colluding adversaries increase delay more than the random adversaries (Figure 4.4(b)). However, along with an increase in $c^\star$, there is also a relative increase in the operational well. Hence, nodes have more freedom in choosing an appropriate number of channels even in the eventuality of imperfect information regarding the behavior of the adversaries.

In Figure 4.4(c) we plot the delay deterioration Φ of the broadcasting protocols with respect to CAT, for various numbers of random adversaries (results for colluding adversaries are similar). It can be seen that increasing the number of adversaries increases the delay deterioration for most protocols of interest (with the exception of RMS which is structurally similar to CAT). We note that NC-based protocols are robust in the presence of adversaries as the unavoidable degradation, due to the increased number of adversary attacks, is limited. This robustness is particularly evident when the size of the Galois field used to code the packets is large. However, even in the case of a small field size the benefits of network coding are not negligible. Even though, in an adversary free network, NC(2^1) performs slightly worse than SLF (Figure 4.4(c), $n = 10$), we can see that it is capable of maintaining a relatively low delay degradation when the nodes have to deal with adversaries in the area. In addition, the delay degradation of NC(2^1) when increasing the number of nodes is rather limited compared to the delay degradation of SLF.

5

Neighbor Discovery for Cognitive Radio Networks

In a Cognitive Radio Network, besides the strict requirements imposed by the opportunistic co-existence with Primary Users, Cognitive Radios may have to deal with other concurrent (either malicious or selfish) Cognitive Radios which aim at gaining access to most of the available spectrum resources with no regard to fairness or other behavioral etiquettes. With their aggressive behavior, these concurrent users are capable of interrupting or delaying the neighbor discovery process initiated by a Cognitive Radio which is interested in using a portion of the available spectrum for its own data communications. Designing algorithms that assure complete neighbor discovery for Cognitive Radio Networks in a distributed and asynchronous way is essential for their correct deployment.

DEPLOYING a Cognitive Radio Network (CRN) which is going to coexist with one or several Primary Users (PUs) in the area is very challenging [37]. In addition, when considering the natural evolution of CRNs to more complex systems, the challenges and problems to be faced increase dramatically [53]. More specifically, the inherent capability of CRs to base their decisions on their "view" of the environment

The material presented in this chapter has been published in [51, 52].

and to learn from experience makes their operation susceptible to a variety of malicious attacks. This is possible because the same Artificial Intelligence (AI)-based modules that help CRs to operate in an optimal manner may be used by malicious CRs. These malicious CRs, by taking advantage of their Software Defined Radios (SDRs), may be able to feed the AI modules of normal CRs with false sensory inputs, leading them to a substantially modified perception of the surrounding wireless environment. Consequently, such external or internal modification of their wireless environment perception may result in sub-optimal (or even denial of) operation in a given licensed spectrum.

Hence, in a CRN, mechanisms such as cooperation, learning and negotiation help CRs to make the necessary decisions to assure communication in very challenging situations. However, the adoption of such mechanisms requires that nodes of the same CRN be aware of each other and have created a network of trusted CRs. In order to activate such mechanisms (and most importantly establish a CRN), CRs have to successfully complete the first step during network deployment: the neighbor discovery phase. Considering the challenging wireless environment where the generated traffic is highly dynamic and imposes several restrictions to CRs, it is essential that this phase be as quick as possible. Moreover, at neighbor discovery termination, CRs must have obtained sufficient information to enable appropriate mechanisms for their coexistence with other wireless devices in the surroundings. To tackle this problem, we propose a Jamming Evasive Network coding Neighbor discovery Algorithm (JENNA) for CRNs which has the following benefits: 1) it is fully distributed, 2) it does not need global time-synchronization among nodes, 3) it assures fast neighbor discovery, 4) its dissemination performance does not depend on the label space size N, but rather on the actual number of nodes n in the network, 5) it does not need to know the number of nodes n in advance, and 6) it is very robust to different jamming attacks.

According to our algorithm, every CR scans the available spectrum resources and maintains a list of channels which are available for communication. Moreover, it gathers additional information for the correct utilization of the spectrum such as channel occupancy, PUs' identification, selfish CRs that might mimic PUs behavior, etc. Once the CRs have gathered such information, they have to wake up and begin to send control packets in order to discover and disseminate the acquired information to their neighbors. We use network coding to disseminate control packets in an efficient and reliable way, making it possible to have substantial gains in terms of dissemination delay and robustness with respect to malicious CR attacks. The combination of network coding with random channel hopping

sequences makes it possible to obtain an effective neighbor discovery algorithm. It enables the deployment of CRNs in challenging wireless environments in a totally distributed and asynchronous way, fitting very well in the next generation wireless networking paradigm, where cognitive devices should adapt in the best possible way to the wireless environment conditions.

In Section 5.1 we discuss existing neighbor discovery algorithms for traditional and Cognitive Radio Networks. In Section 5.2 we describe the network model discussing the structure and capabilities of both normal CRs and jammer CRs. Section 5.3 introduces the proposed system architecture and a detailed description of its main components, concluding with a representative example of the algorithm execution. In Section 5.4 we present some performance evaluation results for the protocol with respect to baseline schemes that represent the behavior of existing neighbor discovery protocols.

5.1 Existing Neighbor Discovery Algorithms

In order to deploy a CRN, CRs have to discover and exchange information such as neighborhood and spectrum availability with their neighbors. The neighbor discovery process starts when a CR wakes up and begins to broadcast beacons and ends when it receives replies from all its neighbors that are within transmission range.

In traditional ad hoc networks, neighbor discovery is easily implemented as all nodes are tuned on the same channel and follow the rules of a precise wireless standard [48, 54–57]. For example, to perform neighbor discovery in a Slotted ALOHA wireless network with n nodes, nodes can simply transmit with a given probability p_t and listen with probability 1 - p_t in each time slot. This way nodes can rapidly exchange control packets, discovering their neighbors within $O(n \log n)$ time slots if $p_t = 1/n$, which further reduces to $O(n)$ time slots in case nodes are provided with a collision detection mechanism [48]. We note that discovery protocols relying on a single channel are very susceptible to jamming attacks. Once the jammers are aware of the wireless communication standard being used during neighbor discovery, they can adopt multiple types of jamming attacks to deny access to the medium. To launch a basic attack, that can easily cause neighbor discovery failure, jammers can simply transmit a continuous signal in that single channel making it impossible for nodes to exchange any type of information [58].

Unfortunately, in CRNs the neighbor discovery process becomes even more challeng-

ing. In such networks, CRs operate over a set of multiple channels which availability may vary from node to node, depending on their proximity to the PUs and other interference sources. To tackle this problem there have been several proposals in the literature which are based on deterministic [59–61] or randomized algorithms [41]. In particular, in [59, 60] the authors propose neighbor discovery algorithms for time synchronous networks which assure neighbor discovery in $O(CN)$ and $O(Cn\log(N))$ time slots, respectively, where C is the maximum number of channels, n the number of nodes and N the size of the label space from which nodes obtain their ids. Another solution is presented in [61], where the algorithm does not require nodes to be globally time-synchronized. In this proposal the time required to elect a leader which subsequently discovers all neighbors is $O(NC^2)$. We note that these solutions, being based on deterministic algorithms, are very susceptible to jamming attacks, as it is very easy to disrupt neighbor discovery once the jammers know the channel hopping pattern followed by the CRs. Hence, these algorithms are not suitable in a wireless environment which might experience any of the following conditions:

- nodes operate over a wide wireless spectrum i.e., large number of channels C,
- there is a jammer with an AI module able to recognize the hopping pattern of a CR,
- the size of the label space N is large with respect to the actual number of nodes n.

Any subset of these conditions is sufficient to considerably slow down (or disrupt) the neighbor discovery process leading to an incomplete estimation of the network conditions (channel occupancy, traffic requirements, CR's availability, etc.).

A different approach is proposed in [41]. Here the authors assume that nodes are globally time-synchronized (e.g., nodes equipped with Global Positioning System (GPS) modules). They consider different frequency hopping patterns, from single frequency hop to random hop patterns in order to derive fast and energy-efficient neighbor discovery algorithms when nodes have to access a spectrum with a large number of available channels. They show that when using a random algorithm based on single frequency hopping, nodes can assure fastest neighbor discovery. Instead, random hopping over all the available channels gives longest neighbor discovery delay, and other algorithms proposed have intermediate performance. We note that, in the presence of jammers, the best strategy to adopt is random hopping over all channels as it does not require nodes to share information prior to neighbor discovery.

As to neighbor discovery in the presence of jammers, there has been some research in

the case of traditional single channel networks [57, 62, 63], while for multi channel CRNs, to the best of the authors' knowledge, no solutions to this problem have been proposed so far. Hence, JENNA is the first solution to the neighbor discovery problem in CRNs in the presence of different types of jammers.

5.2 Network Model

The wireless spectrum is subdivided into a set of C orthogonal channels available for opportunistic CR communications. In addition, this spectrum band is given under license to N_p PUs which have priority in accessing it at any given time and frequency.

5.2.1 Normal cognitive radio

CRs can access any of the licensed channels $k \in \{1, \ldots, C\}$ every time they can assure not to interfere with any activity of the PUs. To achieve this, they are capable of sensing the available spectrum with techniques that will be mentioned briefly in Section 5.3.2. CRs are assigned a unique identifier and are equipped with a single transceiver.

We further assume that CRs use SCSMA as the MAC protocol and do not require tight global time synchronization. They only require to have similar clock ticks in order to be able to synchronize at the slot boundaries with their neighbors [64].

5.2.2 Jammer cognitive radio

Adversary CRs launch jamming attacks [53, 65] denying channel access to normal CRs for an arbitrary period of time, potentially delaying or totally disrupting the neighbor discovery process of the CRN. We divide these jammers into two categories based on different properties which are of interest from the neighbor discovery perspective.

The first type of adversary CR, called static, is capable of acting improperly to gain exclusive access to the available resources. This may happen when multiple CRs are accessing the same limited spectrum resources and any of them behaves selfishly in order to satisfy its bandwidth demands for data communications. Techniques such as Primary User Emulation (PUE) attacks [66] can be adopted by such CRs to achieve their goal. When launching these types of attacks, a malicious CR (jammer) is capable of mimicking the PU signal's characteristics, leading legitimate CRs to vacate the spectrum as they erroneously

believe that the spectrum is being used by a PU. Distinguishing between PU and PUE signals is extremely challenging as normal CRs have to be able to promptly extrapolate the received signal features and exchange useful information to implement cooperative detection in order to increase the trustworthiness of their decision.

This *static jammer* is characterized by the following features:

- single channel operation. It continuously emits radio signals in a given frequency for a long period of time;
- selfish behavior. It aims at illegally reserving spectrum resources for its own communications;
- improper use of AI modules. By mimicking PUs signal characteristics, this CR misleads normal CRs into concluding that the spectrum is occupied by legitimate PUs. The most representative example is the PUE attack;
- reduction of available spectrum resources. Because CRs confuse this jammer with a PU, they will have fewer channels to choose from, since channels that are detected as occupied by a PU cannot be used.

Another type of jammer, called reactive, does not have selfish purposes but rather behaves maliciously by interrupting legitimate CRs' operations. This jammer transmits packets that do not obey CR MAC rules, causing denial of operation for any CR that is attempting to access a given licensed band. In addition, these jammers are very effective in disrupting communications over a wide spectrum by transmitting jamming signals randomly in frequency and time [58].

This *reactive jammer* is characterized by the following features:

- multi channel operation. It randomly emits radio signals, hopping over all the available wireless spectrum;
- malicious behavior. It aims at corrupting any exchanged packet in a given frequency and time slot;
- basic operation. It uses a random frequency hopping pattern to access all the channels and transmits high power spectral density signals. Its signal does not try to emulate a PU signal;
- decreased reliability of the available spectrum resources. By accessing all the available spectrum bands, a reactive jammer can significantly degrade the CRN communication's quality equally in any accessed spectrum band. In these conditions a CR is not

able to find an appropriate spectrum band for its communications but it rather has to adopt a robust communication scheme, to minimize the impact of the jamming attack.

From the neighbor discovery point of view, a static jammer is *unintentional* as it does not explicitly aim at disrupting the neighbor discovery process initiated by the CRN. On the other side, a reactive jammer is *intentional* as its goal is to disrupt any communication attempt by any CRs which is within its transmission range. Note that, since reactive jammers want to disrupt communications between nodes in a normal CRN, it is reasonable to assume that they get activated as soon as they sense any data exchange by the nodes of a CRN. The very first time CRs begin to exchange packets is to discover their neighbors. Hence it is essential that, during neighbor discovery, CRs take adequate countermeasures to minimize the impact of these jammers.

Let $j_k^s \in \{1, \ldots, J^s\}$ represent a static jammer transmitting on a given channel k and $j^r \in \{1, \ldots, J^r\}$ a reactive jammer transmitting randomly in $\{1, \ldots, C\}$. For simplicity, we assume that all jammer CRs have the same communication range R_j and that any CRs which happen to be within their range, tuned on the jammed frequency in a given slot, will receive unrecoverable packets, i.e., all packets interfered by a jammer will be considered to be lost.

5.3 JENNA: System Architecture

In this section we present the system architecture of the proposed Jamming Evasive Network coding Neighbor discovery Algorithm. A conceptual representation is shown in Figure 5.1. We first give a general description, then focus on its main components and finally give an example of the algorithm execution in a simple CRN.

5.3.1 General description

Let $n \leq N$ be the number of CRs which are going to participate in the neighbor discovery process, where N is the size of the label space for assigning ids to the CRs. With reference to Figure 5.2, CRs that want to perform neighbor discovery wake up, enter the *passive* mode, and start the spectrum sensing phase. Let $i \in \{1, \ldots, n\}$ denote a generic CR and x_i the control packet that it has generated at the end of this spectrum sensing phase. Subsequently, it decides to perform neighbor discovery entering the dissemination phase during which nodes mutually exchange random linear combinations of their packets and

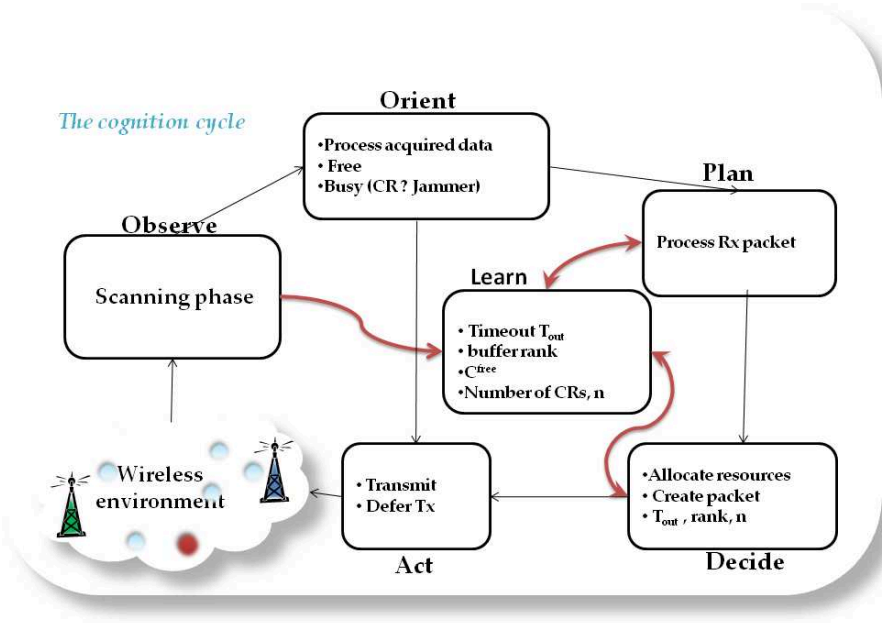


Figure 5.1. *Conceptual representation of the proposed neighbor discovery algorithm.*

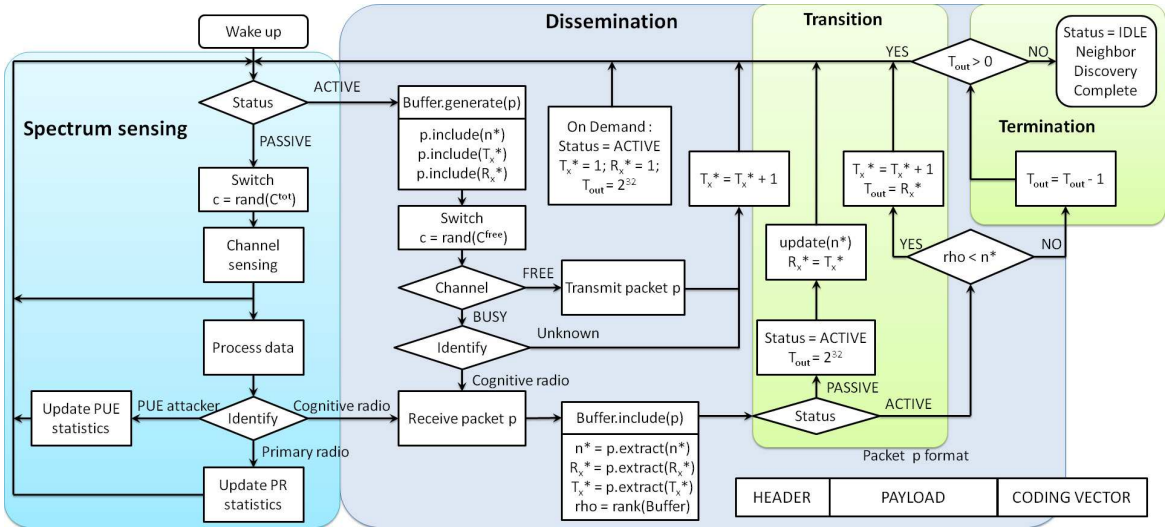


Figure 5.2. *Block diagram for the proposed neighbor discovery algorithm. T_x^* is the time reference since the beginning of the dissemination phase, R_x^* estimates the duration of the transition phase, and n^* is the estimated number of nodes.*

which ends when all nodes have received enough coded packets to retrieve the packets generated by any other node. In the following, we describe in more detail each of these phases.

5.3.2 Spectrum sensing phase

We assume that initially all CRs in the network are in *passive* mode. After a CR wakes up, it enters the sensing phase where it stays until it decides or is requested to perform neighbor discovery by some nearby node. During this phase, cognitive radios scan the set of all channels $\{1, \ldots, C\}$, following a random hopping pattern, detecting independently the existence of PUs activity and of possible PUE attackers (more generally static jammers). This can be achieved by using spectrum sensing techniques, such as Energy Detection (ED), Cyclostationary Feature Detection (CFD) or Matched Filter Detection (MFD) [67]. The problem of detecting PUE attackers can be tackled with techniques such as [68]. As discussed in Section 5.2.2 we assume that reactive jammers are not active during this phase as they initiate jamming as soon as they sense activity in a given channel, i.e., when a generic CR begins to broadcast packets for neighbor discovery [58]. At the end of this phase, each node i in the CRN has created a list of free channels $C_i^{free} = \{k_1, k_2, \ldots, k_{K_i}\} = \{1, \ldots, C\} \backslash C_i^{busy}$ that i can use for communications, where C_i^{busy} is the set of channels used by either PUs or static jammers. This list will be included in the control packet along with additional information such as which channels are used by legitimate PUs and PUE attackers, etc.

5.3.3 Dissemination phase

This phase starts when a generic CR i begins neighbor discovery, sending its first control packet randomly in a channel $k \in C_i^{free}$. The channel hopping pattern used by the algorithm is a random sequence with a generation seed that is calculated in real time. This way reactive jammers cannot disrupt CRs' communications even if they gain access to the internal memory of a CR where predefined generation seeds are stored. If a CR j happens to be synchronized in channel k, it receives the packet and enters the *active* mode. This packet is then included in CR j's buffer and its corresponding encoding vector is included in the decoding matrix. We note that at the beginning of the dissemination phase CRs are more oriented on sending their own control packets, in which case a receiving node does not need to perform any decoding. When the received packet is a network coded packet, i.e., it includes information from multiple CRs, the receiving CR adds the corresponding coding vector as a new row of the decoding matrix, and checks whether it is possible to invert it. Note that in general a node does not know the number of neighbors, n, but only the size of the label space $\{1, \ldots, N\}$ from which node IDs are taken. For this reason, the encoding

vector must have N entries, each corresponding to a generic ID in the whole set. Upon reception of a new coded packet, a node will look at the newly updated version of the coding matrix, where all-zero columns correspond to node ids that have not been assigned to any of the neighbors, or to nodes whose information has not yet been received. After removing these all-zero columns, the node checks whether or not the resulting matrix is invertible. If it is not, the node needs to wait for more coded packets before it can retrieve the original control packets. If instead the matrix can be inverted, the node is able to retrieve the original packets involved. In this case, the node assumes that it has received the full set of packets and, after a time-out interval during which it continues to disseminate packets, it considers the dissemination process complete. More specifically, the dissemination phase consists of the following three sub-phases.

Transition phase: This phase starts at the same time as the dissemination phase, and ends when all CRs participating in the neighbor discovery process have become active. We note that, during this phase, there is dynamic diversity among CRs: a portion of CRs are in *passive* mode, where they are still sensing the available channels, and the rest of them are in *active* mode, disseminating the control packets. Hence, this phase is very delicate as it is the moment when reactive jammers are likely to get activated to counteract the neighbor discovery process initiated by *active* CRs. This is because CRs in *passive* mode may find their sensing data misleading as they will sense activity caused by reactive jammers in terms of short impulses, hence probably including these channels as not free for communication. However, considering the different signal characteristics of static and reactive jammers, a CR may still be able to distinguish between them. Hence, it can avoid those channels which are used by static jammers (which permanently occupy a certain frequency), while keeping in the list of available channels those used by reactive jammers (which randomly hop and therefore cannot be avoided).

Active phase: When all CRs are in the active state, all nodes exchange control packets. This is the *active* dissemination phase, which lasts until a node believes that the dissemination process has finished.

Termination phase: Once a node is able to invert its decoding matrix as explained above, it enters the termination phase where it decrements the T_{out} counter.

Depending on the way CRs use the timeout value T_{out} we implement two different versions of the algorithm, namely, *asynchronous* and *synchronous*. In the first case CRs use the estimated timeout period to disseminate packets to neighbors that have received only partial

information so far. In the second case CRs synchronize to the same timeout value making it possible to end the neighbor discovery process at the same time slot for all CRs. This is done by sharing the value of T_{out} among all nodes, including it in the packets header, and its value is dictated by the last CR that has been able to decode all the received packets.

During the termination phase, nodes are still enabled to transmit and receive packets. This is done for two reasons: (i) to provide a more reliable packet dissemination, as other neighbors might not have yet been able to decode the information stored in their buffers, because they do not have enough linearly independent combinations, and (ii) to help CRs avoid early termination of the neighbor discovery process. Note in fact that it is possible that a node is able to invert its decoding matrix even though it has not received all packets (i.e., the received coded packets are such that inversion of a submatrix is possible). When this happens, according to our algorithm a node would be led to concluding that the dissemination process is terminated, whereas in fact it is not. If during the time-out period a node receives an innovative coded packet (i.e., a packet that increases the rank of its decoding matrix), then it deduces that the dissemination process is not complete, and goes back to the active state, waiting for more packets. If instead during the time-out period no innovative packets are received, the node concludes that there is no more information to be received. We remark that, thanks to this mechanism our algorithm does not require any prior information about the number of nodes, n involved, but rather it estimates this number as part of the dissemination process itself. Of course, there is always a chance that a node prematurely stops the dissemination process, as described above. However, if the system's parameters are chosen carefully, and in particular the degree of mixing is sufficiently high and the value of T_{out} is sufficiently large, this event occurs with low probability and its effects are negligible.

5.3.4 Description by example

To briefly describe the scheme, in Figure 5.3 we show the execution of the algorithm in a simple network consisting of 5 channels and 10 CRs which have to coexist with one PU and a reactive jammer active in the area. With reference to Figure 5.3, at the beginning, all CRs are in *passive* mode, scanning all the channels to detect the presence of PUs or static jammers in the area. When the sensing phase finishes, the CRs have their channel list $C_i^{free} = \{1, 2, 3, 5\}$ for $i \in \{1, \ldots, 10\}$. Note that in general the list of available channels may be different for each CR. At a given moment, slot $s = 1$, CR 8 wakes up and becomes *active*, sending its

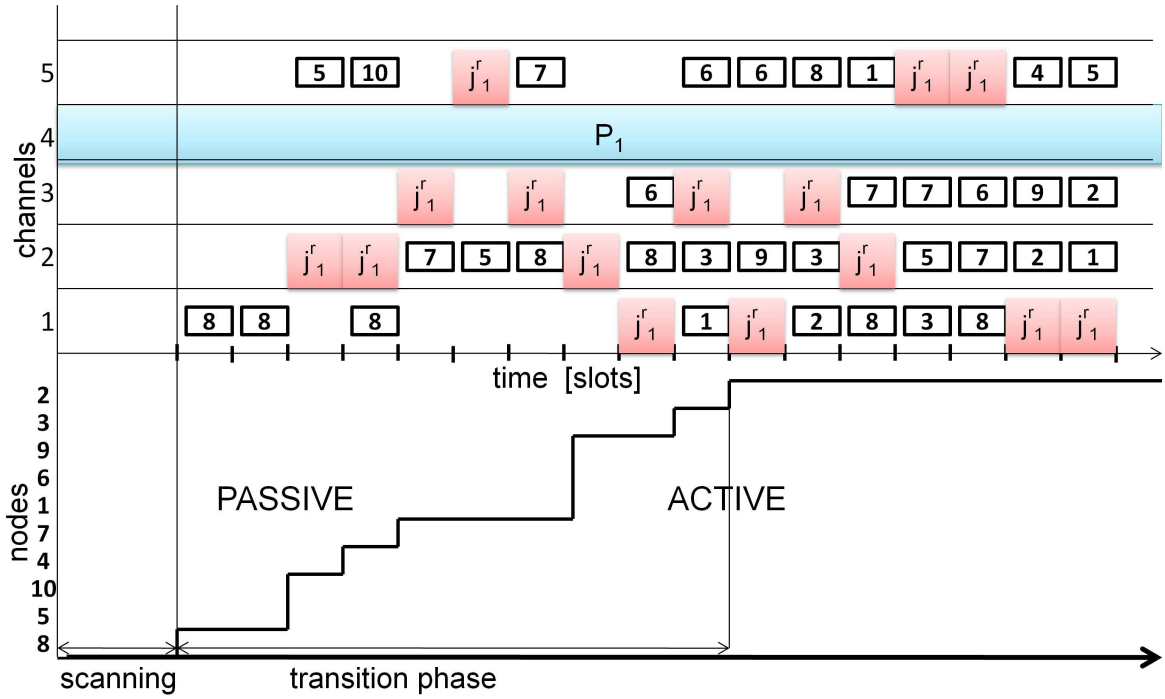


Figure 5.3. *Example of the neighbor discovery algorithm execution for a CRN with 5 channels, 10 nodes, a reactive jammer j_1^r, and a PU P_1 operating in the area of interest.*

control packet on channel 1. This is the first transmitted packet and denotes the beginning of the dissemination and transition phase. The transmission of the packet by CR 8 wakes up j_1^r, which happened to be sensing in channel 1. Hence, j_1^r initiates random jamming over all the available channels with jamming duration equal to the slot length. In slot 2 the transmission of node 8 successfully reaches nodes 5 and 10, which then enter the *active* state. In slot 3 we have CR 5 transmitting in channel 5. The control packet generated by CR 5 contains a linear combination of its control packet and the one it received in the previous slot from CR 8. Following this procedure, in slot 10 all CRs are active, which denotes the end of the transition phase. Nodes continue to disseminate information to each other, hopping randomly in the set of available channels (1, 2, 3, 5 in this case) and avoiding to transmit their control packet every time they happen to fall in the same frequency as the reactive jammer (which they can detect by simple carrier sensing before transmission). When a node receives enough packets to decode the information, it initiates the termination phase during which it reduces the time-out counter and continues packet forwarding to help other nodes in the decoding process.

5.4 Performance Evaluation

In this section we evaluate and compare different versions of the proposed neighbor discovery algorithm (NC(2^q), synchronous (Sync) and asynchronous (Async)), with respect to baseline schemes that use SLF and RMS for data broadcasting (see Section 4.2.2 for more details). We note that, these baseline schemes require nodes to know the actual number of CRs, n in the network for neighbor discovery termination.

We consider a CRN involved in the neighbor discovery process sharing a set of $C = 30$ channels with PUs, static and reactive jammers. Nodes are all within transmission range of each other. Hence, after the sensing phase CRs hop over a similar set of available channels. For the simulation results we assume that this set of channels is the same for all nodes. This is because PUs are assumed to have a transmission range which is higher than that of CRs. Moreover the spectrum is subject to frequent attacks by reactive jammers which, once they have detected CR activity on a channel, begin jamming randomly the available channels to disturb the neighbor discovery process. We assume that reactive jammers, after being activated, do not sense the medium before transmitting their jamming impulse, i.e., on a given frequency there can be more than one jammer in a given slot.

5.4.1 Impact of network coding

We hereby describe the impact of network coding on the dissemination delay performance. This is shown in Figure 5.4 where we plot the dissemination delay, i.e., the average number of slots required for all nodes in the network to discover their neighborhood against different neighbor discovery schemes. As we can see, using network coding provides faster dissemination of the control packets, making it possible to finish the neighbor discovery in less time. The achievable improvement with respect to SLF ranges from 3 to 6 times in these settings. Regarding the coding performance we note that coding over GFs of higher size does not bring significant gains in terms of dissemination delay, except for the case of a small number of CRs where packet diversity is highly beneficial. These results are in line with those obtained in Section 4.4 As an example, for $n = 10$, the gain obtained when using NC(16) instead of NC(2) is around 20% and further increases of the GF size do not provide any additional gain. This lack of further gains from higher coding sizes comes from the fact that the random hopping pattern that CRs follow is already able to provide most of the required diversity for a fast neighbor discovery process.

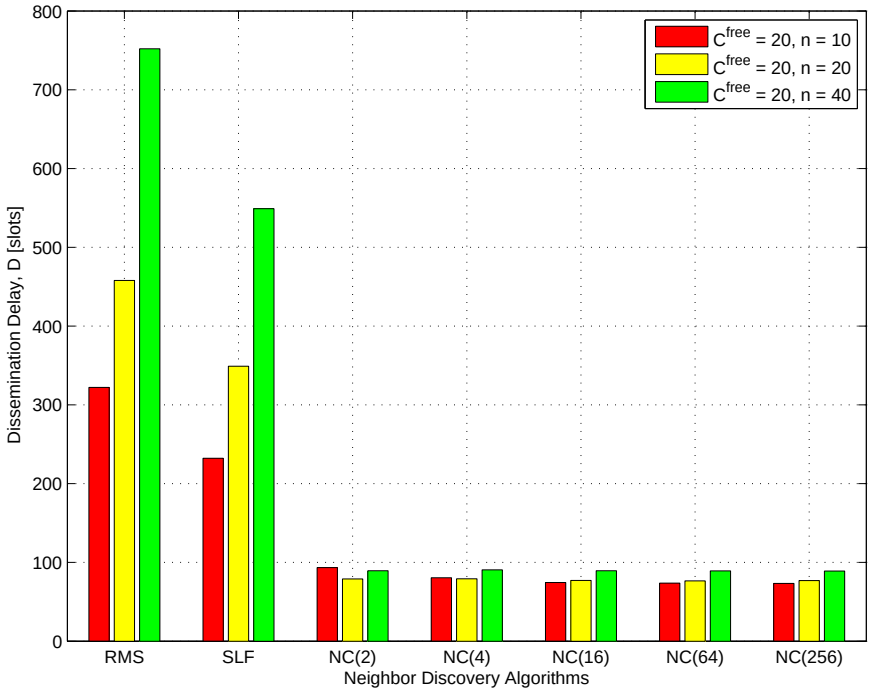


Figure 5.4. *Comparison of the dissemination delay for RMS, SLF and the asynchronous versions (varying the GF size for network coding) of the JENNA protocol for different numbers of CRs, n.*

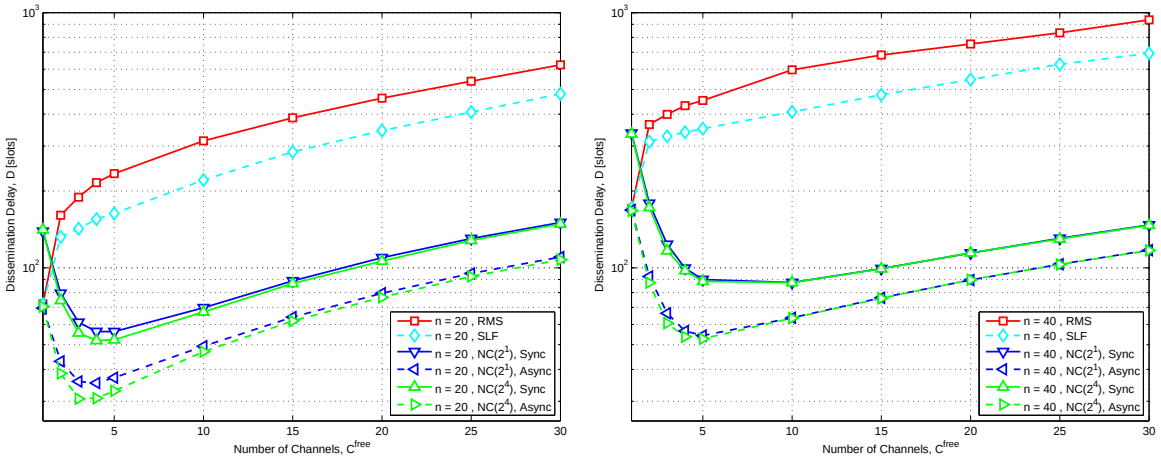


Figure 5.5. *Dissemination delay vs. number of free channels for various schemes and $n = \{20, 40\}$.*

5.4.2 Impact of free channels and number of CRs

In Figure 5.5 we show the dissemination delay as a function of the number of free channels for $n = \{20, 40\}$. As it can be seen the dissemination delay increases for all schemes as the number of free channels increases. This is due to random hopping where, with a wide range of free channels, it is more likely that a CR tunes to a channel where there are no other

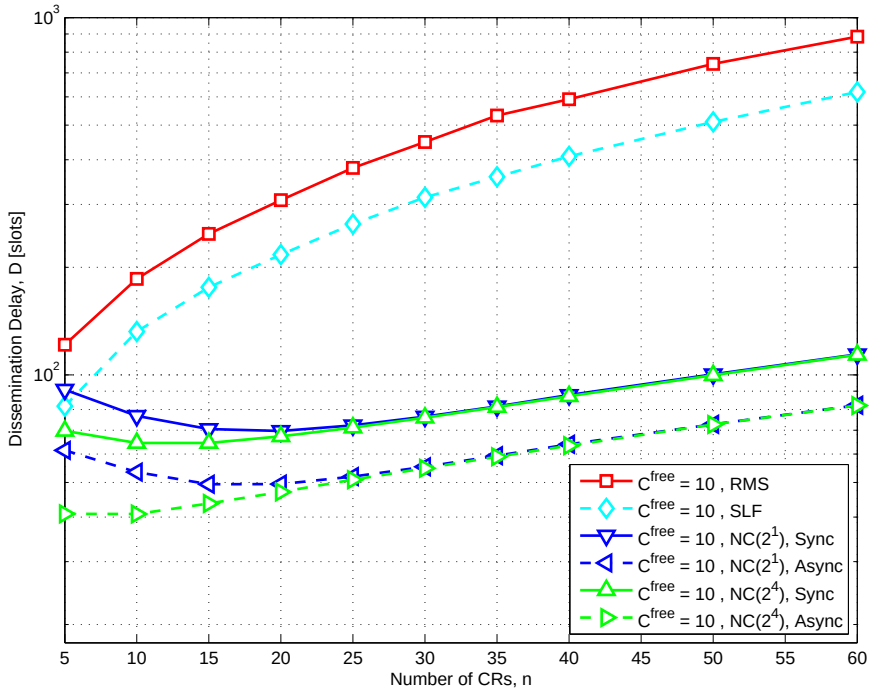


Figure 5.6. *Dissemination delay vs. number of CRs, n for different schemes and number of free channels.*

nodes with which to exchange control packets. In addition, as the number of free channels increases, the number of transmitters per slot also increases because, once nodes are randomly allocated over all free channels, one of them is allowed to transmit per channel. With nodes scheduled for transmission more frequently, the opportunities to receive innovative packets diminish accordingly. We note that in the case of network coding the situation is different. Given the number of nodes n, there exists an optimal number of available channels C^{free} that minimizes the dissemination delay. This is because, with this particular number of free channels, all nodes that are scheduled for transmission bring more innovative packets, maximizing the rate at which the nodes' buffers rank increases.

In Figure 5.6 we observe the same behavior in terms of dissemination delay with respect to the number of CRs which are involved in the neighbor discovery process. However, we note that network coding is particularly robust in terms of dissemination delay for varying number of CRs in the network, providing comparable performance for a wide range of number of nodes, which is not true in the case of RMS and SLF that suffer particularly in those scenarios where the neighbor discovery has to be performed for a large number of CRs. This behavior can be explained as follows. With more nodes in the network, the number of packets each node needs to collect is correspondingly larger, which obviously tends to

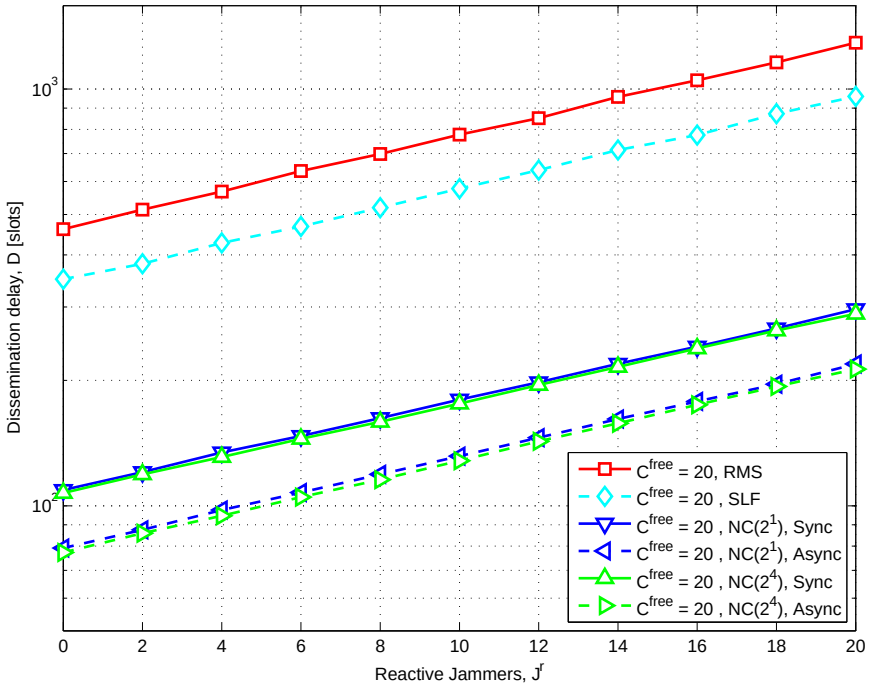


Figure 5.7. *Dissemination delay vs. number of reactive jammers J^r for $n = 20$ and $C^{free} = 20$.*

increase the dissemination time. This is partially mitigated (for all schemes) by the fact that the number of receiving nodes per channel increases and hence packets are disseminated to more CRs in each slot. Moreover, the performance of the schemes using network coding is further improved by the increased diversity in packet mixing (linear combinations contain information generated by more CRs), which leads to considerably faster control packet dissemination and explains why network coding based schemes significantly outperform RMS and SLF. We also note that in all cases the synchronous version of the neighbor discovery algorithm takes more time to terminate. This is because nodes need not only to decode the packets but also synchronize to the same value of T_{out}.

5.4.3 Impact of reactive jamming attacks

In Figure 5.7 we show the performance of the algorithm as a function of the number of reactive jammers which are accessing the available channels randomly over frequency and time. In all cases network coding dissemination is faster with respect to RMS and SLF, with gains that are 6 and 4 times, respectively. We note that this gain is constant with the number of reactive jammers as a packet loss due to reactive jamming has the same effect on the dissemination delay for all schemes under discussion (the same conclusion can be obtained

from the analysis in Section 4.3 given that reactive jammers are a subtype of random adversaries). We also observe that the impact of reactive jammers highly depends on the available spectrum resources which are being accessed by the CRs and the reactive jammers. With the same number of reactive jammers, as the number of available channels decreases the probability to disrupt a packet transmission in a given channel increases, leading to higher neighbor discovery delay. As expected the synchronous version requires additional time to provide simultaneous termination for all nodes.

6

Dynamic Spectrum Access for Cognitive Radio Networks

Dynamic Spectrum Access allows Cognitive Radios to opportunistically and efficiently access the licensed spectrum resources as long as they can guarantee not to interfere with Primary Users' activity. As such, designing a solution for Cognitive Radio Networks poses several challenges such as the reliable detection of Primary Users and the mutual information exchange among Cognitive Radios to coordinate for Dynamic Spectrum Access purposes.

By promising significant gains in the efficiency of usage of the precious electromagnetic spectrum, Dynamic Spectrum Access (DSA) techniques have attracted significant attention from the research community in recent years. Several trends orient towards a distributed architecture where CRs are capable of interacting with each other in order to reliably detect available spectrum resources and coordinate their efficient usage. This is a requirement in wireless networks where the CRs are allowed to operate opportunistically in licensed bands as long as they do not interfere with PU communications.

The recent literature has addressed the numerous challenging technical issues that arise when applying the DSA paradigm to a wireless networking context. However, the major-

The material presented in this chapter has been published in [69–71].

ity of the practical schemes which have been proposed do not provide a comprehensive solution to all these issues, but rather focus on a given subset of interest. In [72–74] the authors propose multi channel protocols which aim at resolving dynamic channel allocation issues such as efficient spectrum resource usage, connectivity and throughput. A major drawback of these solutions is that they assume the existence of a static Common Control Channel (CCC); this is in contrast with the principles of DSA, since it requires a static allocation of spectrum resources for control purposes. An improvement to this approach is [75] where the authors propose the C–MAC protocol, which works over a dynamic CCC able to provide, among other things, broadcast communications.

However, all these approaches [72–75] do not address the problem that the CCC can easily become the bottleneck of the whole system, thus preventing efficient reuse of unused licensed spectrum. To cope with this issue, a set of papers propose the *multiple rendezvous* approach, which consists of eliminating the need for a CCC by adopting other techniques to have the wireless nodes meet in some channel when they are to communicate with each other. An example is [76], where nodes can exchange control information in all channels thanks to their ability to hop synchronously on unused channels when they are not performing data transmission; once they have negotiated the transmission channel, they dwell on that particular channel to perform data communication.

A similar approach can also be found in [77, 78], in which every node has an associated channel hopping sequence to be used for reception, and senders synchronize on the hopping sequence of the intended receiver to perform transmissions. This type of solution is very interesting in that it eliminates the need for a CCC for medium access purposes. However, there are three major drawbacks in this approach: 1) the protocol does not support very well the exchange of control information, such as for instance broadcast packets to be used for routing purposes, 2) the allocation of spectrum resources to data communication is not designed for spectrum efficiency, and 3) these schemes do not take into consideration operation in licensed bands where the nodes are required to avoid interfering with PUs.

Lately, several works have dealt more specifically with the last two of these issues [79–82]. In order to provide a practical scheme able to provide minimum interference to PUs while allowing efficient secondary spectrum access, the authors of [79] propose a decentralized access scheme where CRs sense the medium and opportunistically transmit on channels which are estimated not to be occupied by PUs. This scheme works without the establishment of a CCC, which is an interesting feature; however, this is achieved by having

each CR detect the presence of primary activity and decide whether to access the spectrum independently from other CRs. Due to this last aspect, in order for the performance of PU detection to be satisfactory, CRs need to adopt high performance detection techniques, thus increasing device complexity and cost. To overcome this problem, it is possible to introduce Cooperative Detection (CD) strategies, as suggested in [80]. From the device complexity point of view, the use of CD makes it possible to use cheap technology sensing detectors while providing the same performance as independent decision-making devices with higher cost sensing detectors. However, it is not possible to perform CD without the presence of a facility allowing CRs to exchange detection information, i.e., a control channel.

To summarize, designing a DSA scheme which is completely distributed, can provide an efficient usage of the spectrum, includes an effective strategy for the identification of available spectrum resources, and does not rely on statically allocated spectrum resources for the exchange of control information, is still to be solved. In the following we propose a DSA scheme with the objective of meeting all these requirements in a single hop CRN.

6.1 System Level Description

The key principle on which our proposal is based, is that CRs visit channels in a pseudo-random fashion and exchange control information whenever they happen to meet in any channel. The efficient dissemination of the control information to all CRs is achieved by means of a network coding control channel. The control information exchanged by the CRs consists of all the information (such as intended receivers, PU presence, etc.) which is needed to select channel switch patterns as well as resource allocation for data communication according to a pre-defined deterministic algorithm. If the control information generated by each CR is disseminated to all CRs, then they can run the same deterministic channel allocation algorithm with the same input information. Hence, channel allocation can be done in a distributed fashion without requiring a centralized control scheme or coordination among nodes.

This Network Coded Cognitive Control Channel (NC^4) is naturally fit for CD of available spectrum resources: given that all CRs already switch over all available channels in a pseudo-random fashion for control information dissemination purposes, it is possible for them to carry out a comprehensive PU detection just by using a signal detection technique whenever they switch channel. The detection information by all CRs is then disseminated

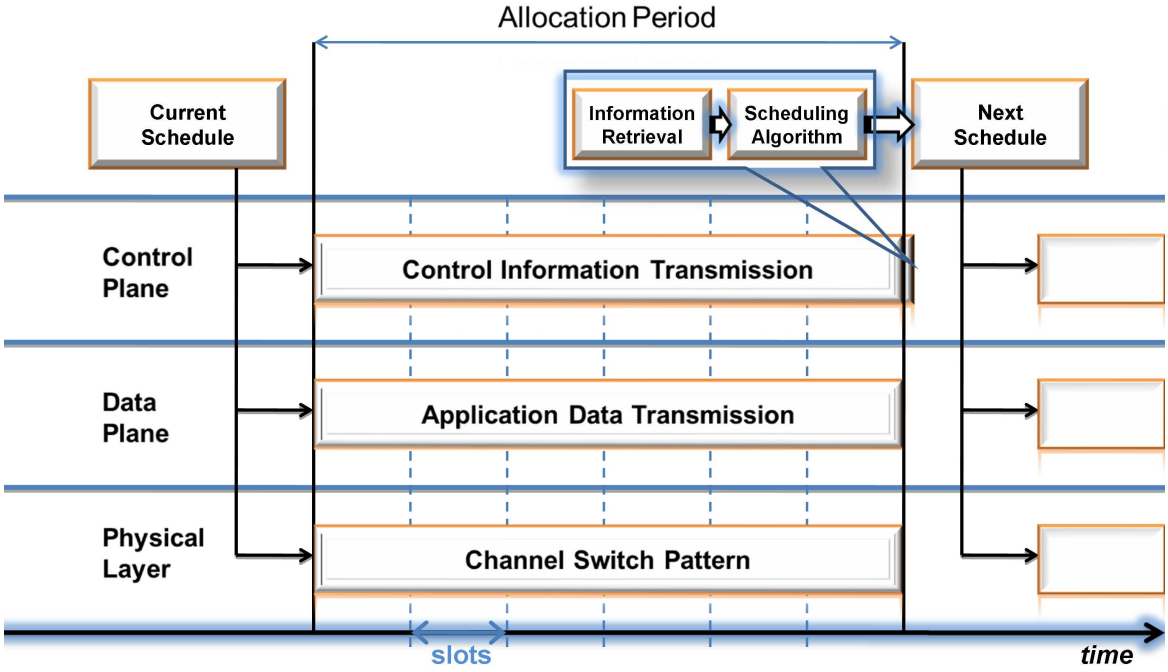


Figure 6.1. *Sequence of operations in the proposed distributed multi channel medium access scheme.*

via the NC^4, so that every CR can independently run the same CD algorithm in order to determine the available spectrum resources. The use of cooperation provides significant improvements with respect to the performance of a single detection attempt; thanks to this feature, the adoption of simple and cheap techniques, such as ED [83], can be very effective. The conceptual representation of the scheme is shown in Figure 6.1.

In the remainder of this chapter, we will first describe the basic version of our scheme, Network Coded Cognitive Control Channel-Medium Access Control (NC^4-MAC), which aims at performing multi channel medium access in a single hop network; then we will discuss its extension to DSA scenarios, which is called Network Coded Cognitive Control Channel-Dynamic Spectrum Access (NC^4-DSA).

6.2 NC^4-MAC

We divide time in *allocation periods* of duration T_{all}, each identified by an index t. Each allocation period is divided into S slots of equal duration. The interval corresponding to each

slot is further divided into two sub intervals of variable duration: the first one is reserved for the transmission of application data by a single CR, and the second is used to exchange control information packets.

We assume that all CRs are synchronized at the slot level. Distributed synchronization strategies such as the ones adopted for wireless sensor networks [84] and vehicular networks [85] are suitable for NC^4-MAC. The inaccuracy of the synchronization can be accounted for by inserting a guard interval at the beginning of each time slot. Moreover, with the accuracy provided by some of the techniques in [84] and for typical values of the slot duration (see Section 6.2.4), the duration of the guard interval would be small with respect to the slot duration, and hence the impact of synchronization inaccuracies on the performance of NC^4-MAC will be low. Transmission is performed using a Time Division Multiple Access (TDMA) scheme based on a scheduling strategy that will be explained later. In a given slot $s \in \{1, \ldots, S\}$, CR $i \in \{1, \ldots, N\}$ will tune to channel $B_{i,s}(t)$; thus, the matrix $B(t) = \{B_{i,s}(t)\}$ summarizes the channel selection pattern of all CRs during a particular allocation period. Each CR transmits exactly one control packet having a fixed duration T_{ctrl} in each time slot. Therefore, the duration of the control sub interval for slot s and channel $k \in \{1, \ldots, C\}$ in the allocation period t is defined as T_{ctrl} times the cardinality of the set $\{i : B_{i,s}(t) = k\}$ of all CRs which are tuned on that channel in that slot. The duration of the data sub interval is defined as the duration of the slot T_{all}/S minus the duration of the control sub interval; since CRs visit channels pseudo-randomly, the average duration of the data sub interval is $T_{all}/S - T_{ctrl}N/C$. We note that T_{all}, S and T_{ctrl} are system parameters which can be chosen to achieve different performance trade-offs; a discussion on this topic will be given in Section 6.2.4.

For every slot s and channel k in the allocation period t, a particular CR $A_{k,s}(t)$ is allowed to transmit application data in the data sub interval of the slot; therefore, the matrix $A(t) = \{A_{k,s}(t)\}$ summarizes the scheduling according to which CRs transmit application data during a particular allocation period. An example of channel switch pattern and data transmission scheduling is represented in Figure 6.2.

The key aspects of our proposal are the following:

- at the beginning of allocation period t, each CR i generates a control packet $x_i(t)$. The exact information that each node includes in its control packet depends on the allocation algorithm which is to be used. For NC^4-MAC, as we discuss in Section 6.2.1,

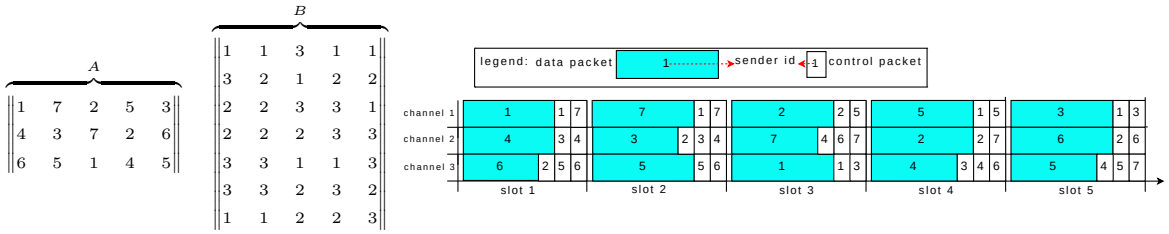


Figure 6.2. *Example of scheduling for the channel allocation A and selection pattern B with $N = 7$, $C = 3$, $S = 5$.*

this information is the MAC address of the node and the list of its intended receivers. For NC^4-DSA, as we discuss in Section 6.3, channel sensing information is additionally included;

- in each slot s, CRs which happen to be in the same channel exchange their control information packets. The objective of this process is to disseminate to all CRs the complete control information $\mathcal{X}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$;
- the channel allocation and selection pattern for a given allocation period are a function of the control information exchanged in the previous period, i.e., $A(t) = f_A(\mathcal{X}(t-1))$ and $B(t) = f_B(\mathcal{X}(t-1))$;
- f_A and f_B are deterministic functions of the control information $\mathcal{X}(t)$. Therefore, if this information is successfully disseminated to all CRs, then the values of $A(t)$ and $B(t)$ determined independently by each CR will be identical, thus implementing a distributed scheme;
- $B(t)$ has pseudo-random properties. Hence, each CR will in general meet with an independent set of CRs in every slot. As a consequence, if S is large enough and if a proper policy to forward the control information is adopted, it is possible to disseminate the control information $\mathcal{X}(t)$ to all CRs by the end of the allocation period t.

6.2.1 Channel allocation and selection pattern

From now on, unless stated otherwise, we will assume that f_A and f_B are performed according to Algorithm 1. Note that since A and B in Algorithm 1 always refer to the same allocation period, the dependence on t is omitted for simplicity. What this algorithm does is to maintain a list L of CRs which are to be allocated a transmission opportunity; the list is initialized with the set of CRs in random order. For every slot and channel pair, the algorithm assigns a transmission opportunity to the first CR in the list which 1) has not already been assigned a transmission opportunity on another channel in the same slot, and 2) has

Algorithm 1 Determination of the channel allocation and selection pattern for NC^4-MAC.

for $i = 1 \dots N$ **do**
  for $s = 1 \dots S$ **do**
    $B_{i,s} \leftarrow 0$;
$U \leftarrow \{1, \dots, N\}$; {list of CRs}
$L \leftarrow \text{Shuffle}(U)$; {list of next CRs to be allocated}
for $s = 1 \dots S$ **do**
  for $k = 1 \dots C$ **do**
    if $|L| < N$ **then**
      {concatenate the two lists preserving order}
      $L \leftarrow L \cup \text{Shuffle}\,(U)$;
    found $\leftarrow$ false ;
    for all $u \in L$ **do** {consider the elements of L in their order}
      {determine the candidate receivers for u}
      $R \leftarrow \{v \in U \,:\, B_{v,s} = 0,\ u \text{ has a pending packet for } v\}$;
      if $((B_{u,s} = 0) \wedge (R \neq \varnothing))$ **then**
        $v \leftarrow \text{RandomElement}(R)$;
        $A_{k,s} \leftarrow u$;
        $B_{u,s} \leftarrow k$;
        $B_{v,s} \leftarrow k$;
        $L \leftarrow L \setminus \{u\}$
        found $\leftarrow$ true ;
        break ;
    if found $=$ false **then**
      $A_{k,s} \leftarrow 0$;
  for $i = 1 \dots N$ **do**
    if $B_{i,s} = 0$ **then**
      $B_{i,s} \leftarrow \text{RandomInteger}(1, C)$;

data to transmit to another CR which has not already been assigned a transmission opportunity. The CR who has been assigned the transmission opportunity is then removed from the list. Whenever the size of the list falls below a given threshold, another randomized copy of the set of all the CRs is appended to its end. The resulting behavior is that, if all CRs are saturated (i.e., they always have pending packets to transmit), each CR will get in the long run a fair share of transmission opportunities. We will exploit this particular characteristic for the performance evaluation in Section 6.2.2; however, we note that Algorithm 1 will also work in the non-saturated case and assign transmission opportunities only to those CRs who have pending packets. We note that this algorithm could be easily extended to more complex resource allocation techniques, for example to address the time-varying channel conditions that are perceived by each CR; this topic is left as a future research direction.

With reference to Algorithm 1, the functions $\text{RandomInteger}()$, $\text{RandomElement}()$ and

$\mathrm{Shuffle}()$ are to be implemented by means of a pre-determined pseudo random number generator whose output is deterministic with respect to the generator seed; this way, using the same seed, all CRs will determine the same A and B matrices.

The control information needed to run Algorithm 1 consists of the set of CRs participating in the channel allocation, and the set of intended receivers for each CR. Thus, the control packet $x_i(t)$ generated by a generic CR i will consist of a set of unique identifiers denoting CR i and its intended receivers. While for the purpose of describing the algorithm it is convenient to denote CRs by an index $i \in \{1, \ldots, N\}$, using the same identifier is not practical in a real system, since the number of CRs may not be known a priori, and each CR is required to determine its own identifier independently. For this purpose, a more practical approach is to use MAC addresses as identifiers. Moreover, it is convenient to include in the control information also some data to determine the random number generator seed to be used by all CRs; as an example, each CR i can include in its control packet $x_i(t)$ an m-bit string generated locally, and the shared seed could then be determined by summing modulo-2^m all bit strings by all CRs. An important requirement for our scheme to work properly is that dissemination of control information reaches all CRs with high probability. Whenever a particular CR fails to retrieve the control information $\mathcal{X}(t)$ at the end of allocation period t, that CR will determine a wrong channel allocation $A(t+1)$ for the subsequent allocation interval, possibly starting transmission in a slot which was meant to be allocated to another CR. We will refer to this event as *spectrum collision*, and we will refer to the CRs that failed to fully retrieve the control information $\mathcal{X}(t)$ as *misinformed CRs*.

As we discussed in Chapter 4, network coding, NC(2^q) is particularly fit to implement such a reliable and efficient dissemination scheme. Hence, we use it for control information dissemination, i.e., it is the core of the Network Coded Cognitive Control Channel for the proposed scheme. We note that at the end of each allocation period nodes have to both decode the received control packets and determine the channel allocation for the next allocation period. These operations have to be done as fast as possible in order not to affect the system performance. The computational complexity for decoding the control packets is $O(N^3)$ if Gaussian elimination is used, while the channel allocation algorithm requires $O(NCS)$ operations in order to determine the channel allocation and selection patterns for the next allocation period; hence, the overall complexity is $O(N^3 + NCS)$. In most practical use cases S and C are fixed. Hence the complexity can be simply denoted with $O(N^3)$.

In Section 6.2.3 we will investigate how the probability of achieving a successful dissem-

ination of the control information varies with respect to different dissemination strategies as well as scenario parameters, and what is the impact of dissemination strategies with non-negligible dissemination failure probability on the overall system performance.

6.2.2 Spectrum utilization

In the previous section we described the channel allocation algorithm that is run by all CRs for data transmissions in NC^4-MAC and the corresponding control information dissemination strategy. We identified the existence of misinformed CRs that, failing to retrieve the control information, are going to either cause spectrum collisions or simply fail to communicate during the successive allocation period. In the following we quantify the achievable spectrum efficiency under these conditions.

Given N CRs and C channels, the number of parallel communications that can occur in a slot is $\min(\lfloor N/2 \rfloor, C)$, since there are only C channels available and for every used channel there must be at least two CRs (the transmitter and the receiver). Consider a single time/frequency slot. This will contain a successful communication if the following conditions are jointly satisfied:

- the considered channel is chosen for data communications. Note that if $\lfloor N/2 \rfloor \geq C$, then all available channels will be used for data communications, while if $\lfloor N/2 \rfloor < C$ then only $\lfloor N/2 \rfloor$ randomly chosen channels will be used. Therefore, the probability that a certain channel is used is given by $\min(\lfloor N/2 \rfloor, C)/C$;
- both the transmitter and the receiver have correctly retrieved the control information. This happens with probability P_{retr}^2, where P_{retr} is the probability that a generic CR correctly retrieves the control information;
- none of the other CRs transmit in the same slot. As for this event, note the following: a given CR, in order to erroneously transmit in a given time/frequency slot, must be misinformed (with probability $1 - P_{retr}$) and transmit in that specific slot (with probability $P_{tx|\text{misinformed}}$, which will be discussed later). Thus, the probability p that a given CR collides in a specific time/frequency slot is $p = (1 - P_{retr})P_{tx|\text{misinformed}}$.

To summarize, the probability P_{cde} that a given channel at a certain time slot contains a correct data exchange is:

$$P_{cde} = \frac{\min(\lfloor N/2 \rfloor, C)}{C} P_{retr}^2 (1-p)^{(N-2)}. \tag{6.1}$$

Now, let Y_k be a random variable assuming value 1 if channel k contains a correct data exchange, and 0 otherwise. The spectrum utilization is then given by $\eta = (\sum_{k=1}^{C} Y_k)/C$. The average spectrum utilization $E[\eta]$ is then obtained as:

$$E[\eta] = E\left[\frac{\sum_{k=1}^{C} Y_k}{C}\right] = \frac{\sum_{k=1}^{C} E[Y_k]}{C} = \frac{\sum_{k=1}^{C} P_{cde}}{C} = P_{cde}. \quad (6.2)$$

To evaluate Equation 6.1 and Equation 6.2, we need to determine the probability P_{tx} that a particular CR transmits in a given slot. We note that P_{tx} depends on two factors: 1) the behavior of the scheduling algorithm, and 2) the number N_d of CRs for which the CR has decoded the control information. For an informed CR, we have $N_d = N$ by definition, and $P_{tx|\text{informed}}$ can be precisely characterized as follows. If $\lfloor N/2 \rfloor \leq C$, Algorithm 1 would randomly select $\lfloor N/2 \rfloor$ transmitters among the N CRs, thus the probability that a given CR is selected as a transmitter and assigned to the channel is $\lfloor N/2 \rfloor / N \cdot 1/C$; if instead $\lfloor N/2 \rfloor > C$, there will be C transmitters among the N CRs so that the probability that the CR is selected as a transmitter and assigned to the channel is $C/N \cdot 1/C = 1/N$. Therefore, we have that $P_{tx|\text{informed}} = \min(\lfloor N/2 \rfloor, C)/(NC)$.

Equivalently, following the same reasoning and substituting N with N_d, for a misinformed CR we have that $P_{tx|\text{misinformed}} = \min(\lfloor N_d/2 \rfloor, C)/(N_d C)$; the problem is that N_d depends on the network coding strategy and on system parameters such as S, C and N, and therefore it is not straightforward to give it a precise analytical characterization. However, we can derive a lower bound $P^{lb}_{tx|\text{misinformed}}$ and an upper bound $P^{ub}_{tx|\text{misinformed}}$ on $P_{tx|\text{misinformed}}$, which used in conjunction with Equation 6.1 and Equation 6.2 will yield respectively an upper and lower bound on $E[\eta]$. Note that an upper bound on $P_{tx|\text{misinformed}}$ translates into a lower bound on $E[\eta]$ because a higher transmission probability by misinformed CRs yields more spectrum collisions and, in turn, a lower successful spectrum utilization by informed CRs. For the same reason, a lower bound on $P_{tx|\text{misinformed}}$ yields an upper bound on $E[\eta]$. In particular, we have $P^{lb}_{tx|\text{misinformed}} = P_{tx|\text{informed}}$, since for misinformed CRs we always have $N_d < N$ by definition; furthermore, since $\lfloor N_d/2 \rfloor / N_d \leq 1/2$, we have $P^{ub}_{tx|\text{misinformed}} = 1/2$.

The resulting upper and lower bounds of $E[\eta]$ are reported in Figure 6.3 versus the control information retrieval probability for $C = 10$ and different values of N. In all cases, the channel utilization is highest for $P_{retr} = 1$, and decreases as P_{retr} decreases.

As a general consideration, we note that in those cases where $N < 2C$ (i.e., $N = \{5, 10, 15\}$ in the figure), the maximum channel utilization is limited to $\lfloor N/2 \rfloor / C$, since not all channels

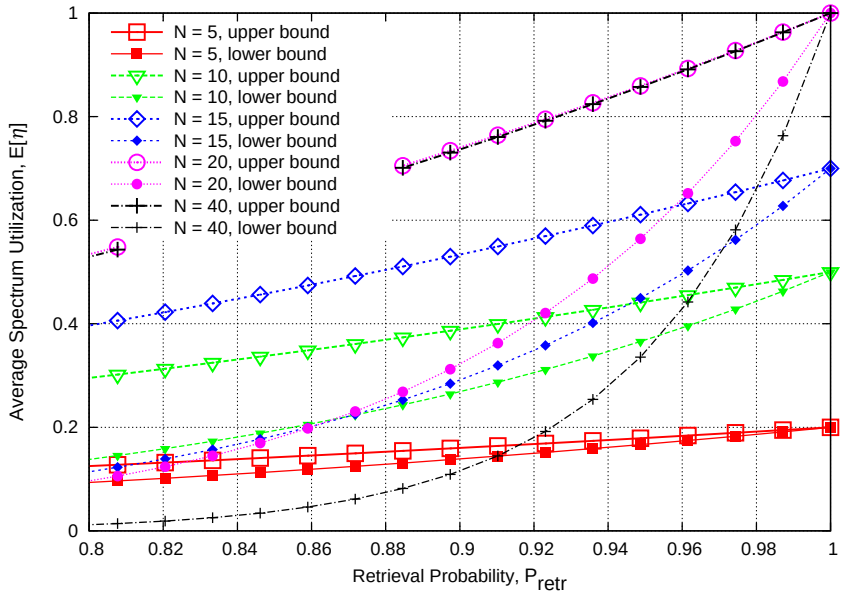


Figure 6.3. *Average spectrum utilization vs. retrieval probability,* $C = 10$.

can be exploited in this case; conversely, full channel utilization can be achieved for all cases in which $N \geq 2C$ (i.e., $N = \{20, 40\}$ in the figure).

We observe that the upper and lower bounds on $E[\eta]$ are relatively close for low values of N, but the difference gets more significant as N increases; furthermore, $\forall N$, when $P_{retr} \rightarrow 1$ the two bounds converge to the same value, as the effect of spectrum collisions becomes negligible. Still, since there are several cases of interest in which the bounds are not close, all references to $E[\eta]$ in the remainder of this chapter will refer to its lower bound and hence result in a worst-case analysis.

Finally, we note that the maximum average spectrum utilization $E[\eta]$ is determined based on the assumption that nodes are saturated, i.e., each node always has a pending packet to transmit; this is a commonly made assumption when analyzing the performance of MAC schemes. Algorithm 1 will still work correctly if this assumption is not satisfied, however the achieved spectrum utilization will be lower than $E[\eta]$ if the number of CRs with pending packets is not large enough. An accurate performance evaluation of this scenario is left as a future study.

6.2.3 Dissemination of control information

The results presented in the previous section highlight the importance of using a dissemination scheme for the control information that yields a high P_{retr}. Existing MAC pro-

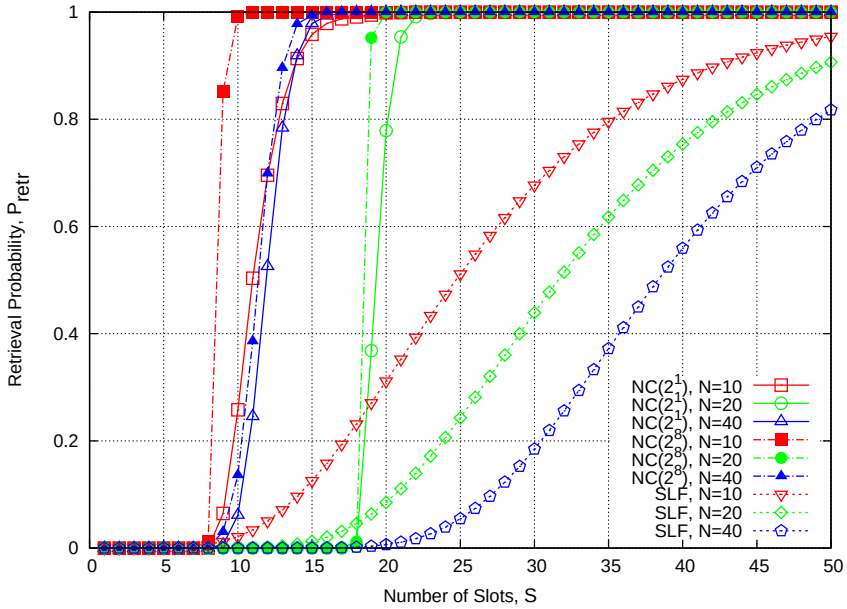


Figure 6.4. *Retrieval probability vs. number of slots,* $C = 10$.

posals such as [77] and [78] disseminate control information just by having all CRs transmit their own control information in each slot; which we introduced in Section 4.2.2 as the SLF scheme and in Section 4.4.1 showed that for SCSMA and SALOHA MAC protocols it required a considerable number of slots to broadcast information to all CRs; to cope with this issue, we investigated the use of network coding to perform a more efficient dissemination.

In the following we will use SLF as the baseline scheme. We note that, unlike the broadcasting protocols introduced in Section 4.2.2 where at most one node may transmit in a given slot, the dissemination strategies (both NC and SLF) considered to implement the control channel for NC^4-MAC allow multiple nodes to transmit in the same slot and channel (refer to the example of Figure 6.2). According to SLF, CRs transmit their control information on a randomly chosen channel; therefore, for two generic CRs j and i, CR j will not receive the control information generated by i in a single slot with probability $(C-1)/C$. The retrieval probability P_{retr} after s slots is then given by:

$$P_{retr} = \left(1 - \left(\frac{C-1}{C}\right)^{s}\right)^{N-1}. \tag{6.3}$$

We note that this is actually an upper bound on the retrieval probability, since it is derived assuming that the impact of spectrum collisions is negligible. As for NC(2^q), in Section 4.3.3 we showed that it is not straightforward to analytically derive its dissemination perfor-

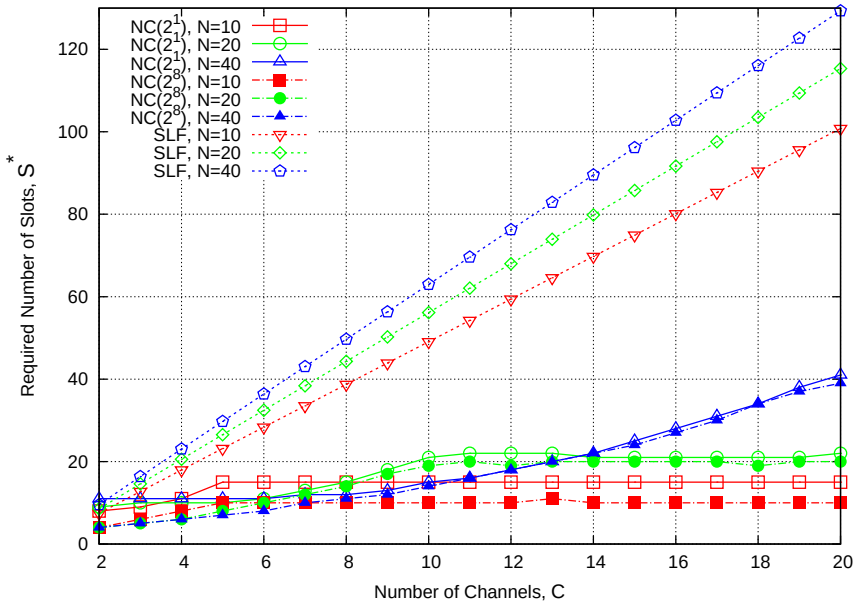


Figure 6.5. *Number of slots needed to have $P_{retr} = 0.95$ vs. number of channels.*

mance. For NC^4 this task becomes even more challenging as all CRs that happen to be in the same channel in a given slot exchange their control packets, increasing the correlation between the exchanged information. Hence, in the following we derive its performance by means of simulations.

The obtained performance is shown in Figure 6.4 and Figure 6.5, where we plot the retrieval probability P_{retr} obtained by simulating dissemination with network coding over GF(2^1) and GF(2^8), and compare it with the retrieval probability of the SLF scheme obtained from Equation 6.3. In detail, Figure 6.4 reports P_{retr} for $C = 10$ and $N \in \{10, 20, 40\}$. In all cases the use of network coding yields a high P_{retr} for significantly fewer slots than the SLF scheme. As expected, NC(2^8) provides a steeper increase in P_{retr} than NC(2^1); however, it is to be noted that this comes with an increase in overhead and decoding complexity. Finally, we note that the performance for both $N = 10$ and $N = 40$ is better than for $N = 20$. To explain this, we report in Figure 6.5 the number S^* of slots required to achieve $P_{retr} = 0.95$ as a function of C. What happens is that, when $C < N/2$ (left portion of the figure), there are on average many CRs in every channel, thus the connectivity of the network in every slot is high, and dissemination is quick; when C increases, the connectivity decreases, and the number of slots required to achieve a high P_{retr} increases. When $C > N/2$ (right portion of the figure), there are unused channels, and Algorithm 1 will result in the use of only $\lfloor N/2 \rfloor$ channels, with two CRs in each channel. Thus, the connectivity of the network reaches its

lower limit for $C = N/2$, and does not decrease for increasing C; for this reason, the number of slots required to achieve a certain P_{retr} reaches its maximum for $C = N/2$. We note that the SLF scheme selects channels randomly, so that as C/N increases it is more likely that CRs are alone in a channel, sending control messages to nobody; this makes the performance of the SLF scheme degrade linearly with C/N. The linearity with respect to C can be inferred directly from Figure 6.5; the linearity with respect to $1/N$ can be inferred from an alternative representation of the same data.

6.2.4 Goodput performance

In order to determine the goodput of the proposed scheme we need to account not only for spectrum collisions by misinformed CRs, but also for the overhead due to the exchange of control information. For this purpose, let T_{ctrl} be the duration of the transmission of a control packet. Since each CR will send exactly one control packet per control slot in an allocation period, the total overhead time T_O spent by all the CRs in all channels during an allocation period is given by $T_O = NST_{ctrl}$. We can therefore determine the goodput of the system as:

$$G = \frac{CT_{all} - T_O}{CT_{all}} E[\eta] \tag{6.4}$$

where $E[\eta]$ is given by Equation 6.2, which is calculated for each scheme using the characterization of P_{retr} presented in the previous subsection. We remind that $E[\eta]$ depends on N, C and P_{retr}, and that P_{retr} depends on N, C, S and the chosen dissemination scheme. Note that $G < 0$ for $T_O > CT_{all}$; this reflects the fact that the scheme is not feasible under this condition, as the time required to transmit all the control information would exceed the available spectrum resources. Furthermore, $\lim_{T_{all}\to\infty} G = E[\eta]$; the scheme approaches its maximum efficiency as T_{all} increases, since the impact of the control information overhead to the goodput becomes negligible. In Figure 6.6 we report the goodput performance obtained in scenarios with different values of N and different control dissemination schemes. In the following we only discuss results for $C = 10$ highlighting that the conclusions that follow also hold for other values of C. The performance obtained by using network coding is almost always very good for $S = 30$; using smaller values of S yields a slight improvement in performance in some cases (e.g., $N = 40$ in the figure) due to the decrease in overhead, but results in an extremely poor performance in other cases ($N = 20$, NC(2^1)) due to poor performance of the control information dissemination. The use of NC(2^8) instead of NC(2^1)

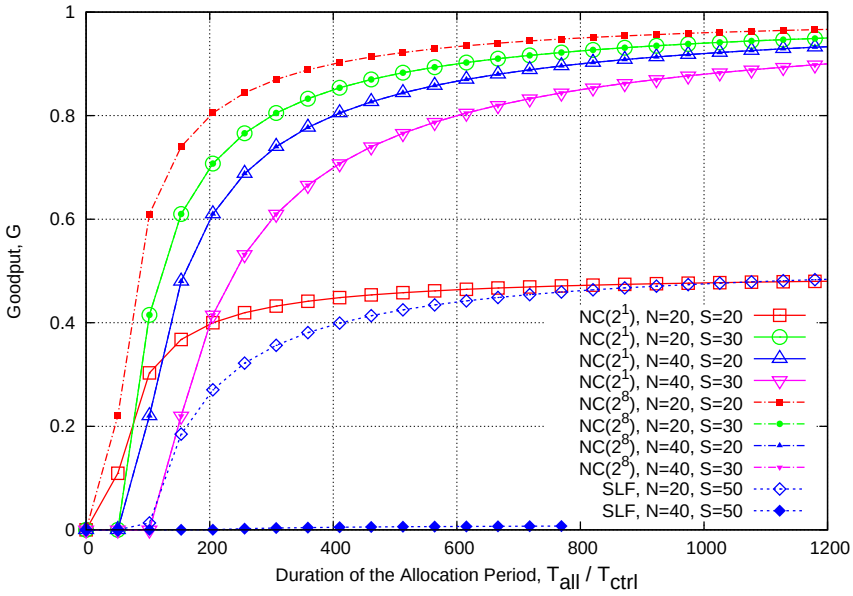


Figure 6.6. *Goodput obtained with different coding schemes.*

is greatly beneficial when a small value of S is chosen, although it is to be noted that, when a higher value of S is adopted, the two coding schemes perform identically. Actually, all other parameters being equal, NC(2^8) would need a T_{ctrl} slightly higher than NC(2^1) due to the more complex representation of the coding vector. The SLF scheme always achieves worse performance compared to the network coding schemes. This is primarily due to the poor control information dissemination performance of the SLF scheme, and is the reason why it can achieve acceptable performance only with a significantly higher value of S than the network coding scheme.

To summarize, the proposed scheme performs very close to the maximum achievable goodput for sufficient T_{all}/T_{ctrl}. To understand what this means in practice we need some figures for T_{ctrl} and the PHY rate of the considered system. As an example, the control information we use for NC^4-MAC would fit in a 270 bytes packet to support $N = 40$; this would result in $T_{ctrl} = 360$ μs using a PHY rate of 6 Mbps. We consider that a control packet needs to include up to 40 node identifiers for the coding vector, plus 5 additional identifiers for the intended receiver; as a worst case, each identifier is assumed to be represented with a MAC address of 6 bytes each. For the PHY rate, we refer to the lowest rate in IEEE 802.11g. Note that PHY overhead is not considered in our calculations. In this case, using $T_{all}/T_{ctrl} = 600$ to achieve a reasonable goodput efficiency would require an allocation period of duration $T_{all} = 0.2$s, which for most applications should be sufficiently low to track system dynamics

such as CRs entering and leaving the system, or changes in the available spectrum resources.

6.3 NC^4-DSA

The NC^4-MAC scheme discussed previously is particularly fit for an extension to DSA scenarios for many reasons. First of all, CRs visit all channels in a pseudo-random fashion providing an interesting opportunity to perform PU detection. Moreover, the NC^4 is an effective means to make CRs cooperate, e.g., by exchanging detection information. Finally, it is straightforward to avoid transmission in channels occupied by PUs: it is sufficient to take into account the detection information gathered in the previous allocation period for the determination of the channel allocation and selection patterns for the next allocation period.

The use of cooperation is expected to improve significantly the detection capabilities with respect to what a single CR could do on its own. This implies that it is possible to successfully adopt simple detection techniques such as ED, which are normally not effective for individual detection due to their relatively poor performance. The benefit with respect to the use of more sophisticated techniques, such as CFD, is that the required sensing time is significantly reduced, and the complexity and cost of the detector are minimal.

Hence, in this section we discuss NC^4-DSA which most important aspects are the following:

- in each allocation period, PU detection is performed over all channels, to track the varying pattern of PUs activity;
- the detection information gathered by each CR during an allocation period is to be disseminated to all CRs using the control channel;
- CD is carried out independently by each CR using the same deterministic algorithm; hence, all CRs which correctly decoded the control information will be able to infer the same set of free channels;
- the resource allocation algorithm, which is run independently by each CR, assigns transmission opportunities only on free channels; however, some CRs will still be instructed to switch to busy channels for PU detection purposes.

In detail, NC^4-DSA operates over three subsequent allocation periods. In the first allocation period t, at the beginning of every generic slot s, each CR i will perform an attempt

to detect the presence of PUs. Since all the CRs are synchronized at the slot level, they will all perform detection synchronously, and only the PU can be transmitting when detection is performed. Thanks to the use of ED, the time required to perform PU detection is on the order of $1/B$, where B is the channel bandwidth [86], and hence the detection overhead is almost negligible. Let now $d_{i,s}(t) \in \{0, 1\}$ be the output of a detection attempt, where $d_{i,s}(t) = 1$ if CR i detects the presence of a PU in slot s of the allocation period t (positive detection), and $d_{i,s}(t) = 0$ otherwise (negative detection). Note that this definition is independent of whether the PU is actually present or not. This detection information, gathered in the allocation period t, is then disseminated to all CRs during the allocation period $t + 1$. After dissemination, and just prior to determining the channel allocation and switch pattern, CD is performed for every channel k. This is done by counting the number of positive detections $D_k(t)$ as:

$$D_k(t) = \sum_{i=1}^{N} \sum_{s:B_{i,s}(t)=k} d_{i,s}(t) \quad . \tag{6.5}$$

If $D_k(t)$ is greater than a pre-defined threshold Q, then it is inferred that channel k is being used by a PU. This way, the set $\mathcal{C}_{free}$ of free channels available for secondary access is determined, and can then be used to determine the channel allocation $A(t+2)$ and selection pattern $B(t + 2)$ for allocation period $t + 2$. It is worth mentioning that such a CD scheme suffers when there are misbehaving or malicious CRs as they may purposely introduce false alarm situations leading to a decrease in network performance. In the following we assume the absence of such malicious CRs, and refer the reader to [81, 82] for a discussion of some methods to deal with this type of behavior.

The choice of the channel selection pattern is very important for PU detection, since it determines not only on what channels PU detection is carried out, but also how many decision attempts per channel are performed. As a consequence, even if only free channels can be used for CR communications, all channels are to be included into the channel switch pattern, so that detection is performed even on channels that have been reported to be busy in the past. Of course, CRs switching to a channel which is known to be busy will not perform any transmission, i.e., they will not disseminate control information on that channel; for this reason, the performance of the dissemination process will degrade when the fraction of busy channels increases. We also need to account that, for $N/2 < |\mathcal{C}_{free}|$, if all free channels are used, then there will be no CR left to perform detection in busy channels; to overcome this issue, in every slot some CRs must abstain from data communication. The choice of the exact

number of CRs assigned to this task creates a trade-off between PU detection performance on one side and dissemination and spectrum reuse efficiency on the other.

The solution that we propose in this section is detailed as Algorithm 2. Note that A and B in Algorithm 2 always refer to $A(t+2)$ and $B(t+2)$, and D_k always refers to $D_k(t)$; the dependence on t is omitted for brevity. This algorithm is a modified version of Algorithm 1, which we proposed in Section 6.2.1 for NC^4-MAC. The main modification is that in each time slot a fraction equal to $1 - |\mathcal{C}_{free}|/C$ of the CRs will always be instructed to switch to a busy channel for PU detection purposes; the rest of the CRs are eligible for transmission opportunities on free channels, and even if they are not involved in data transmission they will still switch to a randomly selected free channel to do PU detection. The resulting behavior is that on average all channels (both free and busy) get the same number of detection attempts. The investigation of more complex strategies in which busy channels are subject to a different number of detection attempts than free channels is left as a future study.

6.3.1 Dissemination of control information

We now investigate the effectiveness of NC^4 in DSA scenarios. As we discussed previously, the main issue is that the channel switch pattern needs to cover all channels, so that PU detection can be performed effectively, but the control information cannot be transmitted by CRs on those channels in which PUs are active. This causes a degradation of the dissemination performance of NC^4-DSA with respect to NC^4-MAC. In order to quantify this performance degradation, we ran some simulations in the single hop scenario where we have C licensed channels, N_p PUs, each operating in a distinct channel, and N CRs trying to opportunistically access the free channels, using Algorithm 2 to determine the channel selection and transmission scheduling. For this first evaluation case, we assume that PUs are always active and that N_p is known to CRs.

In Figure 6.7 we report the results obtained for $C = 10$ and different values of N, plotting the number of slots, S^* necessary to reach a given value of P_{retr} as a function of N_p. The "trend" curves are obtained multiplying the function $f(N_p) = C/(C - N_p)$ by the value of S^* obtained for the case $N_p = 0$. That is, the performance degradation of S^* due to the increase of N_p is approximately inversely proportional to the fraction of channels available for secondary communications, given by $(C - N_p)/C$.

For given values of C and N, S^* is constant for $N_p \leq C - N/2$, i.e., whenever the number of free channels is greater than or equal to $N/2$. This phenomenon, which can be observed in

Algorithm 2 Determination of the channel allocation and selection pattern for NC^4-DSA.

{D_k initialized according to Equation 6.5 $\forall k$; Q is a pre-defined threshold}
$\mathcal{C}_{free} \leftarrow \{k : D_k \leq Q\}$
for $i = 1 \ldots N$ **do**
  for $s = 1 \ldots S$ **do**
    $B_{i,s} \leftarrow 0$;
$U \leftarrow \{1, \ldots, N\}$; {list of CRs}
$L \leftarrow \text{Shuffle}(U)$; {list of next CRs to be allocated}
for $s = 1 \ldots S$ **do**
  $\mathcal{C}_{busy} \leftarrow \{1, \ldots, C\} \setminus \mathcal{C}_{free}$;
  $\mathcal{C}_{data} \leftarrow \varnothing$;
  $\mathcal{C}_{detect} \leftarrow \varnothing$;
  $i \leftarrow 1$;
  while $(i < N|\mathcal{C}_{free}|/C) \wedge (\mathcal{C}_{free} \setminus \mathcal{C}_{data} \neq \varnothing)$ **do**
    $\mathcal{C}_{data} \leftarrow \mathcal{C}_{data} \cup \{\text{RandomElement}(\mathcal{C}_{free} \setminus \mathcal{C}_{data})\}$;
    $i \leftarrow i + 2$; {each data transmission needs 2 CRs}
  while $i < N(1 - |\mathcal{C}_{free}|/C) \wedge (\mathcal{C}_{busy} \setminus \mathcal{C}_{detect} \neq \varnothing)$ **do**
    $\mathcal{C}_{detect} \leftarrow \mathcal{C}_{detect} \cup \{\text{RandomElement}(\mathcal{C}_{busy} \setminus \mathcal{C}_{detect})\}$;
  for $k \in \mathcal{C}_{data}$ **do**
    if $|L| < N$ **then**
      {concatenate the two lists preserving order}
      $L \leftarrow L \cup \text{Shuffle}\,(U)$;
    found $\leftarrow$ false ;
    for all $u \in L$ **do** {consider the elements of L in their order}
      {determine the candidate receivers for u}
      $R \leftarrow \{v \in U : B_{v,s} = 0,\ u$ has a pending packet for $v\}$;
      if $((B_{u,s} = 0) \wedge (R \neq \varnothing))$ **then**
        $v \leftarrow \text{RandomElement}(R)$;
        $A_{k,s} \leftarrow u$;
        $B_{u,s} \leftarrow k$;
        $B_{v,s} \leftarrow k$;
        $L \leftarrow L \setminus \{u\}$
        found $\leftarrow$ true ;
        break ;
    if found = false **then**
      $A_{k,s} \leftarrow 0$;
  for $i = 1 \ldots N$ **do**
    if $B_{i,s} = 0$ **then**
      if $\mathcal{C}_{detect} \neq \varnothing$ **then**
        $B_{i,s} \leftarrow \text{RandomElement}(\mathcal{C}_{detect})$
        $\mathcal{C}_{detect} \leftarrow \mathcal{C}_{detect} \setminus \{B_{i,s}\}$
      else
        $B_{i,s} \leftarrow \text{RandomInteger}(1, C)$;

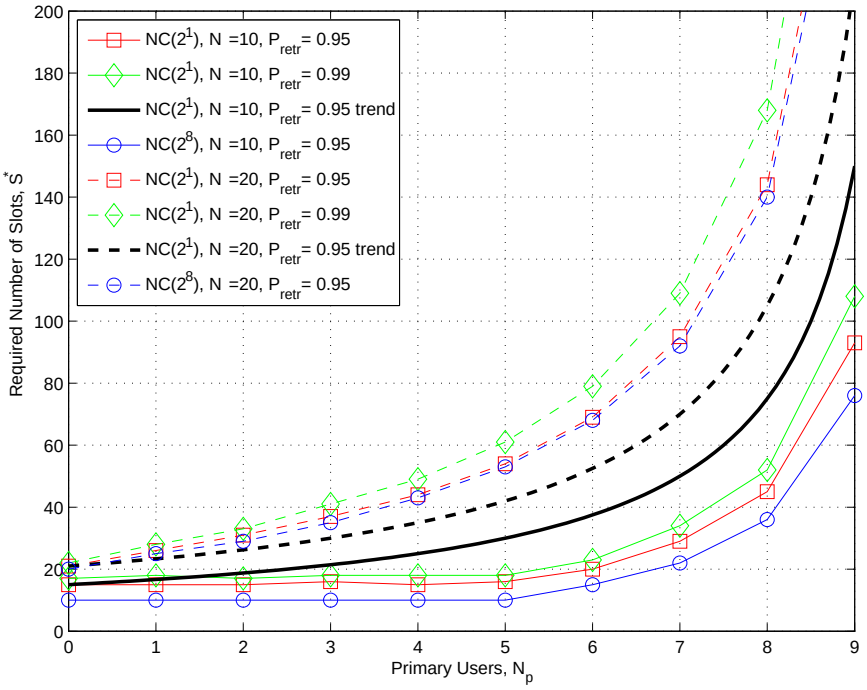


Figure 6.7. *Number of slots required for* $P_{retr} = 0.95$ *and* 0.99 *vs. number of PUs for different coding schemes and number of CRs.*

the curve for $N = 10$, is caused by the channel selection algorithm which attempts to place in free channels at least two CRs for the purpose of exchanging control packets. Moreover, for any fixed value of N_p, S^* is maximum for $N = 2C$, for the same reasons already discussed in Section 6.2.3. From Figure 6.7 we also note that a $P_{retr} = 0.99$ can be obtained with a small increase in S^*. This is due to the waterfall behavior of the P_{retr} curve with respect to S when network coding is used. The difference in S^* becomes higher as the number of active PUs increases. As an example, with $N_p = 4$ the value of S that assures $P_{retr} = 0.95$ is $S^* = 44$, while for $P_{retr} = 0.99$ it is only slightly higher ($S^* = 49$).

Focusing on the network coding technique, we observe that there is no substantial difference between NC(2^1) and NC(2^8) in terms of number of slots required to assure a certain retrieval probability. As is shown in the figure, network coding over GF(2^8) performs slightly better for few CRs and almost the same in other cases. To summarize, NC^4 is effective for the dissemination of control information in DSA scenarios, provided that the S parameter is properly dimensioned with respect to the expected spectrum availability for CR communication.

6.3.2 Primary user detection

We now analyze the PU detection performance of the CD scheme described previously. Consider a single channel, and let X_1 and X_0 denote the event that a PU is respectively present or absent in that channel. Let P_d and P_f be the single attempt probabilities of detection and false alarm, respectively. Due to the channel selection strategy adopted by CRs, in each slot every CR will select the considered channel with probability $1/C$ which is an approximation of the behavior of Algorithm 2; if the channel is actually chosen, the detection attempt by that CR will be positive with probability P_d or P_f, depending on whether X_1 or X_0 happens, respectively.

As a result, conditioned on X_1, the total number n_d of successful PU detections in a given channel and allocation period follows a binomial distribution:

$$\Pr\left(n_d|X_1\right) = \binom{SN}{n_d}\left(\frac{P_d}{C}\right)^{n_d}\left(1-\frac{P_d}{C}\right)^{SN-n_d}. \tag{6.6}$$

Similarly, conditioned on X_0, the total number of false alarms n_f in a given channel and allocation period is also binomially distributed:

$$\Pr\left(n_f|X_0\right) = \binom{SN}{n_f}\left(\frac{P_f}{C}\right)^{n_f}\left(1-\frac{P_f}{C}\right)^{SN-n_f}. \tag{6.7}$$

We recall that CD is performed by comparing the total count D of detections reported by all the CRs with a pre-defined threshold Q: if $D \geq Q$ then it is inferred that the PU is present, otherwise the channel is considered free. Let P_{cd} and P_{cf} denote, respectively, the probability of correct detection and false alarm with a given value of Q, using the CD strategy described above. We have:

$$P_{cd} = \sum_{n_d=Q}^{SN}\Pr\left(n_d|X_1\right), \quad P_{cf} = \sum_{n_f=Q}^{SN}\Pr\left(n_f|X_0\right). \tag{6.8}$$

Of course, in order to evaluate Equation 6.8 one needs to know the figures for a single detection attempt, i.e., P_d and P_f. For this purpose, we adopt the model for the energy detection of unknown signals in Additive White Gaussian Noise (AWGN) channels, which is discussed in [83]. We note that other detection techniques could be evaluated by just using a different model for P_d and P_f. We calculate the performance of cooperative PU detection and compare it with the performance of a single detection attempt by a single CR. The single attempt performance is obtained by varying the energy threshold parameter λ. The CD performance is obtained by having all CRs perform detection attempts with a fixed threshold $\lambda_{CD} = 5$ dB

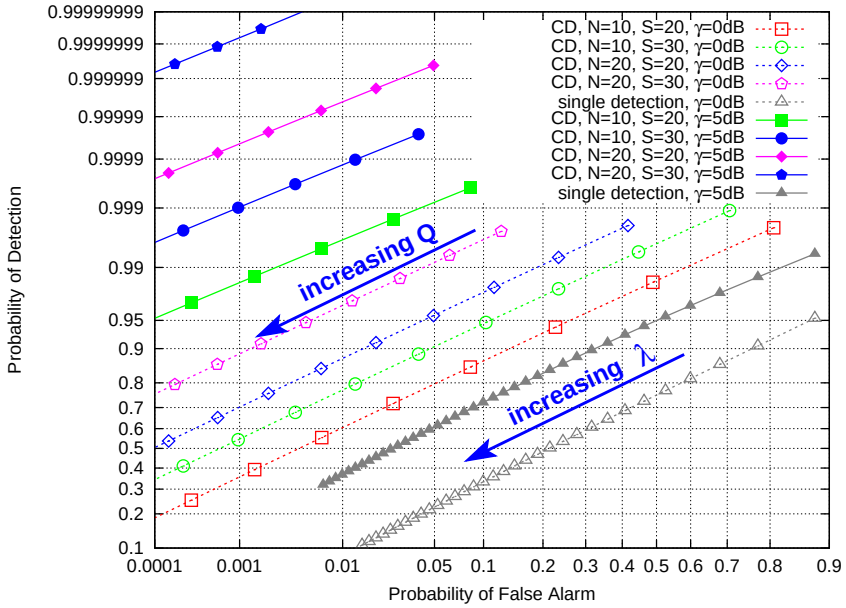


Figure 6.8. *PU detection performance.*

and vary the detection count threshold Q. For all calculations, we set the time-bandwidth parameter $TW = 1$, as this is the value that yields the best performance [86].

The obtained results are reported in Figure 6.8 for values of the Signal to Noise Ratio (SNR) of the PU $\gamma \in \{0, 5\}$ dB. In the figure, the Normal Probability scale is used for both axes; this choice was made to conform with the scale used in [86]. The results show that our CD strategy allows to achieve significant improvements in the achievable trade-offs between PU detection and false alarm probability. Furthermore, we note that the detection improves with increasing N and S, thanks to the increase in the total number of detection attempts.

6.3.3 Primary activity and secondary access

The intended result of the PU detection procedure is that CRs access all channels in which the PU is not active, while avoiding channels with PU activity. Unfortunately, it is possible that CRs end up transmitting in channels actually occupied by PUs, or that an unused channel is undetected and therefore some spectrum resources are wasted. These errors can occur not only due to missed detections and false alarms, but also because of PU activation/deactivation between the detection process and the spectrum access process. In this section we will evaluate the joint impact of these aspects on the communication performance of PUs and CRs.

We model the activation process of each PU in a generic channel as a two-state Markov

chain, with state transitions performed at allocation period boundaries. This model has been widely adopted in the DSA literature [79, 82, 87–89]. State 0 represents inactivity while state 1 represents activity. We denote with $P_{xy}^{(z)}$ the z-step transition probability from state x to state y, and we denote $P_{xy}^{(1)}$ as P_{xy} for brevity. The Markov chain is completely specified by the one-step transition matrix P given by:

$$P = \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix} = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}. \tag{6.9}$$

For $\alpha, \beta \in (0,1)$ the Markov chain is regular, and the steady state probabilities of being in state 0 and 1 are $\pi_0 = \beta/(\alpha+\beta)$ and $\pi_1 = \alpha/(\alpha+\beta)$, respectively.

Since two allocation periods elapse between detection and spectrum access, the success of secondary spectrum access depends on the two-step transition probability matrix $P^{(2)}$ and on the correct detection and false alarm probabilities P_{cd} and P_{cf}. In detail, CRs interfere with a PU when either a) the PU is inactive upon detection, there is no false alarm, and the PU becomes active when secondary access by CRs is performed, or b) the PU is active upon detection, detection fails and the PU is still active when secondary access is performed. We assume that the parameters of the system have been chosen such that P_{retr} is almost 1, so that the effect of spectrum collisions is negligible. The probability P_{interf} that the CRs interfere with a PU, conditioned on the fact that the PU is active when secondary access is performed, is therefore given by:

$$P_{interf} = \frac{\pi_0 \left(1 - P_{cf}\right) P_{01}^{(2)} + \pi_1 \left(1 - P_{cd}\right) P_{11}^{(2)}}{\pi_1}. \tag{6.10}$$

Similarly, an unused channel is successfully used by CRs when either c) the PU is inactive upon detection, detection is performed correctly, and the PU is still inactive by the time spectrum access is performed, or d) the PU is active, detection fails, and the PU becomes inactive before the CRs start using the channel. The probability P_{reuse} that the CRs successfully reuse an unused channel, conditioned on the fact that the channel is actually unused by the PU when secondary access is to be performed, is then given by:

$$P_{reuse} = \frac{\pi_0 \left(1 - P_{cf}\right) P_{00}^{(2)} + \pi_1 \left(1 - P_{cd}\right) P_{10}^{(2)}}{\pi_0}. \tag{6.11}$$

Furthermore, the limit performance that can be achieved by improving the detection capa-

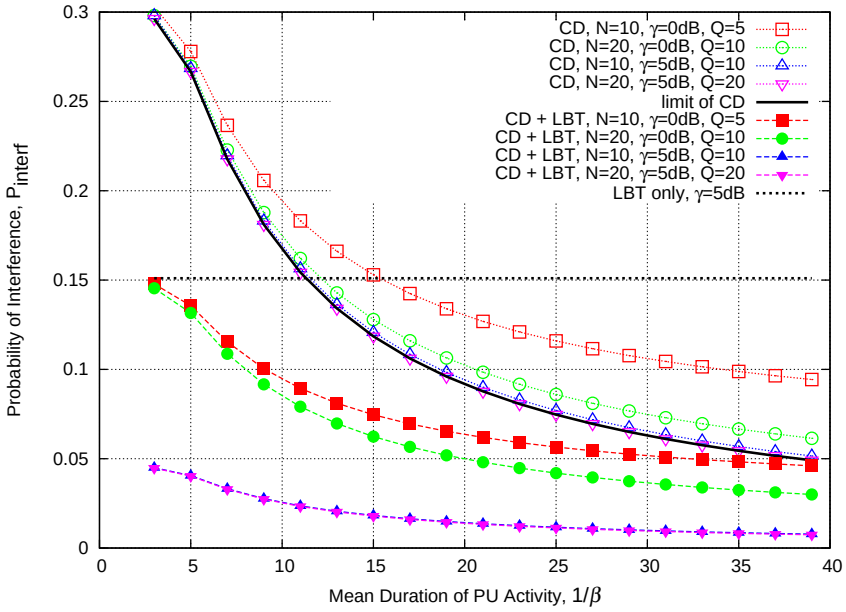


Figure 6.9. *Probability of interference to the PU caused by CRs.*

bilities of the network of CRs is given by:

$$\overline{P}_{interf} = \lim_{P_{cd}\to 1, P_{cf}\to 0} P_{interf} = \frac{\pi_0}{\pi_1} P_{01}^{(2)}, \tag{6.12}$$

$$\overline{P}_{reuse} = \lim_{P_{cd}\to 1, P_{cf}\to 0} P_{reuse} = P_{00}^{(2)}. \tag{6.13}$$

The resulting performance for secondary spectrum access is represented by the "CD" curves in Figure 6.9 and Figure 6.10, where we plot respectively P_{interf} and P_{reuse} as a function of the mean activity duration $\ell_1 = 1/\beta$ and inactivity duration $\ell_0 = 1/\alpha$ of the PU respectively, for a scenario with $C = 10$, $\pi_1 = 0.7$ and $\lambda_{CD} = 5$ dB. For each combination of N and γ a value of the threshold count parameter Q was chosen to yield a good tradeoff between P_{cd} and P_{cf}; the chosen value is reported in the figure. The resulting values of P_{cd} and P_{cf} can be determined using Equation 6.8.

From Figure 6.9 it results that P_{interf} decreases when ℓ_1 increases, since the longer activation period of the PU allows CRs to detect it and avoid interfering with it. Our results show that for most values of N and γ the performance obtained by the cooperative secondary spectrum access scheme is very close to the limit performance of Equation 6.12 and Equation 6.13; in particular, for $\gamma = 5$ dB this performance almost reaches the limit, thanks to the rather good chances in the single detection attempt which result in practically perfect CD. The only case in which the interference is significantly higher than the lower bound is the case with $N = 10$ and $\gamma = 0$ dB, in which the probability of success of a single detection

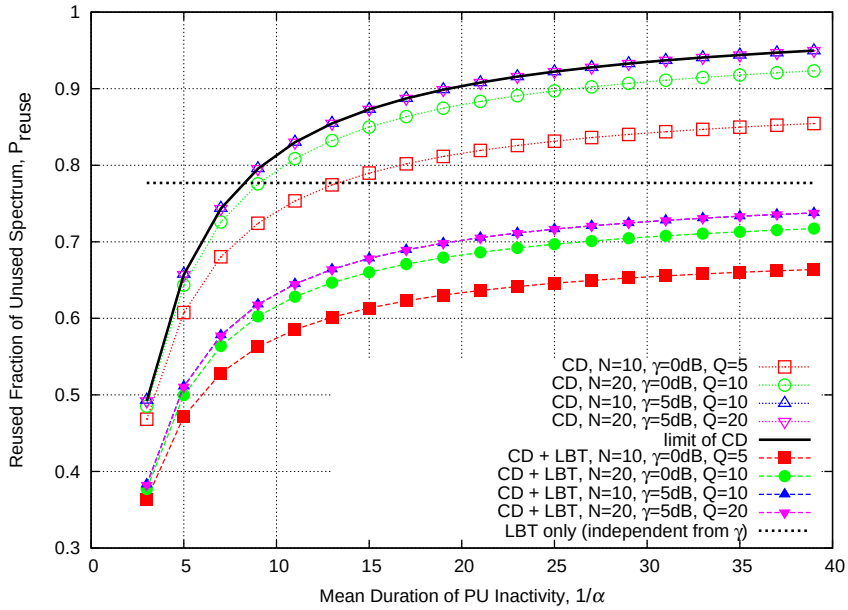


Figure 6.10. *Efficiency of secondary reuse of spectrum unused by the PUs.*

attempt is very low due to the low SNR of the PU, and the benefits of cooperation are limited due to the small number of CRs.

From Figure 6.10 it is evident that also the probability of successfully reusing unused spectrum depends on (in this case, increases with) the mean duration of the inactive period of the PU. This is due to the fact that the secondary spectrum access scheme is more successful when the PU stays inactive for longer periods. Again, the performance is almost optimal when $\gamma = 5$ dB, and is in general rather close to the performance limit, with the exception of the case $N = 10$ and $\gamma = 0$ dB which suffers from poor performance of both single detection and CD. We remark that both P_{interf} and P_{reuse} have a very weak dependence on the steady state activation probability π_1 of the PU; as an example, if we change the value of π_1 from 0.7 to 0.3, the relative difference in the observed P_{interf} is less than 10% for $\ell_1 > 10$, and the same holds for P_{reuse} when $\ell_0 > 10$. In other words, it is not very important how much the PU is active in the long term, but rather how often it changes its status.

Overall, the PU interference probability is rather high ($\sim 5\%$ for $\ell_1 = 40$). As a consequence, while the proposed CD scheme is suitable for very slowly varying PUs (such as TV stations which likely have ℓ_1 on the order of thousands), it is clearly not effective for faster varying PUs (ℓ_1 on the order of tens). The point is that, with a delay of two allocation periods between PU detection and spectrum access, the probability of interfering with the PU is dominated by the two-step PU activation probability, as evident from Equation 6.12. To cope

with this issue, we add a Listen Before Talk (LBT) feature to NC^4-DSA. When using LBT, a CR which has been assigned a transmission opportunity will perform an additional single detection attempt just prior to starting transmission, and actually transmit only if the outcome of this detection is negative. We propose to use LBT in addition to CD; i.e., transmissions on a given channel are performed only if the channel is estimated to be free according to the outcome both of CD based on previously disseminated data and of the additional single CR detection attempt. The reason for this choice is that the additional detection attempt is less accurate than CD, but is based on more recent data, and for this reason it is more effective in tackling PU activation and deactivation issues; it is therefore convenient to adopt both strategies simultaneously. Due to the different way in which these detection strategies work, for LBT we chose to use an energy threshold $\lambda_{LBT} = 3$ dB, different than λ_{CD}. From Figure 6.9 and Figure 6.10 it is evident that the use of LBT in addition to CD provides a significant reduction of PU interference, though at the cost of a reduction in the spectrum reuse efficiency, caused by the relatively high false alarm probability of LBT. Note that the utilization of LBT only (without CD) can achieve good efficiency in terms of spectrum reuse, but fails to avoid PUs activity leading to a higher interference probability. Lowering λ_{LBT} to cope with this issue would greatly harm the spectrum reuse efficiency. This is due to the overall poor performance of single-attempt ED, which we already discussed in Section 6.3.2.

6.3.4 Goodput of secondary access

To conclude, we determine the goodput performance of NC^4-DSA, for both the original "CD" version and the "CD + LBT" variant. In doing this, we will account for all the aspects that we analyzed in the previous sections, i.e., the effectiveness of control information dissemination, the channel utilization efficiency among CRs, the control information overhead, the PU detection performance and the unoccupied spectrum reuse efficiency.

In detail, in Section 6.3.1 we evaluated by means of simulations the retrieval probability conditioned to the number N_p of active PUs. We denote this probability as $P_{retr|N_p}$. We assume that PUs activate independently of each other according to the same Markov process discussed in Section 6.3.3, with the same value of π_1 and ℓ_1 for all PUs. This assumption helps to make our analysis and discussion simpler; we note however that it would be straightforward to analyze the case in which π_1 and ℓ_1 vary with the PU. Thus, the number of active PUs active at a given moment will follow a binomial distribution with parameters

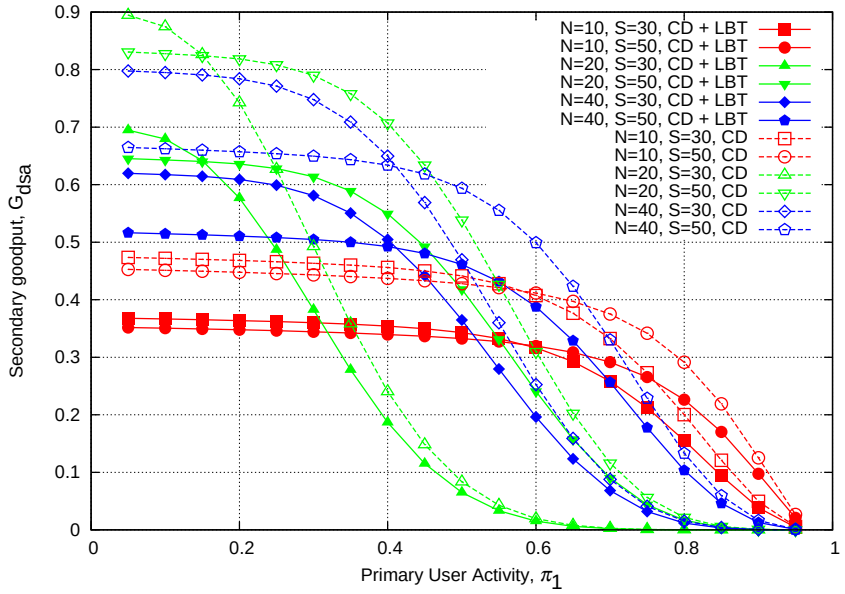


Figure 6.11. *Goodput for secondary spectrum access.*

C and π_1. Hence, we calculate the mean retrieval probability as:

$$E\left[P_{retr}\right] = \sum_{N_p=0}^{C} \binom{C}{N_p} \pi_1^{N_p} \pi_0^{C-N_p} P_{retr|N_p} \tag{6.14}$$

and the overall goodput G_{dsa} for secondary spectrum access, i.e., the overall efficiency according to which the available spectrum resources are used, can be calculated as:

$$G_{dsa} = \frac{CT_{all} - T_O}{CT_{all}} E[\eta] P_{reuse}, \tag{6.15}$$

where T_{all} and T_O are the variables defined in Section 6.2.4, $E[\eta]$ is calculated according to Equation 6.2) substituting P_{retr} with $E\left[P_{retr}\right]$ from Equation 6.14, and P_{reuse} is obtained from Equation 6.11. The resulting performance is reported in Figure 6.11 as a function of the PU activity π_1 for $C = 10$, $\gamma = 5$ dB, $\lambda_{CD} = 5$ dB, $\ell_1 = 35$, $T_{all}/T_{ctrl} = 600$, and $\lambda_{LBT} = 3$ dB when also LBT is used. In general, the efficiency of the proposed scheme degrades as π_1 increases. This is mainly due to the fact that, as discussed previously, the dissemination of control information is very difficult when the number of active PUs is high, and this in turn yields a low retrieval probability and a low medium access efficiency. We note that a choice of $S = 50$ yields a system which is in general more efficient thanks to its better P_{retr} performance with respect to $S = 30$ as PU activity increases; obviously, for very low levels of PU activity, this choice performs slightly worse due to additional overhead.

For $N < 2C$ the system is very robust, since dissemination is again more robust, but the

maximum achievable goodput is low since there are not enough CRs to exploit all available channels. The case $N = 2C$ is the most critical for the dissemination, as we discussed in Section 6.2.3 and Section 6.3.1, and therefore the goodput in this case degrades very quickly as the PU activity level increases. When $N > 2C$ the system becomes more robust again, at the expense of a slightly lower performance for very low values of π_1 which is due to the additional amount of control information which needs to be exchanged. As expected, the use of LBT in addition to Cooperative Detection causes a degradation of the goodput performance; this is the price to be paid when the interference to PUs needs to be further reduced in a scenario with fast-varying primary activity.

Finally, we note that the effect of the detection of PUs and the reuse efficiency of unused spectrum on the overall goodput is limited, since detection itself works almost perfectly with the value of S that is practical for the dissemination of control information in a primary/secondary scenario; moreover, this effect does not increase with an increase in π_1, as we already discussed in Section 6.3.3.

7

Dynamic Spectrum Access for Cognitive Radio Ad Hoc Networks

Cognitive Radio Ad Hoc Networks are designed to operate without an infrastructure support and must rely on local coordination to ensure the required Cognitive Radios functionalities. In these networks, the distributed multi hop architecture, dynamic network topology, diversity on quality of service requirements, and time and location varying spectrum availability are some of the key challenges to be faced during network design.

BESIDES its application to single hop CRNs, DSA has two other main applications which are commonly envisioned. The first is Cognitive Wireless Access (CWA), according to which a Cognitive Access Point (CAP) takes care of identifying unused licensed spectrum, and uses it to provide access to CRs. The second application, which is the one we focus on this chapter, is Cognitive Radio Ad Hoc Networks (CRAHNs), i.e., the use of unlicensed spectrum for multi hop communications among the CRs themselves. These networks are particularly fit for peer to peer content distribution, environmental monitoring, safety communications in disaster recovery scenarios, military communications, and many others.

The material presented in this chapter has been published in [90–92].

Designing a system for CRAHNs presents more difficulties than for CWA, for two main reasons. The first is the identification of unused spectrum. In CWA the CAP is by its role connected to the Internet, and therefore it can infer spectrum availability using simple strategies, such as querying the spectrum regulator at its geographic location, or directly negotiating spectrum availability with the PU or with some intermediary spectrum broker [1]. In CRAHNs, instead, the lack of direct communication with the spectrum regulator or the PUs requires CRs to be able to identify unused spectrum by themselves, using PU detection techniques. The second difficulty is the local coordination of CRs for medium access purposes. In CWA, the presence of a CAP and the fact that commonly all CRs communicate directly with it (i.e., the network is single hop with respect to the CAP) makes it straightforward to use centralized MAC solutions, such as TDMA or Orthogonal Frequency Division Multiple Access (OFDMA). On the contrary, CRAHNs are expected to span over multiple hops, and to lack a centralized controller. While several solutions to this problem are known for traditional single channel multi hop ad hoc networks, it is not straightforward to reuse them for CRAHNs. This is because, assuming we deal with cost effective state of the art technology which allows devices to access only a limited portion of the spectrum at a time, medium access is to be performed across several channels. Moreover, the actual channels which can be used for secondary communications might vary with respect to location and time.

7.1 Technical Challenges

The first issue we encounter in CRAHNs is a chicken-egg problem: CRs need to coordinate among themselves to perform spectrum access, but they also need to access the spectrum in order to communicate among themselves and achieve coordination. In the literature this issue is often referred to as the CCC problem. In Chapter 6 we pointed out that for the practical realization of the CCC, some authors [93] propose to statically allocate some spectrum band. This practice presents two major issues: first, it requires static spectrum regulation, which is exactly what DSA aims at avoiding, and second, the chosen control band could easily become the bottleneck; this is especially true in multi hop scenarios, where the need for control information exchange is potentially very high (e.g., not only for medium access, but also for routing purposes). Some other solutions have been proposed which attempt to solve the first issue by dynamically choosing an unused licensed band to perform

CR control communications; however, the control channel bottleneck issue is not addressed by these proposals.

In the previous chapter we showed that the most interesting way to solve the CCC problem for a single hop CRN is actually to overcome the need for dedicating a channel (i.e., a fixed portion of the spectrum) to control communications only. *Multiple rendezvous* approaches [78, 94] eliminated the need for a CCC for medium access purposes. A side effect of these solutions, however, is that the hopping sequences are defined over a static set of channels, and therefore it is not straightforward to adopt this solution in CRAHNs, in which the set of available channels varies with both location and time. Moreover, the above mentioned multiple-rendezvous strategies provide no means for the exchange of broadcast packets. Both traditional ad hoc routing strategies, and more recent ones such as [95] developed explicitly for CRAHNs, require the availability of broadcast communication services in order to be implemented. As a consequence, implementing a routing solution over a legacy multiple-rendezvous MAC scheme is not at all straightforward. Still, multiple-rendezvous is interesting due to the fact that it mostly solves the issue of the control problem. A first step in this direction was taken in [78], where the authors evaluate the effectiveness of performing broadcast communication for routing signaling purposes within a multiple-rendezvous MAC solution by just having CRs rebroadcast the control packets whenever they switch channel. The authors show that this broadcast strategy may not always be effective in reaching all nodes. This difficulty is exacerbated as the number of nodes and/or hops increases. Thus, in order to support effective and reliable dissemination of control signaling information, a more suitable solution is required.

Of course, the ideal solution for CRAHNs needs not only to address the issue of the exchange of control information, but also to effectively enable an efficient usage of the available spectrum resources. In this respect, it is to be noted that the multiple-rendezvous strategies that we discussed earlier were originally proposed as an extension to single channel technologies (most notably IEEE 802.11); in particular, the advantage that was seen in these solutions was that, just by enabling the use of multiple channels, a significant increase in network capacity could be achieved with respect to the single channel case. However, the capacity limit of multi channel networks is still far from being reached by multiple-rendezvous schemes, which are more of a practical solution to the problem and do not take a systematic approach in maximizing the channel utilization efficiency.

One of the aspects which should be taken into account for an efficient usage of the spec-

trum is that in a multi hop network typically only a subset of the CRs are in the interference range of a given CR. This opens up the possibility of a higher spectrum utilization efficiency by means of frequency reuse. Unfortunately, in practice, this requires more complex spectrum allocation strategies, as well as the availability of more information (e.g., knowledge of the location of each CR). Doing this in a distributed fashion is very challenging. Coupled with this problem is the issue of link scheduling and routing: traditional ad hoc network routing strategies are not effective in multi channel networks, due primarily to the fact that a given link cannot be activated at all times because of the requirement that both the sender and the receiver must be on the same channel. Ideally, channel allocation, link scheduling and routing should be jointly performed in order to maximize spectrum utilization efficiency as well as network performance. In this respect, some interesting solutions have been proposed [96], which however have the drawback of requiring a centralized scheduler. Given the nature of CRAHNs, a distributed solution would be needed in order to allow their practical implementation.

So far, we still have not dealt with what is possibly the most peculiar trait of CRAHNs, i.e., the fact that the identification of those parts of the spectrum which are suitable for secondary spectrum access must be performed by the CRs themselves taking into account that this spectrum availability varies significantly with location and time. However, as discussed in [1] for the case of unlicensed access of TV spectrum, the requirement of maintaining secondary interference to PUs below a certain threshold translates into a sensitivity requirement for single CR detection strategies. This sensitivity requirement is so high that it is not cost effective, if not completely impractical, to implement such detectors with current technology.

For CWA, thanks to the fact that the CAP is by its role connected to the Internet, a straightforward solution is to adopt alternative strategies for the identification of reusable spectrum, such as the consultation of a database reporting available spectrum by geographic location, or explicit negotiation with the owner of the spectrum or with an intermediary spectrum broker. However, for CRAHNs the situation is much harder, since Internet connectivity cannot be assumed to be available. A possible solution to overcome the strict sensitivity requirements is to exploit CD techniques which we showed in Chapter 6 to be very effective for single hop CRNs. Two are the main factors that make these techniques also effective in multi hop scenarios. The first is that, thanks to the fact that more sensing data is available, a better sensing performance can be achieved. The second is that the sensitivity

requirement can be softened due to the multi hop nature of CRAHNs. In fact, the stringent requirement on single CR detection sensitivity is motivated by the need to provide a significant margin to overcome the hidden PU problem, but if the detection is based on sensing data gathered by several CRs at different locations, it is more likely that at least some of the CRs will receive a clear signal from the PU, and therefore a softer sensitivity can be allowed.

7.2 NC^4-DSA for Limited Size CR Ad Hoc Networks

As we discussed in Section 7.1, most prior work in this area has partly addressed the issue of realizing an efficient DSA scheme that takes into account all the over mentioned problems; by contrast, our approach aims at solving all of them simultaneously. Intuitively, the spectrum allocation and transmission scheduling is best performed using knowledge about the particular communication needs (e.g., Quality of Service (QoS) requirements) and spectrum availability (e.g., expressed by PU detection information) of all CRs. As usual, we will refer to this knowledge as the *control information*, obtained by collecting the *control packets* generated by all CRs. Each CR gathers the complete control information, and independently determines for the whole network the resource allocation. The key point is that if the same control information is successfully disseminated to all CRs, and if the resource allocation algorithm is deterministic, then each CR will be able to determine the same resource allocation, without any further interaction among CRs. This is the underlying principle of the NC^4-DSA scheme discussed in Chapter 6 for single hop CRNs, and that we discuss here for use in CRAHNs.

To better illustrate how the proposed scheme works in multi hop networks, we consider the scenario depicted in Figure 7.1. Figure 7.2 shows the channel allocation obtained for one allocation period. As can be seen from the figure, CRs communicate with each other in all those channels which are not occupied by PUs transmission, enabling spatial and frequency reuse.

7.2.1 Control information dissemination

In order to provide an efficient dissemination of the control information our scheme has to assure a high retrieval probability with the lowest possible number of slots. In Chapter 6 we showed that in single hop scenarios NC^4 significantly outperformed other legacy schemes. Unfortunately, in multi hop networks, the presence of PUs at some frequencies

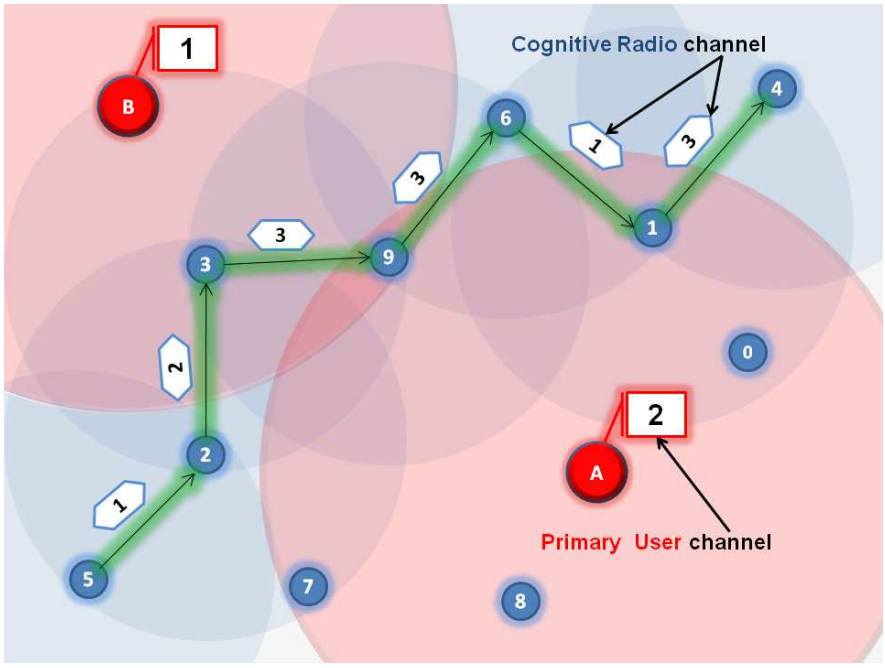


Figure 7.1. *A multi hop scenario with 2 PUs operating in licensed bands.*

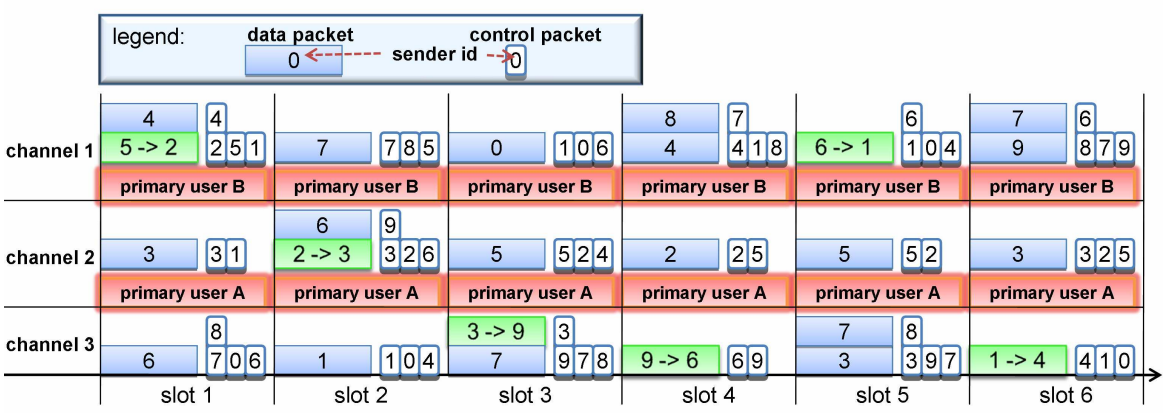
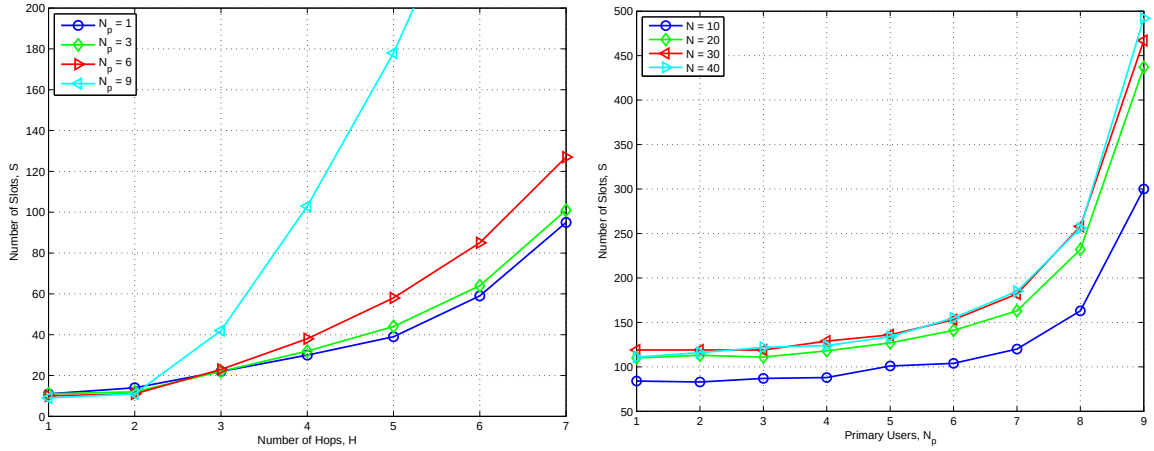


Figure 7.2. *Example of channel switch pattern and transmission schedule.*

and locations affects the effective dissemination of control information, which is crucial for our DSA scheme to work. The reason is that, in order for PU detection to be effective, each CR needs to tune periodically in all channels in order to perform detection attempts. Clearly, only those channels in which it is inferred that no PU is active will be used for data communications. For this reason, the degree of connectivity of the CRAHN decreases as primary activity increases, and this makes dissemination more difficult.

We hereby describe the simulation study we performed using Matlab, with the aim of quantifying these issues and understanding in which conditions NC^4-DSA is practical. In



(a) Number of Channels, $C = 10$

(b) Number of Hops, $H = 7$

Figure 7.3. *Number of slots required for a retrieval probability of* 0.95.

our simulations, CRs and PUs are randomly placed in a 500 m $\times$ 500 m square area. The parameters we vary in our simulations are the number of PUs and CRs, and the number of hops in the network of CRs, which is defined as the minimum number of hops needed for CRs to transmit a packet along the diagonal of the square area. The transmission range of the CRs is adjusted to obtain the desired number of hops; i.e., the transmission range is obtained as $500\sqrt{2}/H$ m, where H is the number of hops. The interference range of the PUs, i.e., the range in which secondary access is not allowed, is set to 1.5 times the transmission range of the CRs. In all simulations, we use a total of $C = 10$ channels, some of which cannot be used by CRs at some locations due to the presence of PUs. We assume that PUs do not change their transmission channel during at least one allocation period. The results presented in this section have been averaged over 500 random topologies per set of parameters.

In Figure 7.3 we report the number of slots required to have $P_{retr} = 0.95$ versus the number of hops, H (Figure 7.3(a)) and the number of PUs, N_p (Figure 7.3(b)), respectively. We observe that the scheme is robust in most cases, although it is to be noted that when the PUs occupy most of the available channels, the number of slots needed to provide a high retrieval probability increases significantly. Despite this, it is still possible to retrieve all the needed information even in cases where almost all channels are occupied given that the number of slots inside an allocation period is large enough.

We also note that our proposed scheme scales well with the number of CRs, achieving successful dissemination with a reasonable number of slots even in the very challenging

scenario where 9 out of 10 channels are occupied at some locations by PUs, and 40 CRs are accessing the remaining bandwidth opportunistically. The behavior of the scheme in scenarios where the PUs use only a fraction (up to 60%) of the available channels is of particular interest, as is expected to be common in real scenarios. In these cases, the proposed scheme provides a high P_{retr} with a significantly lower number of slots (roughly one third) with respect to the worst case.

Finally, it is to be mentioned that we also repeated the same experiments using the dissemination scheme in [78] instead of our network coded control channel solution. Even with an allocation period of 500 slots, this choice resulted in an almost zero P_{retr} in most scenarios, which is of course not adequate for data dissemination.

7.2.2 Goodput performance

In Section 6.2.4 we derived the goodput performance of NC^4-DSA in a single hop scenario taking into account spectrum collisions due to misinformed CRs and the additional overhead introduced by the control information exchange. Following the same approach we can derive the achievable goodput in the case of multi hop networks as well. In this case, in order to calculate the goodput achievable by a given node, it is sufficient to account for spectrum collisions only due to misinformed neighbors. Moreover, the goodput is normalized over its maximum possible value for a given neighborhood, which is equal to the minimum between the total number of channels available, C, and half the number of CRs in the neighborhood. Based on these considerations Equation 6.4 in Section 6.2.4 has been modified accordingly.

The resulting performance is reported in Figure 7.4 for $C = 10$, $N = 40$ and $T_{all}/T_{ctrl} = 6000$, where T_{all} is the duration of the allocation period and T_{ctrl} is the duration of a control packet transmission. In general, the goodput decreases as the number of hops in the network increases. This is because as the number of hops increases the number of slots needed to assure a high P_{retr} increases significantly. This degradation is especially high when the CRs are sharing the spectrum resources with 9 PUs, that is only 10% of the spectrum resources is available for secondary data exchange. Note that, even in this worst case, the scheme is able to assure network connectivity. On the other hand, the performance is very good for a small number of hops, as the dissemination process is always successful, and therefore provides all the required information for correct channel selection. In this case the slight goodput degradation is due to the overhead introduced by the scheme for control in-

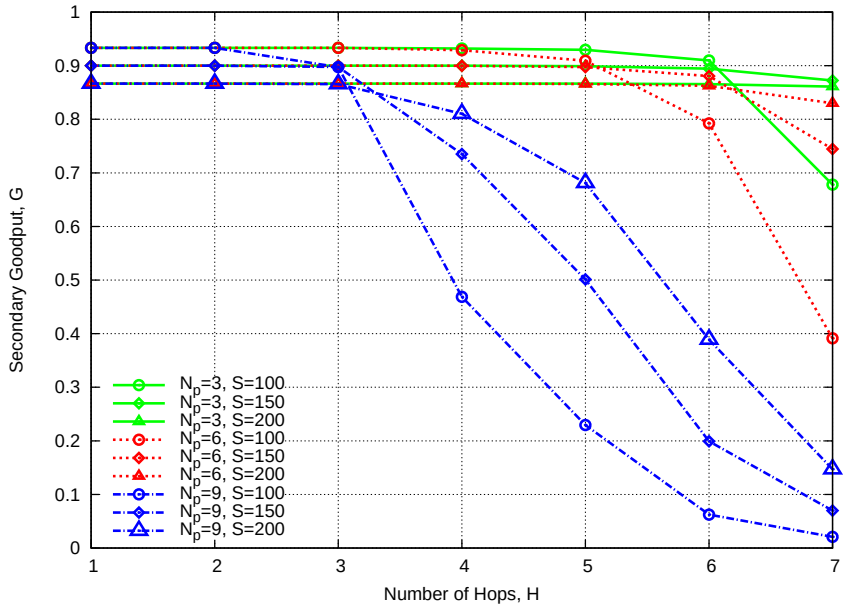


Figure 7.4. *Goodput performance vs. number of hops, H for different number of slots, S and of PUs, N_p ($C = 10$, $N = 40$).*

formation exchange. Note that the increase of the number of slots has a twofold influence: on one side, the fewer slots per given allocation period, the greater the maximum achievable goodput in normal conditions as the control information overhead is reduced; on the other side the scheme suffers in heavy conditions (high number of PUs and number of hops) as the number of slots is not sufficient to effectively disseminate control information during the allocation period.

7.3 Clustered NC^4-DSA for Scalable CR Ad Hoc Networks

So far we have showed that NC^4-DSA is able to address in a joint fashion the problems of the CCC, the multi channel medium access and the efficient channel allocation, and as such it stands as a very promising solution for CRNs. However, our discussion considered only networks having finite size (few hundreds of nodes); this is a significant limitation, since the ultimate DSA architecture is expected to be utilized in very large networks, in order to provide support for coordinated spectrum access by CRs, potentially even belonging to different operator networks.

Hence, in the following we extend our solution to scale up to networks of virtually infinite size which will be called Clustered NC^4-DSA (CNC^4-DSA). The following enhance-

ments are required for the scheme to work in these settings: 1) the definition of a clustering algorithm which enables the creation of multi hop clusters suitable for the operation of the NC^4; 2) the adaptation of NC^4 to work in cluster-based networks where nodes are interested in receiving information from all the nodes of the same cluster while avoiding to interfere with PUs and adjacent clusters communication; 3) the definition of a channel allocation strategy which aims at the efficient repartition of the available spectrum resources to the CRs within a cluster; 4) the evaluation of the performance of the proposed CNC^4-DSA scheme, which is based on the contributions 1), 2), and 3) just described, with respect to the issues of control information dissemination, channel allocation efficiency, spectrum collisions and control information overhead.

We adopt a protocol interference model both for the communications among CRs and for the interactions between PUs and CRs. For CRs, we consider an interference range equal to the communication range, which is for simplicity set equal to the unit length. As for the PUs, they communicate using a Time Division Duplex (TDD) mode, and consequently in any given channel the CRs can detect primary receivers. Let $i \in \{1, \ldots, N\}$ denote a CR. We define:

- $C_i \subseteq \{1, \ldots, C\}$ the set of channels which CR i has detected to be available;
- $\mathcal{N}_i^k$ the set of CRs which are k-hop neighbors of i, i.e., which are at most k hops away from CR i (including node i itself);

7.3.1 Spectrum aware cluster formation protocol

In this section, we discuss practical methods to divide the network into suitable clusters for CR operation. In this respect, several solutions have been proposed in the recent literature. In [97], the authors propose a Bargain Group Formation algorithm which relies on the availability of a CCC. Similarly in [98] the authors propose a cluster based algorithm that divides the network in clusters taking into account the local spectrum availability; as part of this proposal, a neighbor discovery phase is also introduced. These algorithms form clusters by focusing on the constraint that there needs to be at least one channel which is free for all CRs in the same cluster. As a result, these schemes have the tendency to create clusters with a large number of members, but with a small (often equal to 1) number of free channels shared by all CRs.

A different solution [99] aims at creating clusters that provide a good tradeoff between

cluster size and maximum number of shared free channels in the cluster. The original objective in [99] is to make the execution of the cluster formation algorithm less frequent; in fact, if a PU appears in one of the shared free channels, there will still be other channels suitable for the exchange of control information within the cluster. The algorithm in [99] is a maximum bi-clique algorithm which requires nodes to send 3 broadcast messages in order to partition the network into clusters which, as results show, tend to have reduced cluster size. While this algorithm is suitable to create spectrum-aware clusters, its execution time is higher, as nodes have to exchange 3 times more packets than the previous algorithms, making it less reactive to sudden spectrum changes. Moreover, we note that all the above mentioned algorithms partition the network in at most 1-hop clusters. In order to allow NC^4 to work efficiently, we need a clustering algorithm that is able to create clusters that have the highest number of CRs within a k-hop neighborhood, while at the same time guaranteeing that CRs which are members of a cluster share a sufficient number of free channels in order to disseminate control packets and perform data transmissions. The algorithms proposed so far in the literature only provide either a high number of CRs in a cluster [99] or a number of common free channels [97, 98].

In the following, we present a distributed algorithm (Combo) which partitions the CR network into non overlapping clusters based on local spectrum availability; in particular, the proposed algorithm aims at creating clusters of a given size (in terms of number of hops) that takes into account the cardinality of the set of commonly available channels among CRs when making decisions. The algorithm is inspired by [100], where the authors propose a clustering algorithm based on node IDs for the partitioning of the network in clusters. Modified versions of the neighbor discovery algorithms discussed in Chapter 5 can be used to provide to the CRs the list of their k-hop neighbors, along with their corresponding available channels. After neighbor discovery, all CRs run the clustering algorithm independently, and base their decisions on the information stored in the ternary key $\tau_j = \{c_j, d_j, \text{ID}_j\}$, where d_j is the k-degree of connectivity of CRs j, namely the cardinality of its k-hop neighbors set $\mathcal{N}_j^k$, ID_j is the cognitive radio ID, and c_j is defined as follows:

$$c_j = \min_{i \in \mathcal{N}_j^k} |C_j \cap C_i|, \tag{7.1}$$

i.e., c_j is the minimum number of common channels that CR j has with each of its neighbors. Based on this information each CR calculates a weighted priority key ψ_j that will be used during the cluster formation process to decide whether the CR will be a cluster head or join

an existing cluster. A CR j is elected as cluster head if its weighted priority key is the highest among its neighbors, i.e., if the following condition is satisfied:

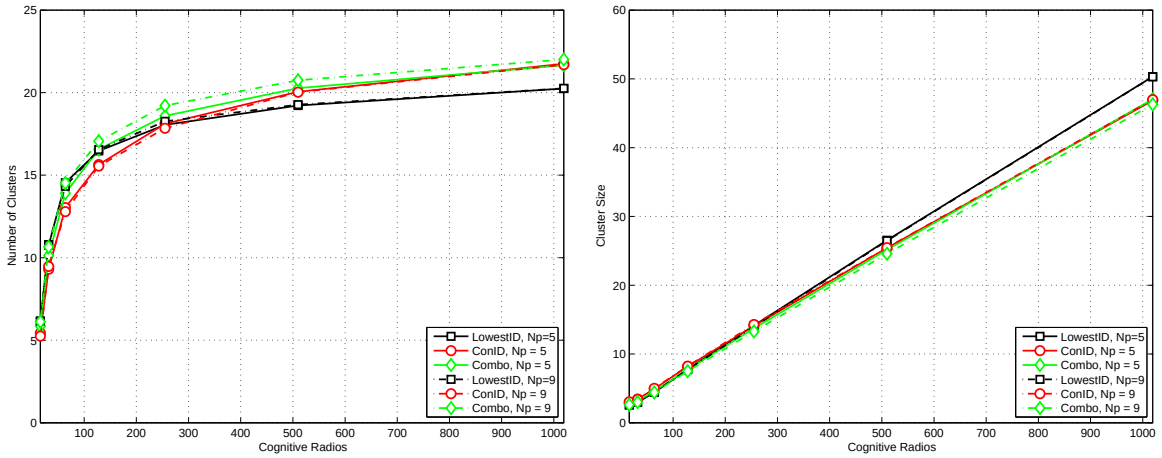
$$\psi_j = \max_{i \in \mathcal{N}_j^k} (\psi_i). \tag{7.2}$$

A cluster head CR initiates the clustering process by sending a cluster formation request, broadcasting its ψ_j to its k-hop neighbors following the same procedure as for the neighbor discovery phase [98, 99]. All nodes whose weighted priority key is the highest among the neighbors request the creation of a cluster with their ID as cluster ID. Nodes that overhear the request join the cluster if their priority is lower, otherwise, in case that they do not hear a broadcast message from any higher priority clusters, they elect themselves as cluster heads. The algorithm terminates once all nodes have made their choices and have been assigned uniquely to a cluster. Note that, even though all nodes only become members of a single cluster, it is possible to have some nodes (border nodes or gateway nodes) which store information about adjacent clusters. This information can be used to implement routing protocols and, most importantly, makes it possible to improve the control information dissemination performance.

In the following we show, by means of simulations, that the algorithm organizes efficiently CRs in clusters based on the variations of their channel availability. To perform our evaluation, we randomly deployed CRs in a 50x50 square area with different numbers of nodes. Furthermore, for each scenario we took into account different numbers of PUs N_p, each of them transmitting over a fixed number of channels which are randomly selected from the set C of all channels. Transmission ranges are set to 1 for CRs and to 1.5 for PUs; the total number of channels is 10. Simulation results are averaged over 100 different randomly deployed topologies.

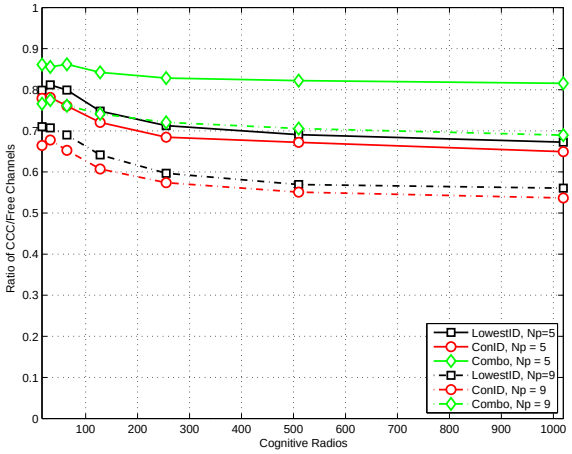
We compare our protocol (Combo) with the following protocols: a) the lowest id algorithm (Lowest ID) [100], and b) the distributed clustering algorithm (ConID) proposed in [101], where the weight is set to the degree of nodes connectivity.

The metrics we focus on are: in Figure 7.5(a) the number of clusters in the network, in Figure 7.5(b) the average cluster size, and in Figure 7.5(c) the ratio of the average number of common free channels in a cluster to the total number of free channels. As we can see all algorithms behave quite similarly in terms of the number of clusters and the cluster size for different numbers of PUs operating in the area. However, the Combo algorithm pro-



(a) Number of Clusters

(b) Cluster Size



(c) Ratio of CCC over Free Channels

Figure 7.5. *Performance metrics vs. number of CRs for different cluster protocols and PUs activity.*

vides a higher ratio of common free channels in all cases, making it possible to considerably improve the dissemination performance, as we will discuss in the next subsection.

7.3.2 NC^4 in clustered CR Ad Hoc Networks

As discussed previously, the cluster formation protocol assigns CRs with similar spectrum availability to the same cluster, which is identified by a cluster ID. Consider two nearby clusters Δ and Γ. Each node $\delta \in \Delta$ reserves memory for a master buffer β_Δ which is used to store all the control packets generated within cluster Δ. When a node has to transmit a control packet, it generates a linear combination over GF(2^1) of the packets in β_Δ, and broadcasts it to all other nodes which happen to be on the same channel. The control packets include in their header the coefficients of the linear combination and the cluster ID. This

ID also identifies the generation set, i.e., the buffer where the packets have to be stored, making it possible not to mix packets from different clusters. This way it is possible for each node $\delta \in \Delta$ to retrieve the control information generated by all the cluster members in sufficient slots.

Now, focus on a border node which, based on the clustering formation algorithm, has decided to be a member of cluster Γ but at the same time can overhear packets sent by all those nodes belonging to Δ that are in its reception range. Basically, due to the fact that the node is situated within the cluster border, it is able to receive control packets from both clusters (the one the node is a member of, Γ, and the adjacent one, Δ), and also to calculate the channel switch pattern and transmission schedule of both clusters. This is done by allocating memory for one more slave buffer where the node stores packets of the adjacent cluster Δ. Overall, the number of slave buffers depends on the number of adjacent clusters from which the border node can receive control packets. This way it is possible for border nodes to avoid transmitting in those slots in which they would collide with the transmissions of other CRs in the adjacent clusters. We note that if the CRs in the border of the adjacent cluster adopt the same strategy, they would also defer from transmission. Hence, the solution that we just proposed is conservative, and its efficiency could be improved by identifying a suitable strategy for selecting which border node will refrain from transmitting; this topic is left for future research.

We continue discussing the dissemination performance of control packets within a cluster, assuming that there are no packet losses due to intra-cluster interference. Once the clusters have been created, we have a disjoint set of clusters with different numbers of nodes interested in mutual exchange of control information. An alternative approach that we consider for performance evaluation purposes is the RMS baseline scheme introduced in Chapter 4.

In Figure 7.6 we focus on the performance of control information dissemination in a 1-hop cluster topology for CRNs with different node densities and number of common free channels. As evident from the figure, the use of network coding reduces considerably the number of slots required to assure a high retrieval probability for the control information. The performance gain with respect to RMS depends on the node density as well as on the number of free CCCs. More precisely, for low values of the node density NC^4 is approximately 3 times faster than RMS to deliver all the control packets with $P_{retr} = 0.97$; the gain becomes more substantial as the node density increases, reaching a 60 times reduction for

$\lambda = 30$. This is due to the capability of network coding to increase the rate of innovative information per packet. Under the RMS scheme CRs pick randomly a packet from their buffer and forward it every time they have to transmit, making it less likely that the packet will be useful to any other CR that happens to be in the same channel. On the other side, using network coding, CRs send linear combinations of their buffer's content providing to their neighbors packets that are most likely going to increase their decoding matrices rank. Hence, when network coding is used, the CRs are able to decode the control information earlier.

It is noteworthy that for NC^4 the number of slots required to assure a given P_{retr} decreases with increasing λ. This decrease is associated to the fact that network coding performance improves with increasing number of CRs per channel, as the degree of connectivity of the CR network is higher. In the case of RMS, the benefit of a higher connectivity degree is overwhelmed by the overall increase of nodes per cluster which reduces significantly the probability of forwarding an innovative packet at the end of the dissemination phase. This is also confirmed by the fact that, for high densities, RMS requires almost the same number of slots to assure $P_{retr} = 0.97$ independently of the number of free common control channels.

This behavior is emphasized when the number of channels used by each PU increases i.e., the number of common control channels decreases. Limiting the number of free channels forces the nodes to access the same channel, leading to a reduced number of slots to assure the same P_{retr} as before. The results suggest that in terms of promptness of the dissemination of control information it is better to have the fewest possible available channels and small clusters sizes. However, this implies that the control dissemination overhead would increase drastically when the number of channels is reduced, leading to very low resources for data transmission. Furthermore, small cluster sizes would lead to increased inter-cluster interference.

7.3.3 Primary user detection

Since the purpose of our scheme is to support the unlicensed reuse by CRs of the spectrum resources which are unused or underused by their licensed holders, we need suitable methods to identify unused channels. Clearly, if all CRs have access to a database providing information on the spectrum availability at each particular location, this problem is solved. Unfortunately, this situation is not expected to be encountered very often in practice: for example, not all CRs might have an Internet connection available to query a centralized

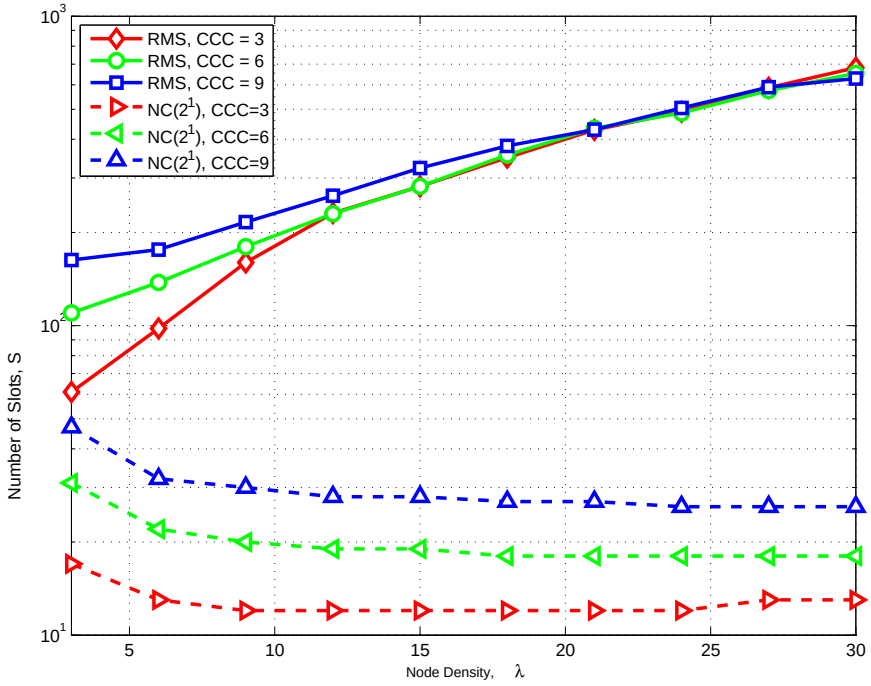


Figure 7.6. *Number of slots required to assure* $P_{retr} = 0.97$ *as a function of the node density,* λ *for different numbers of free CCCs.*

spectrum database managed by the spectrum regulator or spectrum broker. For this reason, we consider the case in which every CR is equipped with an independent PU detection system, such as ED or CFD. We distinguish two different use cases for the determination of the available spectrum resources. The first use case is clustering: as per the algorithm that we described in Section 7.3.1, every CR is required to know the set of unused channels at its own location in order to perform the clustering. We note that, in this context, the consequence of a misdetection is that the formation of clusters will be affected; however, a misdetection does not necessarily cause interference to PUs, since whether the CRs actually use a channel for transmission or not is determined afterwards during the operation of CNC^4-DSA. As a consequence, we argue that for the purpose of clustering it is satisfactory to have each CR determine the set of available channels at its location based solely on its own PU detection data. The second use case for the determination of available spectrum resources is CNC^4-DSA. In this case, the accuracy of the PU detection process is critical, since those channels which are identified as free will be allocated to CRs for data and control communications. We note, however, that the PU detection data gathered by each CR as a result of its own sensing activity can be disseminated to all other CRs within the other control information. As a result, a CD strategy can be adopted to identify with greater ac-

curacy and faster response times those spectrum resources which can be reused by CRs. In Chapter 6 we discussed possible ways of implementing CD on top of NC^4, and showed that this approach is effective in achieving a low probability of interference to PUs while at the same time yielding a good spectrum reuse efficiency for CRs.

7.3.4 Channel allocation

Once the control information has been disseminated among the CRs inside a cluster, transmission opportunities (i.e., slots in available channels) need to be allocated to CRs for data transmission. With respect to this issue, in Chapter 6 we investigated a simple proof-of-concept channel allocation algorithm which aimed at uniform resource sharing among all CRs in a single hop network. However, in multihop scenarios more complex resource algorithms need to be considered, in order to provide means for frequency reuse while at the same time addressing the issue of interference among nodes reusing the same frequency. In the following we describe a practical method for their implementation.

We note that the problem of channel allocation with frequency reuse has been extensively analyzed in the past, first in the context of cellular networks and more recently for multi channel mesh and ad hoc networks. In particular, the vast majority of channel allocation techniques are based on graph coloring techniques; a survey of these techniques can be found in [102]. For CR networks in which the maximum number of channels is not a design parameter but is rather imposed by the scenario constraints (i.e., PU location and activity), we found it more appropriate to consider the variation of the Graph Coloring problem which is known as the Call Control problem [103–105]. According to this formulation, we model the resource allocation problem as an undirected graph $G(V, E)$ where V and E are the sets of vertices and edges, respectively. A vertex $v \in V$ represents a pair of CRs (the transmitter t_v and the receiver r_v) which are requesting a channel to be allocated for data transmission. We have an edge $(v, w) \in E$ between two vertices $v, w \in V$ if t_v and r_w (or t_w and r_v, since the graph is undirected) are neighbors; in words, there is an edge between two transmitter-receiver pairs whenever they cannot be simultaneously allocated the same channel. We have a number of channels C available for the allocation; the problem we need to solve consists in allocating channels to pairs of CRs in such a way that if there exists an edge among them, then they will be assigned to different channels. Note that, unlike the traditional graph coloring problem, it can happen that it is not feasible to assign a channel to every node. A good solution (i.e., a good channel assignment) would maximize the

Algorithm 3 Modified version of the Greedy Frequency Allocation of [103].

for $n = 1 \dots N$ **do**
  $\mathcal{C}_n = \varnothing$ {initial channel assignment for CR n}
$\mathcal{U} \leftarrow \{1, \dots, N\}$ {set of candidate transmitters}
$\mathcal{R} \leftarrow \varnothing$ {set of candidate receivers}
$\mathcal{T} \leftarrow \varnothing$ {set of allocated transmitters}
for $n = 1 \dots \lfloor N/2 \rfloor$ **do** {randomly select $\lfloor N/2 \rfloor$ candidate receivers}
  $r \leftarrow \mathrm{RandomElement}(\mathcal{R})$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$; $\mathcal{U} \leftarrow \mathcal{U} \setminus \{r\}$
while $\mathcal{R} \neq \varnothing$ **do**
  $r \leftarrow \mathrm{RandomElement}(\mathcal{R})$ {consider a new candidate receiver}
  $\mathcal{W} \leftarrow \{u \in \mathcal{U} : u \in \mathcal{N}_r^1\}$ {candidate transmitters in range of this receiver}
  if $\mathcal{W} \neq \varnothing$ **then**
    $t \leftarrow \mathrm{RandomElement}(\mathcal{W})$ {randomly select a candidate transmitter}
    $\mathcal{C}' \leftarrow \{c : c \notin \mathcal{C}_x \forall x \in \mathcal{R} \cap \mathcal{N}_t^1\}$ {allowed channels for the transmitter}
    $\mathcal{C}'' \leftarrow \{c : c \notin \mathcal{C}_x \forall x \in \mathcal{T} \cap \mathcal{N}_r^1\}$ {allowed channels for the receiver}
    $\mathcal{C} \leftarrow \mathcal{C}' \cap \mathcal{C}''$ {allowed channels}
    if $\mathcal{C} \neq \varnothing$ **then**
      $\mathcal{C}_t \leftarrow \{\mathrm{RandomElement}(\mathcal{C})\}$ {allocate channel to transmitter}
      $\mathcal{C}_r \leftarrow \mathcal{C}_t$ {the receiver will tune to the same channel}
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$ {update list of allocated transmitters}
    else {no channel available for this transmitter-receiver pair}
      $\mathcal{R} \leftarrow \mathcal{R} \setminus \{r\}$ {remove from set of candidate receivers}
      $\mathcal{U} \leftarrow \mathcal{U} \cup \{r\}$ {add to set of candidate transmitters}
  else {no suitable transmitter found for this receiver}
    $\mathcal{R} \leftarrow \mathcal{R} \setminus \{r\}$ {remove from set of candidate receivers}
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{r\}$ {add to set of candidate transmitters}

number of colored nodes, i.e., the number of transmitter-receiver pairs which are assigned a transmission opportunity.

Several solutions have been proposed in the literature to solve this type of problem [103–105]. In our case, we argue that a greedy algorithm such as the one proposed in [103, 106] is the best choice. The main reason is that the allocation algorithm must be executed by each node for every allocation period, where the duration of an allocation period ranges from fractions of a second to a few seconds. Therefore, the allocation algorithm must be very fast. To satisfy this requirement, we adopt a modified version of the algorithm in [103], where the modifications aim at introducing randomness in the selection of the nodes which are allocated first. The reason for this is that NC[4] requires the pseudo-random allocation of channels to CRs with the aim of enhancing the dissemination of the control information. The pseudo-code of the resulting algorithm is provided in Algorithm 3.

The obtained average spectrum efficiency $\mathrm{E}[\zeta]$ in number of allocated transmissions per

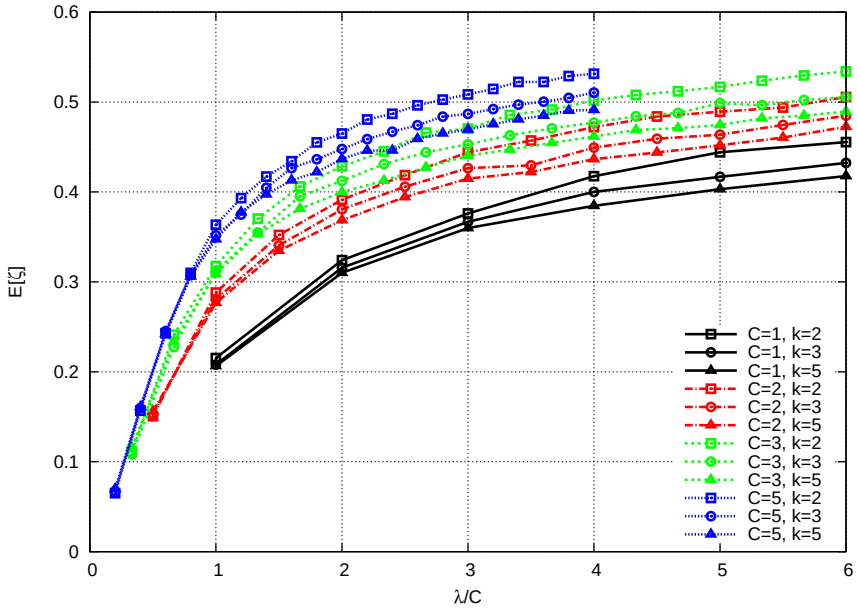


Figure 7.7. *Channel allocation efficiency.*

unit area per channel per slot as a function of the node density λ per unit area is shown in Figure 7.7. We observe that the efficiency reaches its maximum for $\lambda/C \to \infty$, which confirms the observation made in [107] that if the node density per channel is large enough then the maximum efficiency can be achieved even if every node is equipped with a single wireless interface. For a fixed value of λ/C, we note that a higher value of C achieves a higher efficiency; this effect is due to the fact that the allocation algorithm has more degrees of freedoms. Finally, we note that there is a very weak dependency of the efficiency on the cluster radius k.

7.3.5 Impact of spectrum collisions

In this section we derive an analytical model for the spectrum utilization efficiency taking into account the interference coming both from misinformed CRs inside the considered cluster and from CRs in adjacent clusters. Let A denote the area of the cluster. Consider a single node; let B denote the area within its communication range, and furthermore let $I = A \cap B$ and $O = \overline{A} \cap B$, i.e., I and O denote the part of the communication area inside and outside the cluster, respectively. For a generic area X, let $N(X)$ denote the number of CRs in area X. From the theory of Poisson processes [108], we recall that:

$$\Pr N(X) = n = \frac{e^{-\lambda|X|}(\lambda|X|)^n}{n!}. \tag{7.3}$$

We consider values of λ such that the probability that one CR is isolated is negligible. Furthermore, we use the results of the simulations described in Section 7.3.2 and Section 7.3.4 to obtain a characterization of 1) the dissemination performance in terms of P_{retr}, and 2) the probability P_{rx} and P_{tx} that Algorithm 3 allocates respectively a reception and transmission opportunity to a certain CR.

A generic node will be the receiver of a correct data exchange if all the following events are verified simultaneously:

- event R: the node is chosen as a receiver. This happens with probability $P_R = P_{rx}$.
- event D: both the transmitter and the receiver correctly retrieved the control information. This happens with probability $P_D = P_{retr}^2$;
- event M: no misinformed CR in I transmits in the same channel as the considered CR. This happens with probability:

$$P_M = \sum_{n=0}^{\infty} q_I(n) \times \left(1 - \frac{(1 - P_{retr})P_{tx}}{C}\right)^n \quad (7.4)$$

  where:

$$\begin{aligned} q_I(n) &= \Pr N(I) = n + 2 | N(I) \geq 2 \\ &= \frac{\Pr N(I) = n + 2}{1 - \Pr N(I) = 0 - \Pr N(I) = 1}, \end{aligned} \quad (7.5)$$

- event F: no CRs in O transmits in the same channel as the considered CR. Note that it does not matter whether the CRs in O retrieved the control information or not, since they are in another cluster and do not participate in the same allocation as the considered CR. Event F is verified with the following probability:

$$P_F = \sum_{n=0}^{\infty} \Pr N(O) = n \left(1 - \frac{P_{tx}}{C}\right)^n = e^{-\frac{\lambda |X| P_{tx}}{C}}. \quad (7.6)$$

To summarize, the probability P_{rxcde} that a chosen node is the receiver of a correct data exchange is given by:

$$P_{rxcde} = P_R P_D P_M P_F. \quad (7.7)$$

We note that P_{rxcde} depends on the position of the chosen node, as well as the parameters of the scenario being considered (λ, A, B, P_{rx}). We define the random variable X_n which is equal to 1 if CR n is the receiver of a correct data exchange in the considered timeslot, and to 0 otherwise. We define the spectrum utilization efficiency per unit area as:

$$\eta = \frac{\sum_{n=1}^{N(A)} X_n}{|A|\, C}. \quad (7.8)$$

Since η is a random sum [108], and assuming that $\mathrm{E}[X_n] = P_{rxcde}\,\forall n$, we get:

$$\mathrm{E}[\eta] = \frac{\mathrm{E}[N(A)]\ \mathrm{E}[X_n]}{|A|\ C} = \frac{\lambda\ P_{rxcde}}{C}. \tag{7.9}$$

The assumption is reasonable for the channel allocation scheme of Algorithm 3, since the CRs are selected in random order. We note that this is not the case for other schemes, such as the greedy coloring algorithm in [103], in which the order for coloring the nodes in the graph is determined according to their number of neighbors.

7.3.6 Overall goodput

From Equation 7.9 it follows that the expected number of successful transmissions within the cluster in a certain time slot is given by $\lambda|A|P_{rxcde}$. Let T_{all}, T_{slot} and T_{ctrl} be the duration of respectively the allocation period, the time slot and the control packet, as defined in [71]. If the entire duration of a time slot were assigned to data transmission, the expected total time $\mathrm{E}[\tau]$ allocated for successful data transmissions (obtained summing the time spent at different channels and locations in the cluster) would be given by:

$$\mathrm{E}[\tau] = \lambda|A|P_{rxcde}T_{slot}, \tag{7.10}$$

where $T_{slot} = T_{all}/S$. However, the transmissions of control packets by all CRs also need to be accommodated. Since each CR needs to transmit exactly one control packet per slot, the expected total overhead time $\mathrm{E}[O]$ spent in that time slot for the transmission of all control packets in the cluster is given by:

$$\mathrm{E}[O] = \mathrm{E}[N(A)]\,T_{ctrl} = \lambda|A|T_{ctrl}. \tag{7.11}$$

We can therefore express the expected overhead efficiency $\mathrm{E}[\rho]$ of our CNC^4-DSA scheme as:

$$\mathrm{E}[\rho] = \frac{\mathrm{E}[\tau] - \mathrm{E}[O]}{\mathrm{E}[\tau]} = 1 - \frac{ST_{ctrl}}{P_{rxcde}T_{all}}. \tag{7.12}$$

We note that, similarly to what was observed in [71], the effect of the overhead vanishes for $T_{all} \to \infty$, and that $ST_{ctrl} > P_{rxcde}T_{all}$ does not yield a feasible system, since the time to be spent for the transmission of control packets would leave no room for data transmissions.

Finally, the expected system goodput $\mathrm{E}[G]$ per channel per unit area is calculated as:

$$\mathrm{E}[G] = \mathrm{E}[\rho]\,\mathrm{E}[\eta]\,. \tag{7.13}$$

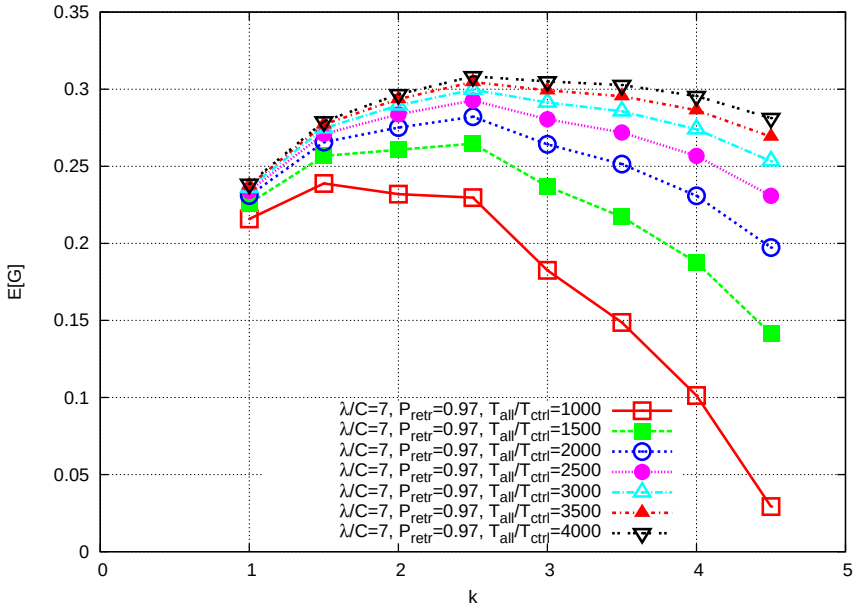


Figure 7.8. *Goodput performance of CNC^4-DSA.*

The resulting goodput performance is reported in Figure 7.8. We note that, for fixed values of λ and C, there exists an optimal value of k which provides maximum goodput. To understand this, we recall that in Section 7.3.5 we showed that the bigger the cluster size the better; on the other hand, in Section 7.3.2 we have seen that a bigger cluster has dissemination problems, i.e., will require a longer dissemination phase to reach a reasonable P_{retr}, which will in turn yield a higher control information overhead. This explains the presence of a maximum in the dependency of the goodput performance on the cluster size.

7.4 Security Considerations

In this section we discuss the security implications of CNC^4-DSA and we compare its vulnerabilities versus two typical CRN architectures: i) centralized and infrastructure based (e.g., IEEE 802.22 [109]); ii) ad hoc and infrastructure-less (e.g., CORVUS [110]). The objective is to understand the advantages and disadvantages offered by CNC^4-DSA from a security perspective. We consider a system with an authority responsible for assigning credentials and identities to all system entities. In particular, we assume that integrity and authentication of the control messages sent by every CR are guaranteed.

With respect to the adversary model, we assume that adversaries have the same communication capabilities of the normal CRs. Moreover, we also assume that they can participate

in the CRN activity as any other CR. CNC^4-DSA scheme expects that every CR in the network obeys the following rules:

- not to intentionally create unacceptable interference to licensed PUs;
- not to prevent other CRs from using available bands, for either selfish or malicious objectives;
- to correctly encode and forward the received control packets, thus contributing to the distribution of signaling and state information;
- to participate in the timely dissemination of correct state information, thus contributing to a sound knowledge base for decisions in the CRN.

In the following we will discuss the following four attacks: 1) Jamming of the CCC; 2) PUE attack; 3) Byzantine modification; 4) Byzantine fabrication which are summarized in Table 7.1. In this table the vulnerabilities of traditional centralized and distributed CRNs are compared to those of a network based on the CNC^4-DSA.

7.4.1 Jamming of the CCC

Traditional DSA schemes rely on the presence of a CCC, which can be either statically allocated or following a predefined channel hopping pattern. In both cases it is straightforward for a malicious CR to prevent control information exchange among CRs by simply jamming the allocated CCC. The impossibility of exchanging control information will prevent CRs from coordinating for data communication in all channels, thus resulting in a Denial of Service (DoS) attack to the CRN. Hence, it is clear that the CCC needs to be secured in some way to avoid this type of attack. As an example the IEEE 802.22 working group has already proposed mechanisms to enhance security in their standard [111].

We note that, the use of CNC^4-DSA avoids the CCC jamming problem, since the control information is disseminated over all available channels, and consequently there is no single point of failure (i.e., the whole system band would need to be jammed for the attack to be successful). For a detailed discussion of this topic refer to the results presented in Chapter 4.

7.4.2 Primary user emulation attack

According to the PUE attack, a CR capable of transmitting a signal with the same spectral characteristics of a legitimate primary signal may prevent other CRs from occupying a

certain band. The objective may be both selfish, (i.e., the CR attacker is interested in occupying the bandwidth itself, so that it performs the attack till the neighbor CRs have switched to a different channel), or malicious (i.e., the CR is interested only in damaging other CRs' communications).

Centralized CRNs, such as IEEE 802.22, require CRs to operate in only one channel at a time, and switch to a different one as soon as they collaboratively detect the presence of a PU operating in that channel. This type of CRN is particularly vulnerable to the PUE attack, since a malicious CR may jump to the frequency channels where the CRN is switching after spectrum handoff, and iteratively perform the attack, thus leading to a DoS for the whole network.

Distributed CRNs, such as CORVUS, are vulnerable to adversary attacks with selfish objectives, but are more robust to the same attack with malicious objectives than a centralized CRN. In fact, in this case the whole network of CRs does not switch to the same frequency channel, after the detection of a PU, but different groups of CRs may decide to switch to different channels according to the result of their spectrum decision making process. As a result, a malicious CR may decide to iteratively perform the emulation attack following the spectrum handoffs of a group of CRs, thus damaging the communications of only a group of CRs in the network. This attack has been extensively studied in the literature [53, 66] and schemes based on the PU emitter locations and on the signature of the primary signal have been proposed.

The CNC^4-DSA scheme is robust to this attack. In particular, with respect to the selfish objective, an adversary CR emulates the primary signal in the frequency channel it is interested in occupying, so as to force other CRs to vacate that channel. However, as soon as the adversary CR starts using the channel for secondary transmissions, interrupting the mimicking of the primary signal, other CRs (randomly hopping over the available channels) will eventually detect this channel as free from primary activity, thus considering it for communications. As a result, this attack is not rewarding for a selfish CR. On the other side, when the attack is performed with a malicious intent, the CNC^4-DSA scheme is even more robust than other approaches, as CRs are spread over all the available channels and spectrally move in order to sense the multiple channels in different instants and distribute sensing information about them using the network coded cognitive control channel. As a result, an effective attack should involve multiple colluded malicious CRs emulating primary signals in all the available channels, which would be significantly more difficult.

7.4.3 Byzantine modification

The CNC^4-DSA scheme is vulnerable to all the attacks which can be inflicted to network coding. A possible attack is the Byzantine modification, according to which a malicious node may intentionally modify the messages in transit. In particular, changes in the coefficients and/or the encoded payload may render the native packets unrecoverable. The proposed protocol is particularly vulnerable to this attack, since network coding relies on mixing the content of multiple data packets, so that a single corrupted packet may jeopardize the entire information flow, at any time. Possible defense towards this attack is the application of end to end error correction or misbehavior detection schemes, as discussed in [112].

On the other hand, CRNs based on both centralized and ad hoc architectures are normally immune to this attack, unless they rely on cooperative and relaying mechanisms. In fact, such a cooperative scheme can suffer from the Byzantine attack since CRs have a chance to act as relays of the primary communications, and instead of forwarding correct information, they might send arbitrary information to the primary destination, thus significantly damaging the physical layer primary system performance [113].

7.4.4 Byzantine fabrication

In general, CRNs rely on collaborative spectrum sensing and primary detection procedures, the CNC^4-DSA scheme is vulnerable to the so called Byzantine fabrication attack, which consists in reporting false feedback about a certain event. For example, due to false information propagated by one or by a group of colluded malicious CRs, CRs may consider as occupied a free frequency band, or viceversa. A frequently proposed solution for this kind of node misbehavior in ad hoc networks is to use reputation systems, where each node is associated with a certain value of trust or reputation derived from a fairly long history of past behaviors. In general, in infrastructureless CRNs, it is infeasible to monitor the neighbors' behavior for a sufficiently long time, since nodes are characterized by high spectral mobility and switch from one frequency channel to another. The security implications of this are very similar to those of a vehicular ad hoc network, characterized by spatial mobility instead of spectral mobility. Consequently, interactions among CRs are commonly quite transient and do not rely on any prior association, so that traditional trust schemes cannot be directly applied. This is not the case for both centralized CRNs, where the CRs can be expected to trust the CRs belonging to the same network, and for the CNC^4-DSA scheme that

we propose, where the CRs that are geographically close meet each other very frequently in the network coded cognitive control channel. As a result, in this context, a malicious CR propagating false information may be easily detected if this information is considered as incongruous by the other CRs. Based on that, reputation schemes can be built to reduce as much as possible the impact of misbehaving CRs.

Attack name	Description	Centralized DSA network	Distributed DSA network	CNC^4-DSA
CCC jamming	Radiation of signals that intentionally disrupt communications in the CCC	Those DSA network relying on a CCC implemented in separated or dynamically chosen bands are vulnerable to this attack.	Those DSA network relying on a CCC implemented in separated or dynamically chosen bands are vulnerable to this attack.	The CCC does not exist in the CNC^4-DSA scheme, and is substituted by a virtual channel, so that the jamming is not feasible.
PUE attack with selfish objectives	An adversary CR transmits signals whose characteristics emulate those of incumbent signals, with the objective of occupying the bandwidth itself.	A DSA network is vulnerable to this attack. Possible countermeasure include signature of primary signal and information about primary location.	A DSA network is vulnerable to this attack. Possible countermeasure include signature of primary signal and information about primary location.	The scheme is vulnerable to this attack, but can recover quickly from it. As soon as the adversary stops mimicking the primary signal, the channel can be occupied again by other CRs. This makes this attack not significantly rewarding for the adversary CR.
PUE attack with malicious objectives	An adversary CR transmits signals whose characteristics emulate those of incumbent signals, with the objective of damaging other CRs' communications.	A DSA network based on e.g. IEEE 802.22 is vulnerable to this attack, since all the CRs in the network switch to the same frequency channel, so that the attack can be iteratively performed in different channels, thus leading to a denial of service. Possible countermeasure include signature of primary signal and information about primary location.	A DSA network based on e.g. CORVUS architecture is vulnerable to this attack, but since after the primary detection, the CRs, organized in groups, are free to switch in the frequency channel they consider more appropriate, a malicious CR may only jeopardize the communication of a group of CRs in the network. Possible countermeasure include signature of primary signal and information about primary location.	Under the condition of not collusion of multiple CRs, the scheme is not particularly vulnerable thanks to the collaborative spectrum sensing procedure realized in different channels and to the spectral mobility of CRs.
Byzantine modification	A malicious node acting as a relay modifies the messages in transit.	A DSA network is not vulnerable to this attack, unless it does not rely on cooperative and relying mechanisms	A DSA network is not vulnerable to this attack, unless it does not rely on cooperative and relying mechanisms	The proposed protocol can be affected by changes in the coded packets in transit, especially by changes in coefficients and/or the encoded payload, which may render the original packets undecodable.
Byzantine fabrication	A malicious node generates messages containing false information	A DSA network relying on mechanisms of collaborative spectrum sensing may be vulnerable to this attack in the primary detection procedure.	A DSA network relying on mechanisms of collaborative spectrum sensing may be vulnerable to this attack in the primary detection procedure.	It is vulnerable because it relies on mechanisms of collaborative spectrum sensing and primary detection.

Table 7.1. *Summary of possible attacks that can be inferred to Cognitive Radio Networks.*

8

Conclusions

In this thesis we presented a novel architecture which enables Dynamic Spectrum Access for Cognitive Radio Networks. We showed that network coding techniques can be used for reliable and robust dissemination of control information among Cognitive Radios. This control information allowed Cognitive Radios to cooperate with each other in a timely manner, guaranteeing the stability of their communications and the integrity of the Primary Users' communications. In order to provide an efficient solution for Dynamic Spectrum Access we first identified essential open problems in the area of Cognitive Radio Networks, and tackled them accounting for the fact that the modules required to implement different functions had to fit in the same architecture.

More specifically, we proposed novel neighbor discovery algorithms which exploit network coding for fast and reliable control packet dissemination. The algorithms provided full neighbor discovery for all CRs in the area and proved to be very robust to various jamming attacks. Considering the system requirements of our scheme in terms of Cognitive Radios' synchronization at the beginning of the allocation periods, we proposed a particular version of the neighbor discovery algorithms that guaranteed Cognitive Radios to synchronously discover their neighbors with high probability. We compared our algorithms with random neighbor discovery schemes representing the current state of the art solutions. Simulation results showed that our proposals bring significant benefits in a variety of wireless environments, ranging from jamming-free to multiple reactive and static jamming scenarios. As a first step, this work identified the benefits of network coding for neighbor discovery in a single hop network setting. However, it also raised important questions which have to be answered in order to provide a suitable solution for those cases where neighbor discovery

has to be implemented in multi hop networks with high node densities which is the particular case of operation of CNC^4-DSA. In such scenarios, network coding can suffer as the decoding matrix size and the global encoding vectors associated to the encoded packets can increase since the amount of information to be mixed is proportional to the number of nodes interested in neighbor discovery. Moreover, the scheme can be significantly improved to efficiently exploit very wide spectrum resources where the utilization of random hopping patterns, while being very robust to jamming attacks, decreases the probability that CRs meet in a given spectrum band for information exchange. Hence, future work in this context is focused on more intelligent channel hopping sequences able to conserve jamming resistance and at the same time improve Cognitive Radio cooperation for a faster neighbor discovery process. Moreover, the implementation in real devices and the development of an extended neighbor discovery algorithm for multi hop cognitive radio networks would give significant insights on the suitability of the proposed scheme for Cognitive Radio Ad Hoc Networks.

We continued presenting a novel Dynamic Spectrum Access scheme based on a Network Coded Cognitive Control Channel. The proposed solution had the remarkable properties of being completely distributed, supporting the exchange of control information with no need for dedicated spectrum resources, and implementing a cooperative Primary User detection strategy. Performance evaluations in single hop Cognitive Radio Networks showed that the proposed solution effectively provides reliable dissemination of control information, efficient spectrum utilization, low interference to Primary Users and satisfactory goodput performance. As a next step we moved towards the study of the feasibility of the same scheme for Cognitive Radio Ad Hoc Networks where we identified the same implementation limitations of network coding in terms of scalability and decoding complexity related to the increase in network size. Hence, to solve these problems we proposed CNC^4-DSA, which aims at providing Dynamic Spectrum Access capabilities to Cognitive Radio Ad Hoc Networks of virtually infinite size. It features a distributed clustering formation protocol and channel allocation based on Graph Coloring for an efficient assignment of the spectrum resources. Our performance evaluation showed that CNC^4-DSA achieved a satisfactory performance even when we consider issues such as control information dissemination reliability, channel allocation efficiency, and both intra- and inter-cluster interference. Furthermore, when compared with state of the art Dynamic Spectrum Access schemes, CNC^4-DSA offered many advantages with respect to security issues. Future research directions include the devel-

opment of more efficient dissemination and channel allocation techniques, as well as the investigation of joint clustering and neighbor discovery strategies specifically designed for CNC^4-DSA as well as the introduction of more refined resource allocation algorithms which account for the time-varying channel conditions perceived by each user.

A

Cross-layer Optimization for Wireless Networks: an Overview

A.1 Introduction

For many years the main philosophy in communications network design has been based on the *layering concept*, e.g., as found in the well-known International Organization for Standardization/Open Systems Interconnection (ISO/OSI) and Transport Control Protocol/Internet Protocol (TCP/IP) reference models. This concept proved vital in the design of many successful wire-line networking solutions with global popularity as it allowed for a clean and modular protocol design, thereby leading to protocol standardization, minimum information exchange between different protocol layers, and thus interoperability among different networks. It relies on the separation among various protocol layers, allowing for the functions implemented at a certain layer to be realized independently of the specific implementation of the other layers. However, in practice the layering principle is rarely adhered to in its integrity, especially in wireless environments. This is because functions realized at different layers interact with each other in a complex way, making the layering model a fictitious (and often inefficient) simplification of the real architecture [115].

A search for answers to the many challenges that arise in wireless networks opened a new horizon on understanding the networking functionalities. A thorough investigation of the relationships among different protocol layers along with the design of mechanisms that

The material presented in this chapter has been published in [114].

break the classical layering structure led to the concept of Cross Layer Optimization (CLO). The general CLO concept consists in coupling the functionalities of different protocol layers by means of a suitable inter-layer communication plane [116]. It leads to more intensive communication among protocol layers (rather than exchanging simple primitives), thus providing a more interconnected protocol stack in general, with the aim to boost the system performance. The potential of CLO design for improving system performance is today regarded as a promising research direction for wireless networks where different and often unpredictable environmental factors, such as noise, interference, fading and mobility, may cause severe performance degradation. As a result, different CLO schemes have been proposed, analyzed and verified via mathematical analysis, simulations and/or experimental testbeds. On the other hand, if not carefully implemented, the CLO approach may reduce the level of modularity and abstraction in the network, resulting in increased network complexity. Therefore, the key to practical CLO lies in the derivation of suitable layer abstractions and the definition of effective coupling mechanisms, designed by adopting a holistic approach.

The CLO system design solutions found in the literature are very versatile. They can rely on using dedicated signaling channels that carry inter-layer information, on merging adjacent layers into a single one or on complete redesign of the protocol architecture. There are also CLO solutions that capture the behavior of a limited number of layers, often adjacent in the classical protocol stack, through mathematical models that are, then, used to devise optimization strategies. Such an approach to CLO brings along the risk of engendering unexpected collateral effects when implementing the cross-layer solutions in a real system. The recognition of this deficiency has fueled a new branch of research aimed at providing fundamental theory for CLO by using rigorous mathematical arguments for defining new concepts in the layering design. One proposal in this direction formulates the cross-layer optimization as an optimization decomposition problem. In this case, the objective of network optimization is mathematically decomposed in several subproblems, each of which is solved by a specific layer of a new layered architecture. The approach is promising, though further research is needed to consolidate it. Nevertheless, the approach followed in the design of the majority of CLO schemes today is mainly heuristic and the literature still lacks a consolidated and widely accepted rigorous framework for CLO.

We aim to systematically describe the CLO paradigm by providing a thorough overview of many relevant CLO aspects. It provides a novel classification framework of the CLO tech-

niques, highlighting the benefits that these techniques bring to the system design. As the CLO approaches are very diverse, we aim at giving a unified analysis framework based on several distinct features of the actual CLO process. Moreover, using the introduced framework, we additionally define the notion of possible CLO architectures realized by superimposing the introduced CLO approaches. Finally, we integrate the CLO system design with an additional emerging research topic, i.e., cognitive networking. Last but not least, we give an overview of the potential pitfalls and disadvantages that may arise when CLO is used improperly for system design.

The organization of this chapter is as follows. Section A.2 provides a novel classification of the fundamental CLO approaches proposed in the literature by using various classification criteria. Section A.3 builds upon the CLO approaches to define architectures that convey the CLO principle today. Section A.4 discusses the application of CLO in different levels of the protocol stack. Section A.5 introduces some relevant mathematical models which are suitable for CLO while Section A.6 considers the relationship of CLO techniques with cognitive networking and elaborates on the idea of CLO system design as being a subset of cognitive networking in general. Section A.7 gives insight into the necessary cautionary viewpoint on CLO system design and the possible hampering of the actual adoption of CLO techniques in real devices. Finally, Section A.8 gives future research directions on CLO and concludes the treatment of CLO.

A.2 Classification Criteria

As mentioned, the popularity of wireless networks has fostered the proliferation of many different and heterogeneous CLO schemes. In order to facilitate the analysis and comparison of the different solutions, we need to define some classification criteria. After a thorough analysis of the literature, we identified the following four criteria:

- *Approach* – based on the degree of changes brought to the classical layered architecture;
- *Scope* – referring to the number of layers interested by the CLO scheme;
- *Target* – addressing the object of the optimization;
- *Implementation* – concerning the way in which CLO techniques are implemented.

A very compact summary of the CLO classification based in these criteria is reported in Table A.1.

Table A.1. *Taxonomy of CLO approaches*

FEATURE	CLASSIFICATION
Approach	Evolutionary vs. Revolutionary
Scope	Targeted vs. Joint
Target	User-Centric vs. Network-Centric
Implementation	Centralized vs. Distributed

A.2.1 Approach: evolutionary vs. revolutionary

A first differentiation among different CLO techniques can be performed based on the degree of changes that the protocol stack has to undergo [117, 118]. According to this criterion, CLO schemes can be divided into two broad categories: *evolutionary* and *revolutionary*.

The evolutionary schemes concentrate more on compatibility rather than on performance. According to this approach, the cross layering paradigm is implemented by permitting a vertical communication through the protocol layers, keeping the existing protocol stack in the background. Usually, evolutionary CLO can be pursued via basic evolutionary solutions or system-wide solutions. The former are simple, yet effective extensions of the strict layering structure, while the latter provide stack-wide layer interdependencies to optimize the overall network performance. On the other hand, the revolutionary CLO techniques (also known as alternative CLO design) are free from any existing layered concept. They target new and more generic definitions of functional entities and their mutual interactions. Revolutionary CLO techniques aim at providing maximum performance with no requirements of backward- or inter-compatibility. As a result, these approaches are often used in isolated environments.

A.2.2 Scope: targeted vs. joint

The CLO approaches can be divided into *targeted* and *joint* [119], depending on the number of protocol layers involved in this optimization. Targeted CLO techniques aim at performance improvement on a particular layer or subset of layers via maximization of the objective functions at the layer(s) in focus. Therefore, even when all the layers are involved in the CLO design, the major benefit is experienced only by a restricted part of the protocol

stack. Conversely, joint CLO techniques allow all protocol layers to experience some (albeit possibly limited) performance improvement. They require that a cumulative multi objective function is maximized, constraining the objective functions at each layer. The joint CLO techniques play a key role in the description-based optimization of layered communication systems [120], where an optimum setup for every layer is derived based on the description of every layer's capabilities by a finite set of feasible operation points.

A.2.3 Target: user-centric vs. network-centric

Another possible classification of CLO schemes is based on whether the optimization effort is focused on the *user* or on the *network* performance [121]. User-centric optimization is more common since the scarcity of transmission resources, which fuels the need for optimization, is generally more severe at the user rather than on the network side. It is interesting to note that the user-centric schemes, in general, aim at adapting the user terminal behavior to the actual network conditions, in order to provide consistent QoS guarantees, whereas the network-centric schemes focus on Network Utilization Maximization (NUM) [122].

A.2.4 Implementation: centralized vs. distributed

Finally, a substantial distinction can be made between *centralized* and *distributed* CLO approaches [121]. Centralized approaches use a central cross-layering optimizer that interacts with all protocol layers and provides a common solution based on appropriate layer abstractions. This approach, however, requires the development of specific interfaces between the central optimizer and all the layers of the protocol stack. Furthermore, the central optimizer needs to know all the protocols and algorithms it interacts with at each layer. On the other hand, the distributed CLO approaches make use of the cross-layer information to optimize each single layer in a distributed fashion. Hence, each layer is capable of independent reconfigurability that, however, depends on mechanisms both within a single layer and across different layers.

A.3 Cross-Layer Architectures Taxonomy

A CLO architecture should make it possible to collect, share and set all or a part of the parameters that, in traditional system architectures, are generally confined within the bor-

ders of their respective layers. As a result, there are several challenges in the design of a CLO architecture, with modularity, affordable complexity and scalability of the architecture being of primary importance. The cross-layer architectures proposed in the literature are mainly based on three possible approaches [123]:

1. *Merging* - where adjacent layers are merged together in order to accomplish a specific goal;

2. *Streaming* - where even non-adjacent layers may exchange information using dedicated interfaces;

3. *Parallel* - where the cross-layer interaction is realized through a parallel structure that acts as a shared database of the system state.

Having in mind the classification criteria described in the previous section, in the following we will provide a taxonomy of the CLO system architectures according to the above listed approaches.

A.3.1 Merging cross-layer architecture

The merging CLO alternative is the oldest and simplest. It usually comprises user-centric, evolutionary and targeted CLO approaches. The most representative example of a merging architecture consists in bringing together the physical and the data link layer in order to provide efficient link adaptation. The advantages of the merging CLO architectures lie in the low complexity, ease of implementation and limited violation of the layered protocol stack. The main disadvantage is the limited performance improvement that can be obtained with this approach as the optimization only involves few parameters.

A.3.2 Streaming cross-layer architecture

The streaming approach usually requires that a new interface is created at each layer involved in the cross-layer design, beyond those existing between adjacent layers. As a result, cross-layer schemes based on the streaming architecture are sometimes referred to as *direct cross-layer communications*. A typical example of streaming architecture is shown in Figure A.1, taken from [118]. The focus is on the specific interactions among layers and on the required signaling messages. In particular, each layer interested by the CLO scheme needs to exchange control information only with a subset of the other layers by means of an

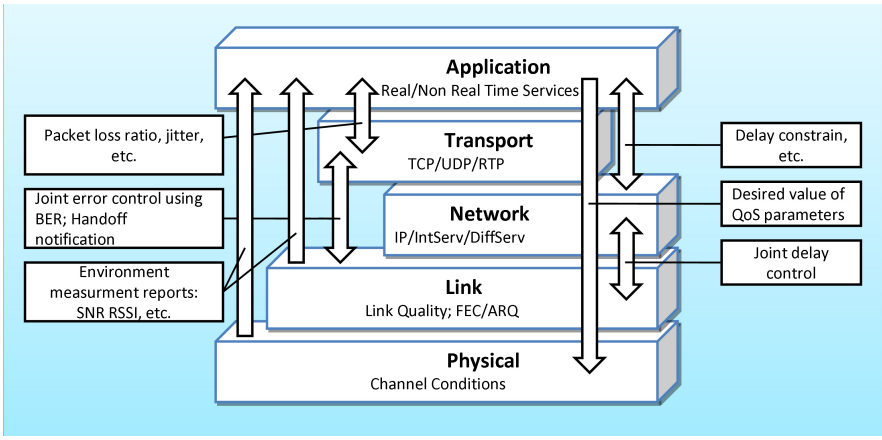


Figure A.1. *Streaming cross-layer architecture [118].*

appropriate control function. With reference to the CLO approaches discussed in the previous section, the architecture in Figure A.1 can be classified as evolutionary and targeted, since the interaction between the application and the physical layer can be used to adjust the users demand according to the PHY performance and vice-versa. The advantages of the streaming cross-layer architecture are in the affordable complexity and the freedom to optimize various parameters using information from non-adjacent layers, whereas the main disadvantage is the loss of system modularity with the potential risk of ending up with a *spaghetti*-like stack implementation.

A.3.3 Parallel cross-layer architecture

The parallel cross-layer architecture introduces a shared entity among all the protocol layers, thus avoiding direct inter-layer communication. The shared entity is a database that can be accessed by every layer that needs to utilize it. This type of architecture is also referred to as *indirect CLO communication*. System architectures based on this approach are increasingly popular since it appears suitable in addressing many of the CLO challenges. Two examples of cross-layer architecture based on the parallel approach are depicted in Figure A.2 and Figure A.3. Figure A.2 proposes an architecture where all layers communicate with a single control plane that performs CLO for all layers in a unified way, according to some optimization criteria. In this case the control plane, which becomes the core of the network node, can actually be used to create a new abstraction of the network functionalities and thus the layered structure loses most of the original meaning. The CLO approaches that follow this design principle can be classified as evolutionary and joint. A less general

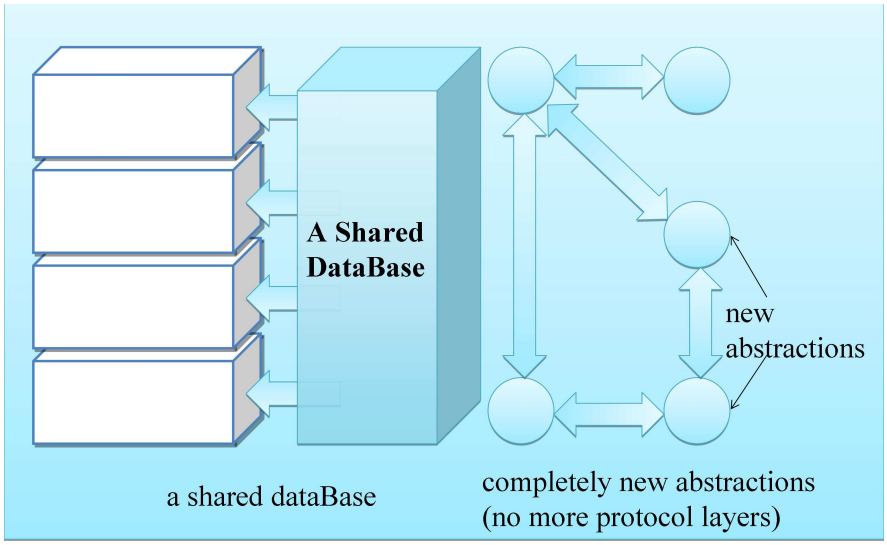


Figure A.2. *Parallel cross-layer architecture.*

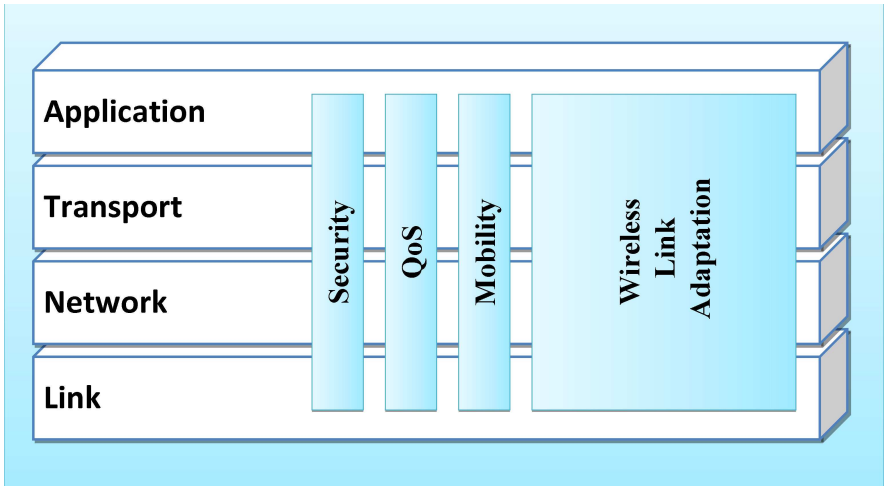


Figure A.3. *Parallel cross-layer architecture [124].*

approach is shown in Figure A.3 [124], where the classic protocol stack is augmented by superimposing transversal control planes, each in charge of a different function and capable of interacting with all layers in order to achieve its targeted optimization goal. Hence each plane acts as both a communication and a control plane. The main advantage of this CLO system architecture is clearly its powerful optimization space, and is paid in terms of implementation complexity.

The increased variety and popularity of wireless access networks give rise to new problems, such as interoperability and seamless roaming in a heterogeneous environment. These

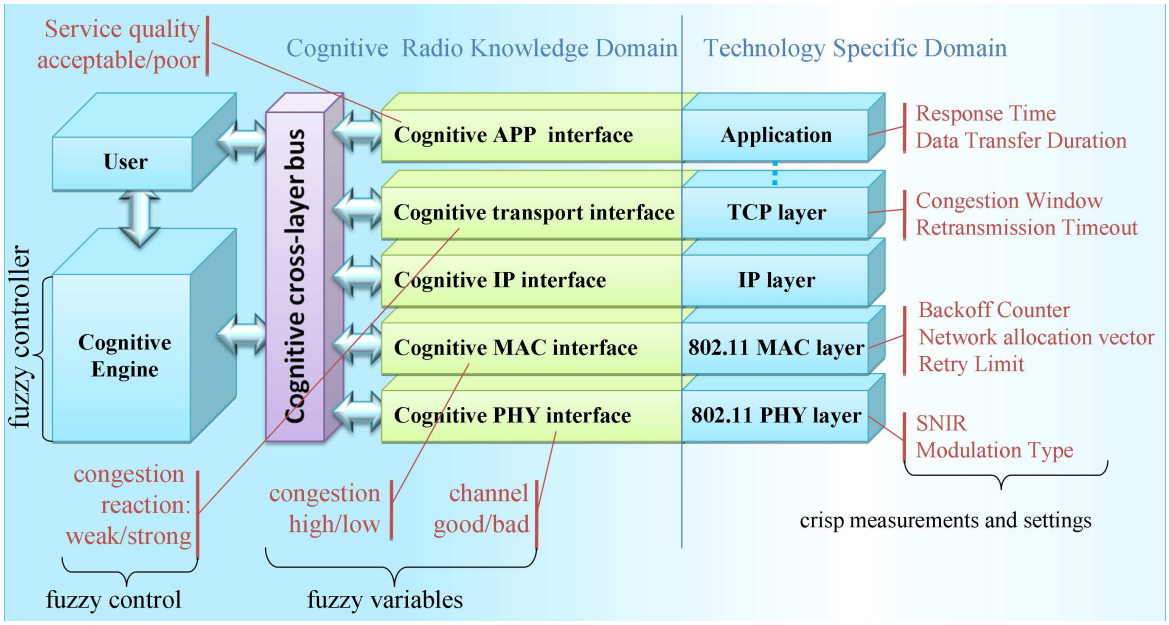


Figure A.4. *Example of fuzzy cognitive cross-layer architecture [125].*

aspects are tackled by a third type of parallel CLO architecture that abstracts the specificity of heterogeneous systems by referring to a fuzzy representation of the requirements and performance measurements. In other words, the cross-layer Control Plane considers the use of fuzzy logic as an effective means for realizing both knowledge representation and control implementation. An example of fuzzy cross-layer architecture is given in Figure A.4, taken from [125], where the authors propose the use of fuzzy logic for cross-layer interaction by claiming that only an abstract and technology independent representation and management of the information belonging to the different layers will permit to meet the modularity and scalability constraints that are posed by the cross-layer concept. The use of fuzzy logic, in fact, permits to limit locally the implementation complexity of a given layer, while assuring the exchange of highly significant pieces of information between all layers. In this way, the risk that the design of the cross-layer engine becomes impractical due to overwhelming complexity is reduced. Clearly, the counterpart of this approach is that it cannot give optimality guarantees.

A.4 Classical CLO schemes for wireless networks

In this section we discuss some of the classical CLO schemes that have been proposed for wireless networks. An extensive overview of the entire literature devoted to this topic

would turn out to be an overwhelming task, due to the impressive amount of work carried out in this area in the last few years. Rather, we focus our attention to a limited number of instances that can be considered as representative of the most significant approaches to the problem. As mentioned, most of the classical CLO schemes address specific problems rather than providing a general system optimization. According to this observation, the survey will be organized following a top-down approach, in which we first present the cross-layer schemes aimed at improving the application performance, subsequently we discuss the solutions devoted to the transport layer optimization and finally we analyze the schemes addressing network layer enhancements. The CLO between MAC/PHY layer is not considered as the literature is very vast and because the aim of most of these approaches is to fuse both layers into a radio link layer which is used as the reference layer for CLO of the upper protocol layers.

A.4.1 Application layer

The classical protocol stack inherited from the wireline world does not work efficiently in wireless networks. As a consequence, several applications with strict QoS requirements experience unsatisfactory performance when delivered over wireless links. Hence, one of the main goals of CLO is to alleviate the QoS degradation perceived by end-users in wireless networks. In [126] the authors focus on maximizing the end-to-end quality of the video streaming between a base station and several subscribers. The main idea here is to extract different parameters from the application and the MAC/PHY layer, such as video encoding distortion, modulation, code rate, and optimize them based on different objective functions that take into account specific QoS restrictions. Simulations results show that the CLO scheme brings some improvements with respect to classical layered architectures, however the implementation complexity of such a centralized scheme is not discussed. Moreover, the size of the optimization space tends to explode when the number of subscribers communicating with the base station increases, thus limiting the practical applicability of the scheme.

Inspired by the previous work, Khan et. al [119] generalize the CLO approach by providing a more systematic description of the CLO architecture and abstraction of the parameters. The authors of [119] also provide an analysis of the computational costs of the CLO scheme, discussing the limits of the scheme while increasing the number of subscribers.

A heuristic CLO for real-time applications is discussed in [127] where the scheme is

Table A.2. *Comparison of CLO algorithms for QoS improvements.*

Protocol	Classification	Architecture	Cross-Layer Information	Benefits
CLO Video Streaming [126]	Evolutionary, Targeted, User-centric	Streaming	Data rate, PER, packet size, channel coherence time, encoding distortion	Maximize the minimum PSNR in the system
Video Streaming [119]	Evolutionary, Targeted, User-centric	Parallel	Data rate, PER, rate distortion profile	Maximize the users perceived video quality
CLO Multimedia Traffic [127]	Evolutionary, Targeted, User-centric	Streaming	packet loss rate, mean delay,	Better QoS and improved system's throughput
APOS [128]	Evolutionary, Targeted, User-centric	Parallel	voice codec, playout buffer delay, SNR, collision rate, average channel busy/idle time	Maximize perceived voice quality

somehow less centralized as the base station shares part of the optimization calculations with the subscribers. The base station collects information regarding the traffic generated, the channel conditions and the QoS parameters of the active connections, and suggests proper adjustments of the modulation and/or traffic rate at the subscriber's side in order to optimize the overall systems throughput and QoS. A last example is presented in [128], where the authors propose a cross-layer architecture which aims at improving the performance of Voice over IP (VoIP) applications over IEEE 802.11g. This is done by evaluating the medium status in terms of SNR, collision probability and average channel busy time observed by the wireless device. Based on these parameters the CLO architecture jointly optimizes the transmission rate and the retry limit with the objective of maximizing the voice quality perceived by the users.

Table A.2 summarizes the main features of the CLO schemes here considered.

A.4.2 Transport layer

The congestion problems in wireless networks, especially in multi hop mobile ad hoc networks, have been tackled in many ways. An extensive overview of the existing proposals, and of their key ideas and interrelations, is given in [129].

Several cross-layer schemes focus on the interaction between TCP and MAC, with the aim of alleviating the serious problems generated by medium access contention [130–132]. In [130] the authors present a rate based Wireless Congestion Control Protocol (WCCP), able to exploit cross-layer interactions between traditional TCP and MAC layers in order to re-

duce the performance degradation and unfairness of the TCP protocol in wireless networks. WCCP relies on an easily measured metric, namely the *channel busyness ratio*, in order to calculate the ideal sending rate. The channel busyness ratio is defined as the ratio between the time intervals during which the channel is busy and the total time. This metric provides a reliable indicator of network congestion and bandwidth utilization, hence adequate feed-back information to the TCP control mechanism in order to avoid network overload. Results show that WCCP provides significant improvements in terms of channel utilization, end-to-end delay and fairness, compared with traditional TCP. However, WCCP does not address mobility issues.

Another cross-layer scheme that involves interactions between the transport and MAC layers is the TCP Contention Control protocol proposed in [131]. The idea behind this protocol is to track at each Round-Trip-Time the value of several MAC parameters, such as available bandwidth, throughput and packet contention delay. This information is then used to estimate the amount of traffic that can be sent to get a good tradeoff between throughput and contention delay. Results show that the integration of this protocol with the standard TCP protocol can drastically improve the throughput and end-to-end delay of TCP in different topologies and flow patterns while being backward compatible with classical TCP.

A different approach to cross-layer TCP/MAC optimization has been proposed in [132], where the resource allocation in the network is formulated as a utility maximization problem using contention graph and contention matrix with constraints that arise from channel access contention. The system problem is solved by using two algorithms that are spatially distributed and decomposed into two protocol layers. The first algorithm runs at the MAC layer, where it generates congestion (contention) prices based on local aggregate source traffic. TCP sources adjust their rates based on the aggregate prices in their paths. The second algorithm schedules link layer flows according to the congestion prices of the links. Such an approach provides a systematic method to jointly design TCP congestion control and MAC algorithms, though it is largely based on ideal assumptions that make its practical deployment rather complex.

Other forms of TCP/MAC cross-layer optimizations address the reduction of multi-layer ACK redundancy. For example, [133] proposes the deployment of an ACK agent lying on top of the MAC that locally generates TCP ACKs for the transport protocol as soon as the link layer confirms successful packet delivery, thus avoiding transmission of TCP ACK packets over the wireless channel. In addition, [133] also proposes Cross-layer Congestion

Control for TCP (C^3TCP), which tries to optimize the TCP output flow rate based on the bandwidth-delay product obtained from link layer measurements. The method requires the introduction of an additional module within the protocol stack of the mobile node, able to adjust the outgoing data stream based on capacity measurements. The author implemented C^3TCP for chain and grid topologies in multi-hop ad hoc wireless networks and the results show a throughput improvement of around 27%, 18% and 7% against standard TCP, TCP Westwood and TCP Vegas, respectively, for the chain topology, whereas smaller throughput fluctuations were observed for the grid topology. An approach that achieves congestion control without requiring dedicated congestion-related signaling is introduced in [134]. The solution is named Cooperative Cross-layer Congestion Control (CXCC) and is based on implicit hop-by-hop congestion control. CXCC provides a "semi-reliable" packet transport, where packets may only be lost in case of failing nodes or links, but not due to queue overflow. CXCC uses an Request For Acknowledgment (RFA) mechanism for eliminating the unwanted acknowledgments, resulting in high and constant throughput in various scenarios. The author in [134] also proposes the implicit Backpressure Multicast Congestion Control (BMCC) protocol that provides hop-by-hop congestion control.

Besides the widely used cross-layer approaches between transport and MAC layers, there are also other forms of cross-layer interactions that involve the transport layer. In [135] the authors introduce a joint TCP and physical layer congestion control scheme by using mathematical operations such as Lagrange multiplier, gradient and steepest descent method. The PHY layer adapts the transmission power according to the channel conditions, the level of interference, and the congestion in the network, whereas the TCP layer controls congestion using Reno-2 window based flow control. As a result, the cross-layer congestion control technique provides performance improvements in terms of throughput and window size variations, but only in good channel conditions. In bad channel conditions, the algorithm does not converge.

Table A.3 provides a comparison of the previously elaborated cross-layer aided congestion control algorithms.

A.4.3 Network layer

The design of routing protocols for wireless networks poses serious challenges due to the dynamic and ever changing nature of the wireless medium that imposes restrictions to the application of the traditional routing protocols used in wireless networks. The solu-

Table A.3. *Comparison of cross-layer aided congestion control algorithms.*

Protocol	Classification	Architecture	Cross-Layer Information	Benefits
Wireless Congestion Control Protocol [130]	Evolutionary, Targeted, Network-centric	Streaming	Channel Busyness Ratio	Improves fairness; Improves throughput; Decreases end-to-end delay
TCP CCP [131]	Evolutionary, Targeted, User-centric	Streaming	Available bandwidth, Throughput, Packet contention delay	Decreases application response time; Decreases average link layer attempts; Increases throughput
LLE-TCP [132]	Evolutionary, Targeted, Network-centric	Streaming	Link-layer feedback	Throughput improvement; Fairness optimization
C3TCP [133]	Evolutionary, Targeted, User-centric	Streaming	Bandwidth and delay measurements	Throughput improvement; Smaller throughput fluctuations
CXCC [134]	Evolutionary, Targeted, User-centric	Streaming	Hop-by-hop implicit feedback	High and constant throughput
PHY-Transport [135]	Evolutionary, Targeted, User-centric	Streaming	SINR, Price derived from capacity, Average traffic	Stabilized and better throughput in good conditions

tions proposed for routing in wireless networks may vary from simple modifications of the classical protocols to complex multilevel hierarchical schemes [118]. Furthermore, the introduction of cross-layer system design may lead to improved throughput and scalability in dynamic wireless networks. A good survey on the usage of multi-rate, multi channel technologies in order to provide cross-layer based routing in mobile ad hoc networks is given in [136].

The implementation of a new metric based on PHY/MAC feedback information to the network layer can lead to significantly higher network throughput and lower network congestion. In [137] the authors propose PARMA (PHY/MAC Aware Routing Metric for ad hoc wireless networks with multi-rate radios), a routing metric for proactive ad hoc routing protocols that helps spreading the network traffic across good links in the network, thus increasing network capacity and reducing packet loss and delay. To this end, PARMA makes use of the SNR and the channel access delay measures exported by the MAC and PHY layers. The PARMA metric increases the system throughput by a factor of 2.5 with respect to the conventional routing metrics in chain topologies.

In [138] the authors present some experimental results for a link aware, cross-layer routing scheme. The authors elaborate on the unidirectional link detection scenario by using the Early Unidirectionality Detection and Avoidance (EUDA) technique via transmitted power and received signal strength. This technique is used in conjunction with a channel load-based routing which takes into account the Measured Transmission Time (MTT) and multi-path routing. The experimental results show that the EUDA technique is not 100% accurate, whereas the MTT based metric performs better.

Ref. [139] explores the usage of three basic PHY/MAC layer parameters, namely interference, packet success rate and data rate, for the derivation of a cost function that allows the network level to find paths with low levels of generated interference, high transmission rates and high reliability in terms of packet success rate. Under these constraints, the routing problem becomes NP-Complete. The solution is purely theoretical and is applicable for wireless mesh networks. Another example of a PHY/network interaction that is implemented on a testbed is given in [140], where the authors propose a rate-aware routing protocol that uses the physical transmission rate as a metric. The rate-aware metric is used to create a cost function for the price of every link, with the aim to maximize the transmission rates of the links. The experimental results show low packet loss ratio and high throughput.

Ref. [141] proposes Expected Throughput (ETP), a new and improved routing metric for wireless mesh networks. ETP takes into consideration the bandwidth sharing mechanism of 802.11 DCF and yields more accurate throughput estimates than existing routing metrics. ETP is calculated using the packet success probabilities of a certain link in the forward and reverse directions and the expected bandwidth experienced by that particular link. The routing policy is to choose the path with the highest routing metric. Another example of a cross-layer interaction between the MAC and the network layer is the Directional Routing Protocol (DRP) [142], an on-demand routing protocol that uses directional antennas to improve packet delivery and includes an efficient route discovery mechanism, establishment and maintenance of directional routing and directional neighbor tables, and novel directional route recovery mechanisms. The results show that DRP exhibits low latency, low overhead and fast route repair.

The cross-layer optimization between the network and the MAC layer has also been shown to improve the energy efficiency of wireless sensor networks. References [143, 144] present MAC and routing based optimization protocols that minimize the signaling overhead through stateless routing decisions made at the receiver rather than at the sender,

which permits some energy saving. Another routing/MAC cross-layer interaction is given in [145], where a region-based opportunistic routing is proposed. This cross-layer routing solution utilizes the spatial diversity due to the high node density to reduce the latency and the average power consumption while increasing robustness to adverse channel conditions and node failures.

A general modeling and solution framework for the throughput optimization problem in multi-hop wireless networks that jointly considers routing, medium contention and network coding is given in [13]. First, the authors give a high-level formulation of the optimization problem, which involves variables from both the network layer and the physical layer. Then, by using Lagrange relaxation and sub gradient optimization, they decompose the overall optimization problem into a sequence of smaller sub problems that deal with data or power allocation, each involving variables from either the network layer or the physical layer. The elaborated framework can handle the throughput optimization problem in an efficient and distributed fashion for a broad range of wireless network scenarios.

Table A.4 summarizes the major aspects in the elaborated cross-layer aided routing protocols. Besides the cross-layer interactions between the network layer and the lower layer(s) for yielding routing aids, there are solutions that define cross-layer metrics based on the information exchanges between the network and the application layer. Because of the dynamic topology of the wireless networks and the applications dynamic resource requests, static routing often leads to poor results. Therefore, the implementation of an application-aware communication system that builds routes based on applications needs is essential. Ref. [146] uses an intuitive generalization to source routing which facilitates discovery of a resource in a mobile ad hoc network and the corresponding creation and maintenance of the required route.

A.5 Mathematical Models

All the above described approaches and considerations on cross-layering are based on pre-existing layered structures and make use of intuitive, heuristic or trial-and-error considerations which need to be designed and verified case-by-case.

Clearly, the availability of a complete mathematical model for the network behavior would make it possible to address the performance optimization problem in a more systematic way. This approach has been recently taken by some authors that have proposed

Table A.4. *Comparison of cross-layer aided routing protocols.*

Protocol	Classification	Architecture	Cross-Layer Information	Benefits
Cross-layer enhanced DSDV [137]	Evolutionary, Targeted, Network-centric	Streaming	SNR, channel access delay	Throughput increase; Reduced network congestion
Link-Aware Routing Protocol [138]	Evolutionary, Targeted, Network-centric	Streaming	EUDA, MTT, Bandwidth, Delay	Route discovery overhead reduced; Performance improvement; Multipath gains
PHY-aware routing [139]	Evolutionary, Targeted, Network-centric	Streaming	Interference, Packet success rate, Data rate	Low interference; Higher available transmission rate; Higher reliability
MeshDV [140]	Evolutionary, Targeted, User-centric	Streaming	Throughput, Delay	Packet loss ratio decreased; Throughput increase in multihop scenarios
MAC-Aware Routing [141]	Evolutionary, Targeted, Network-centric	Streaming	Packet success probability, expected bandwidth	Better routes in mesh networks; Easy implementation
Directional Routing Protocol [142]	Evolutionary, Targeted, Network-centric	Streaming	PHY layer information	Low latency and overhead; Fast route repair
Distributed Passive Routing Decisions [143]	Evolutionary, Targeted, Network-centric	Streaming	Node location	Higher energy efficiency; Lower signaling overhead

mathematical models based on simplified scenarios that, nonetheless, provide interesting and useful insights. These models are based on the concept of layering as optimization decomposition, which is a powerful way for analytically defining cross-layer optimization problems and, at the same time, designing feasible algorithms for their solution. It consists of modeling the overall communication network as a generalized network utility maximization problem, where each layer corresponds to a decomposed subproblem and intra-layer interfaces are quantified as functions of the optimization variables coordinating the subproblems. A survey of the different decomposition strategies applied to the wireless layering architecture can be found in [122].

A seminal work introducing a control theory approach to network optimization is pre-

sented in [147]. Here the authors focus on the problem of congestion control at the transport layer, which is addressed using control theory. Sources compute the optimum flow rate based on a feedback price accounting for the network load and by using an iterative algorithm, which is proven to converge to the optimum solution under certain assumptions. After that, algorithms for joint congestion control and transmission scheduling have been proposed to jointly optimize source rate, link scheduling, routing [80, 148–150] and transmission power [151, 152].

The mathematical tools widely used in these analytical approaches are optimization problem decomposition by Lagrange relaxation, sub gradient algorithms and Lyapunov stability [153, 154]. A comprehensive study about different ways of decomposing the optimization problem is presented in [155].

In general, joint congestion control and traffic scheduling can be formulated as a constrained maximization problem. Each unicast end-to-end flow is associated to a given utility function of the flow rate, which is assumed to be strictly concave in order to permit convex optimization techniques. The goal is to maximize the sum of the utility functions of the different flows, under the constraint of flow conservation, feasible rate region (physical link rates have to be feasible in the considered interference model), and link stability (the aggregate flow on a link cannot exceed the physical rate). Different utility functions correspond to different optimization goals, such as throughput, delay, energy, and so on. By considering a simple protocol interference model, according to which multiple transmissions by nodes within a given distance from the receiver result in destructive collisions, the scheduling problem for a single channel scenario becomes a weighted maximum independent set problem, which is in general NP-hard [156]. Clearly, a greedy centralized algorithm that simply selects at each step the link with the highest metric and discards all the interfering links can achieve a capacity region reduced by a factor of 1/K where K is the interference degree [156]. In [157], it is pointed out that such a greedy approach is optimal in graphs with particular structure (tree, clique). Algorithms based on a maximal independent set scheduler are known for single hop networks and are presented in [158, 159], but this approach cannot be extended to the multi-hop case. In this case a different scheduler has to be used, which exploits additional information on the traffic intensity or number of hops. Two different approaches to these problems are the so-called *backpressure* and *link-centric* solutions.

The *backpressure* approach makes use of control theory techniques to maximize the optimization function [160]. Given a set of input rates that lie inside the capacity region of the

system, this algorithm is able to guarantee stability, i.e., bounded queue lengths. The core of the scheduler is based on the maximization of a metric which depends on the rate allocated to each link, multiplied by the difference between the queue lengths at the link receiver side and the queue length at the transmitter side, thus the name backpressure. In [160], the authors consider a simple physical layer model, where the evolution of the channel states can be modeled as a multidimensional, finite Markov chain. The time is slotted and within each slot the network status (topology and channel) is constant. Under this assumption, the proposed algorithm is able to maximize the utility function to a value that is close to the optimal one, while stabilizing the network. The algorithm is provided with tunable parameters that can be used for trading optimality with stationary queue lengths, i.e., trading utility for delay.

A novel approach for the scheduling problem is also proposed in [115], where a randomized algorithm is used. The problem of maximizing the backpressure function is converted into the problem of comparing the backpressure value obtained in subsequent random schedules. At each time slot, the backpressure achieved by a new random maximal matching is compared with the one achieved by the previous schedule. The best schedule is then applied.

A different approach to the optimization decomposition refers to the *link centric* case. This and similar formulations have been used for early studies in congestion control [147, 153]. In the link centric approach, the maximization problem is solved under the assumption that the input rate of each flow is applied to all the links traversed by the flow simultaneously. Each flow is associated with a predetermined path and it is assumed that the rate computed by the congestion controller is applied simultaneously to all the links. In this way, the congestion controller reacts to the sum of the queues along the path, thus leading to the optimal throughput.

The main drawback of this approach is the difficulty of translating the theoretical optimization techniques into practical schemes, suitable for implementation in actual communication devices. In fact, they generally assume that the scheduling algorithm is provided with perfect knowledge of the feasible rate region and is able to make optimum choices in the maximization of the scheduling metric. The use of an imperfect scheduler in the joint scheduling and congestion control may, in general, lead to poor performance [156]. In case an imperfect scheduler is used, the backpressure algorithm presented in [160] is proved to be able to guarantee stability within a capacity region scaled by a factor that depends on

the imperfect scheduler. This opens the way to the implementation of reduced complexity schedulers.

A.6 Cognitive Techniques

CLO's increasing popularity leads to its inclusion and application in various wireless networking research areas. Relevant examples comprise CLO system design for energy efficient wireless networking (green communications paradigm), CLO enabled reconfigurable interoperability of heterogeneous wireless systems (seamless and transparent vertical handover), etc. One of the most promising applications of CLO techniques nowadays is in the emerging area of cognitive networking.

Cognitive networks share information that is not available in the strict layered architecture. As a result they inherently perform CLO in a certain manner [149, 161]. However, the cognitive paradigm extends much farther than the classical CLO scope by including the ability to "learn" from experience. The cognitive abilities of wireless nodes refer to their awareness of flexibility, services and opportunities in the communicating environment aiming to choose optimal system operation under various conditions. Furthermore, possible cooperation among nodes may yield altruistic (and sometimes non-altruistic) interactions between network entities in order to minimize the network overhead and optimize various system parameters. The integration of the CLO concept with cooperation and cognition has been discussed in [162], where the authors propose a novel design paradigm that permits an efficient and intelligent adaptation of wireless systems to the ever-changing wireless communication environments.

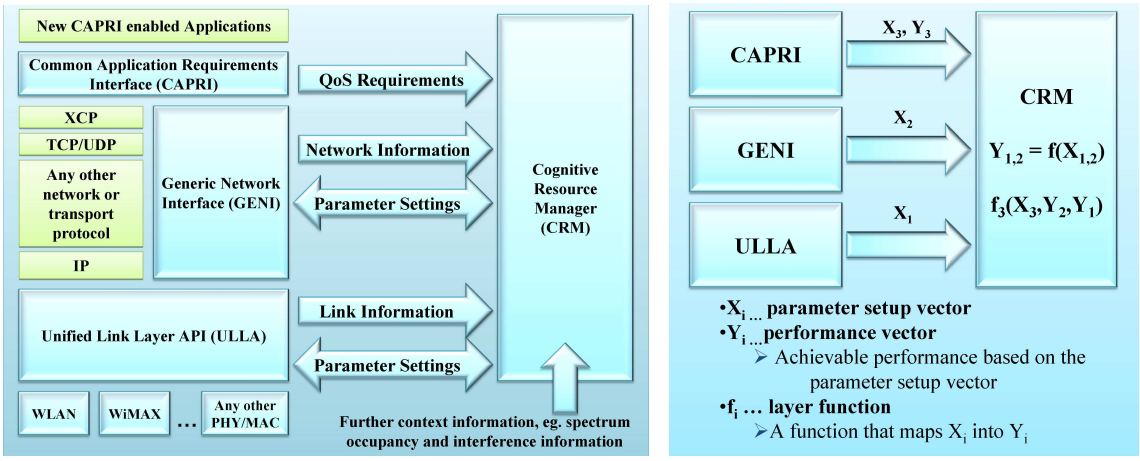
Today, this adaptation is mainly realized by means of autonomic computing principles [163], according to which a system can automatically perform tasks, which previously required intensive manual reconfiguration, by means of an "observe-plan-act" feedback loop enhanced by learning capabilities [163]. This represents a complex optimization problem whose solution involves different technology-specific information and parameters. In order to achieve optimal end-to-end performance, a cognitive network must strike a balance between multiple goals and, to this end, it has to perform multi-objective optimizations using several cross-layer optimization techniques simultaneously.

Recent publications and research projects recommend the usage of Artificial Intelligence (AI) techniques for Cross-layer Optimization in cognitive networks [164, 165]. In this con-

text there are several fields of AI that are intensively investigated, though most of them have been focused on the solution of specific problems. An attempt to overcome this discrepancy and use AI techniques along with machine learning in order to optimize the protocol stack of a wireless node as a whole is given in the ARAGORN architecture [166], a subsumption-based, CLO-enabled cognitive and collaborative wireless system. A key part of the architecture lies in its ability to perform CLO that is evolutionary, targeted, user-centric and centralized. In order to do so, the ARAGORN project introduces a Cognitive Resource Manager (CRM) which interacts with the protocol stack by using special interfaces. These interfaces are named ULLA (for interaction with the link/physical layer), GEneric Network Interface (GENI) (for interaction with the transport/network layer) and Common Applications Requirements Interface (CAPRI) (for imposing application layer requirements), Figure A.5(a). They constantly feed the CRM with relevant information from various parts of the protocol stack. For example, Universal Link Layer API (ULLA) is responsible for giving up-to-date information on available bandwidth, delay, link availability, etc., GENI may expose information such as congestion window size, routing metric, addressing scheme, mobility management protocol, etc., and finally CAPRI imposes the QoS requirements. The ARAGORN project aims to identify appropriate layer functions in order to map the parameter setup vectors, calculate appropriate layer descriptions (which are subsets of all possible achievable performance vectors) and define an appropriate objective function that conveys the QoS requirements (application targeted optimization). The CRM always has a complete set of all feasible parameter setup vectors at the appropriate layers. It is therefore able to perform CLO using multi-dimensional optimization algorithms, Figure A.5(b), with the aid of AI and machine learning algorithms. All these aspects contribute to an efficient use of both node-local and shared resources in a collaborative wireless system.

Another popular CLO enabling technique for cognitive networks are neural networks. They have been considered for channel assignment [167] and routing [168], and lately for signal classification [169]. A recent work [170] proposes the use of Multilayer Feed-Forward Neural Networks (MFNNs) to model the performance characteristics of Cognitive Radio emphasizing the modularity of this approach that can be applied in several layers, thus providing a multi-objective optimization tool.

In addition, genetic algorithms seem to be suited to handle large sets of heterogeneous variables, possibly carrying out cross-layer optimization over heterogeneous wireless interfaces, protocols and applications. In [171] the authors propose the use of a Hidden Markov



(a) ARAGORN system architecture (b) ARAGORN's CLO approach

Figure A.5. *ARAGORN's CLO vision.*

Model (HMM) trained by a genetic algorithm to model the channel response. It has to be emphasized that genetic algorithms usually need a long learning process in order to give acceptable solutions, which may or may not be acceptable according to the requirements of the running applications.

Also, few works [172, 173] exploit Fuzzy Logic to enable CLO. These works propose the use of a fuzzy logic controller to implement a technology-specific cross-layer solution, but make no effort to generalize the proposed approach to different CLO problems. On the contrary, in [125] the authors actually use Fuzzy Logic for the representation of cross-layer information and for the implementation of optimization strategies, making a special effort to generalize this approach.

This section explored the benefits of marrying CLO with cognition and cooperation. Even though not all efforts in this direction could be reported, the discussed contributions serve as evidence that CLO system design is constantly getting attention from both academia and industry, and finds its place in the latest wireless networking research topics.

A.7 On the Potential Pitfalls of CLO

In the previous sections we gave a detailed overview of cross-layer design and its architectural ramifications. Thanks to the capability of cross-layer design to introduce interactions between various layers of the protocol stack, researchers have achieved significant performance improvements in current wireless networks. However, attention has to be paid

when using this kind of design as it may run at cross purposes with long term architectural principles, potentially leading to various negative consequences [117, 147, 174].

The ever-present tension between performance and architecture in network design may be seriously disrupted by the use of cross layering. Performance targets short-term gains, while protocol and system architectures address stability, interoperability and long term evolution, forcing designers to evaluate an adequate trade-off between them. As an example, a particular cross-layer solution may yield immediate benefit to the wireless network performance, but at the expense of the long-term robustness of the system architecture that may yield network instability and other collateral effects.

That is, the cross-layer design principle brings along the risk of a "'spaghetti"' effect. For example, once the entire network stack is considered, cross-layer design may lead to cycles in the logical architecture due to hidden interactions that are not easily recognized by the designer. As a result, cross-layer optimization should be taken as a holistic rather than as a fragmented concept.

In order to illustrate more concretely the possibility of unintended interactions, in the following we present two examples of CLO-based schemes that, in certain conditions, suffer from the problems discussed above.

A first example of bad cross-layer design, discussed in [147], is based on an 802.11 scheme using rate adaptation. The idea behind rate-adaptive MAC protocols is to send data at higher rates when the channel quality is good. It is shown, though, that such schemes can have undesirable consequences for the higher layers. In particular, when such a scheme is combined with minimum-hop routing, say a protocol like Destination Sequenced Distance Vector (DSDV), they can lead to poorer performance than the original system. Essentially, the reason for this adverse behavior is that when considering MAC layer design, no one took into account the collateral effects that cross-layer decisions were going to have on the routing layer. In another example, cross-layer design is used in order to enhance TCP performance based on the capability of adjusting the number of neighbors of each node. The authors of [147] show that, in certain scenarios, the network oscillates between connectivity and disconnection, affecting TCP performance adversely and, more importantly, leading to network instability.

Therefore, cross-layer solutions are much more difficult to standardize and adapt to particular applications. This causes an increase in the system design and maintenance costs, which is the main reason why cross-layer based solutions often end up being applied in

stand-alone networks or in isolated parts of more complex systems.

A.8 Future Research Directions

The traditional design principle of network protocol stacks (i.e., the layering concept) comprises protocol layers usually designed to operate in worst case conditions rather than being able to adapt. The latter is often a must in the wireless domain, as wireless networks exhibit dynamic behaviors that make conventional protocol stack design inflexible in many ways. Therefore, the rapid evolution of wireless networking technologies today requires new concepts and methods which are potentially different from the classical solutions applied so far.

A prime example of this development is CLO system design, where the protocol stack exhibits various, protocol relevant, information exchanges by coupling various protocols' mechanisms. A natural consequence of this attempt is to improve the system performance at the local (i.e., node-limited) or global (i.e., network) level. CLO system design has several goals. It may increase the network throughput by reducing the unnecessary overhead in the network. It may also reduce latency (i.e., delay) which is a crucial parameter in mobile environments. Furthermore, the CLO techniques provide reduction of the network disconnection time and reduce power consumption. From a users' point of view, they also improve the application performance and increase the user satisfaction. Overall, CLO system design offers better utilization of the available resources in wireless networks.

The development of wireless networking technologies towards the incorporation of CLO system design faces many challenges. Several key features must be carefully addressed before deploying CLO solutions in practice. Backward compatible solutions require an evolutionary system design (preferred by industry), however they that may not achieve the full performance gains promised by the thorough application of the CLO concept. On the other hand, the revolutionary concepts (mostly limited to academic approaches) often exhibit high performance improvements, but are left in isolated environments. Therefore, the actual application domain of the wireless network solution and the various performance gain trade-offs must be scrutinized in order to find the best possible match. Furthermore, the emerging topic of cognitive networking inherently introduces a memoryful CLO system design. Finally, the potential pitfalls of CLO system design show that the embracement of

this approach actually requires to radically change the networking philosophy and to abandon the strict layering vision of communication network architecture.

This chapter has given a comprehensive overview on the various aspects of CLO system design. It provided a novel CLO analysis framework that makes it possible to go beyond the heuristic analysis and to establish a clear technical basis for CLO system design. The framework is based on definitions and provides a classification of the majority of the CLO approaches found in the literature today. Some of the most prominent CLO architectures are briefly explained and fit in the context of the newly defined framework. In addition, the chapter introduced a comparison between CLO design and cognitive networking, a hot research topic in the years to come.

The CLO concept is an intricate way to analyze and couple protocol functionalities. It frees the researchers from the constraints of the classical protocol layers and unleashes a novel optimization space where many parameters can be adapted and improved. It is tightly related to cognitive networking, an additional argument in favor of the approach. As the popularity of wireless networking increasingly emerges, CLO will have an enormous impact and may prove vital in the development of true ubiquitous communications.

List of Publications

J1 A. Asterjadhi and M. Zorzi, "JENNA: A Jamming Evasive Network-coding Neighbor-discovery Algorithm for Cognitive Radio Networks," in IEEE Wireless Commun. Mag. [Special Issue in Dynamic Spectrum Management], vol. 17, no. 4, pp. 24-32, 2010.

J2 N. Baldo and A. Asterjadhi and M. Zorzi, "Dynamic Spectrum Access Using a Network Coded Cognitive Control Channel," in IEEE Trans. on Wireless Commun., vol. 9, no. 8, pp. 2575-2587, Aug., 2010.

J3 A. Asterjadhi and E. Fasolo and J. Widmer and M. Rossi and M. Zorzi, "Toward Network Coding-based Protocols For Data Broadcasting in Wireless ad Hoc Networks," in IEEE Trans. on Wireless Commun., vol.9, no.2, pp. 662-673, Feb. 2010.

J4 A. Asterjadhi and N. Baldo and M. Zorzi, "A Distributed Network Coded Control Channel for Multi-Hop Cognitive Radio Networks," in IEEE Network [Special Issue on Multi-Hop Cognitive Radio Networks], vol. 23, no. 4, pp. 26-32, Jul.-Aug., 2009.

J5 A. Asterjadhi and V. Atanasovski and L. Gavrilovska and A. Zanella and M. Zorzi, "Cross-Layer Optimization for Wireless Networks: an Overview," IEEE Communications Surveys and Tutorials, (submitted).

C1 A. Asterjadhi and M. Zorzi, "JENNA: A Jamming Evasive Network-coding Neighbor-discovery Algorithm for Cognitive Radio Networks, " in IEEE International Conference on Communications Workshops (ICC), Cape Town (SA), pp. 1-6, May, 2010.

C2 N. Baldo and A. Asterjadhi and L. Giupponi and M. Zorzi, "A Scalable Dynamic Spectrum Access Solution for Large Wireless Networks, " in 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC), Modena (IT), pp. 430-435, Apr., 2010.

C3 A. Asterjadhi and N. Baldo and M. Zorzi, "A Cluster Formation Protocol for Cognitive Radio ad Hoc Networks, " in European Wireless Conference (EW), Lucca (IT), pp. 955-961, Apr., 2010.

C4 N. Baldo and A. Asterjadhi and M. Zorzi, "Cooperative Detection and Spectrum Reuse using a Network Coded Cognitive Control Channel" in 6th IEEE SECON Workshop on Networking Technologies for Software Defined Radios (SDR) Networks, Rome (IT), pp. 1-7, Jun., 2009.

C5 N. Baldo and A. Asterjadhi and M. Zorzi, ”Distributed Dynamic Spectrum Access using a Virtual Network Coded Control Channel” in International Wireless Communications and Mobile Computing (IWCMC), Leipzig (DE), pp. 1000-1005, Jun., 2009.

C6 A. Asterjadhi and R. Kumar and T. La Porta and M. Zorzi, ”Broadcasting in Multi Channel Wireless Networks in the Presence of Adversaries” in 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), (submitted), Salt Lake City, Utah (USA), Jun., 2011.

Bibliography

[1] Ofcom, "Digital Dividend: Cognitive Access Consultation on Licence-exempting Cognitive Devices using Interleaved Spectrum," Feb. 2009. [Online]. Available: http://www.ofcom.org.uk/consult/condocs/cognitive/cognitive.pdf

[2] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas In Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[3] J. Mitola, "Cognitive Radio: an Integrated Agent Architecture for Software Defined Radio," Ph.D. dissertation, Royal Institute of Technology (KTH), 2000.

[4] Federal Communications Commission, "Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies," 2005.

[5] "Decision No 676/2002/EC of the European Parliament and of the Council of 7 March 2002 on a Regulatory Framework for Radio Spectrum Policy in the European Community (Radio Spectrum Decision)."

[6] R. Ahlswede, C. Ning, S.-Y. R. Li, and R. Yeung, "Network Information Flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *41st Allerton Conference on Communication Control and Computing*, Allerton, IA, US, Oct. 2003.

[8] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[9] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The Benefits of Coding Over Routing in a Randomized Setting," in *International Symposium on Information Theory (ISIT)*, 2003.

[10] Y. Wu, P. Chou, and K. Jain, "A Comparison of Network Coding and Tree Packing," in *International Symposium on Information Theory (ISIT)*, 2004.

[11] D. Tuninetti and C. Fragouli, "Processing Along the Way: Forwarding vs. Coding," in *ISITA*, Parma, Italy, oct 2004.

[12] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[13] J. Yuan, Z. Li, W. Yu, and B. Li, "A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, Nov. 2006.

[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in The Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, Pisa, Italy, Sep. 2006.

[15] S. Omiwade, R. Zheng, and C. Hua, "Practical Localized Network Coding in Wireless Mesh Networks," in *IEEE SECON*, San Francisco, USA, jun 2008, pp. 332–340.

[16] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 136–150, 2005.

[17] H. Yomo and P. Popovski, "Opportunistic Scheduling for Wireless Network Coding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 2766–2770, jun 2009.

[18] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "Efficient Broadcasting Using Network Coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, 2008.

[19] C. Gkantsidis and P. R. Rodriguez, "Network Coding for Large Scale Content Distribution," in *IEEE INFOCOM*, Miami, FL, USA, mar 2005, pp. 2235–2245.

[20] R. Koetter and M. Médard, "An Algebraic Approach to Network Coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, oct 2003.

[21] C. Fragouli and E. Soljanin, "Information Flow Decomposition for Network Coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 829–848, mar 2006.

[22] C. Chekuri, C. Fragouli, and E. Soljanin, "On Average Throughput and Alphabet Size in Network Coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2410–2424, jun 2006.

[23] H. Dubois-Ferriére, D. Estrin, and M. Vetterli, "Packet Combining in Sensor Networks," in *ACM SenSys*, San Diego, CA, USA, nov 2005, pp. 102–115.

[24] Y. E. Sagduyu and A. Ephremides, "Crosslayer Design for Distributed MAC and Network Coding in Wireless Ad Hoc Networks," in *IEEE ISIT*, Adelaide, Australia, sep 2005, pp. 1863–1867.

[25] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, mar 2002.

[26] P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulie, "Epidemic Information Dissemination in Distributed Systems," *Computer*, vol. 37, no. 5, pp. 60–67, may 2004.

[27] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "Toward Network Coding-Based Protocols for Data Broadcasting in Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 662–673, Feb. 2010.

[28] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "On MAC Scheduling and Packet Combination Strategies for Random Network Coding," in *IEEE ICC*, Glasgow, UK, Jun. 2007.

[29] ——, "A Proactive Network Coding Strategy for Pervasive Wireless Networking," in *IEEE GLOBECOM*, Washington DC, US, Nov. 2007.

[30] "ns2 Simulation Code for Wireless Network Coding." [Online]. Available: http://telecom.dei.unipd.it/simulation

[31] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "ns2-MIRACLE: a Modular Framework for Multi-Technology and Cross-Layer Support in Network Simulator 2," in *ACM NSTools*, Nantes, France, Oct. 2007.

[32] J. Widmer, C. Fragouli, and J.-Y. L. Boudec, "Low-Complexity Energy-Efficient Broadcasting in Wireless Ad Hoc Networks Using Network Coding," in *NetCod*, Riva del Garda, Italy, Apr. 2005.

[33] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

[34] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, "Unsaturated Throughput Analysis of IEEE 802.11 in the Presence of Non Ideal Transmission Channel and Capture Effects," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1276–1286, 2008.

[35] A. Gkelias, M. Dohler, and H. Aghvami, "Throughput Analysis for Wireless Multi-hop CSMA," in *IEEE PIMRC*, Barcelona, Spain, Sep. 2004.

[36] A. Asterjadhi, R. Kumar, T. L. Porta, and M. Zorzi, "Broadcasting in Multi Channel Wireless Networks in the Presence of Adversaries," in *the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), submitted*, Jun. 2011.

[37] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: a Survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[38] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice-Hall, 2001.

[39] R. A. Poisel, *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006.

[40] L. Baird, W. Bahn, M. Collins, M. Carlisle, and S. Butler, "Keyless Jam Resistance," in *IEEE Inf. Assur. and Sec. Workshop*, Jun. 2007, pp. 143–150.

[41] K. Balachandran and J. H. Kang, "Neighbor Discovery with Dynamic Spectrum Access In Adhoc Networks," in *IEEE 63rd Vehicular Technology Conference (VTC)*, May 2006, pp. 512–517.

[42] M. Strasser, S. Capkun, C. Popper, and M. Cagalj, "Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping," in *IEEE Symposium on Security and Privacy*, 2008, pp. 64–78.

[43] M. Strasser, C. Pöpper, and S. Čapkun, "Efficient Uncoordinated FHSS Anti-jamming Communication," in *ACM Mobihoc*, 2009.

[44] D. Slater, P. Tague, R. Poovendran, and B. J. Matt, "A Coding-Theoretic Approach for Efficient Message Verification over Insecure Channels," in *ACM WiSec*, 2009, pp. 151–160.

[45] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication," in *IEEE InfoCom*, 2010.

[46] C. Popper, M. Strasser, and S. Capkun, "Anti-Jamming Broadcast Communication using Uncoordinated Spread Spectrum Techniques," *IEEE J Sel. Area Comm.*, vol. 28, no. 5, pp. 703–715, 2010.

[47] J. Chiang and Y.-C. Hu, "Dynamic Jamming Mitigation for Wireless Broadcast Networks," in *IEEE InfoCom*, 2008.

[48] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor Discovery in Wireless Networks and the Coupon Collectors Problem," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Sept. 2009, pp. 181–192.

[49] S. Deb, M. Medard, and C. Choute, "Algebraic Gossip: a Network Coding Approach to Optimal Multiple Rumor Mongering," *IEEE Trans. on Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, 2006.

[50] J. Kobza, S. Jacobson, and D. Vaughan, "A Survey of the Coupon Collector's Problem with Random Sample Sizes," *Methodology and Computing in Applied Probability*, vol. 9, pp. 573–584, 2007.

[51] A. Asterjadhi and M. Zorzi, "JENNA: a Jamming Evasive Network-coding Neighbor-discovery Algorithm for Cognitive Radio Networks [Dynamic Spectrum Management]," *Wireless Communications, IEEE*, vol. 17, no. 4, pp. 24–32, Aug. 2010.

[52] ——, "JENNA: a Jamming Evasive Network-coding Neighbor-discovery Algorithm for Cognitive Radio Networks," in *IEEE ICC Workshop on Cooperative and Cognitive Mobile Networks (CoCoNet3)*, May. 2010.

[53] T. C. Clancy and N. Goergen, "Security in Cognitive Radio Networks: Threats and Mitigation," in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, Singapore, May 2008, pp. 1–8.

[54] R. Zheng, J. Hou, and L. Sha, "Asynchronous Wakeup for ad Hoc Networks," in *ACM MobiHoc*, New York, NY, USA, 2003, pp. 35–45.

[55] S. Vasudevan, J. Kurose, and D. Towsley, "On Neighbor Discovery in Wireless Networks with Directional Antennas," in *IEEE INFOCOM*, vol. 4, Mar. 2005, pp. 2502–2512.

[56] J. Luo and D. Guo, "Neighbor Discovery in Wireless ad Hoc Networks Based on Group Testing," in *46th Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2008, pp. 791–797.

[57] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Towards Provable Secure Neighbor Discovery in Wireless Networks," in *ACM FMSE*, Oct. 2008, pp. 31–41.

[58] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, New York, NY, USA, 2005, pp. 46–57.

[59] N. Mittal, S. Krishnamurthy, R. Chandrasekaran, S. Venkatesan, and Y. Zeng, "On Neighbor Discovery in Cognitive Radio Networks," *Journal of Parallel and Distributed Computing*, vol. 69, pp. 623–637, 2009.

[60] S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mittal, S. Venkatesan, and R. Prakash, "Time-Efficient Distributed Layer-2 Auto-configuration for Cognitive Radio Networks," *Computer Networks*, vol. 52, no. 4, pp. 831–849, 2008.

[61] C. J. L. Arachchige, S. Venkatesan, and N. Mittal, "An Asynchronous Neighbor Discovery Algorithm for Cognitive Radio Networks," in *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Oct. 2008, pp. 1–5.

[62] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J.-P. Hubaux, "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad hoc Networking," *IEEE Communications Magazine*, vol. 46, no. 2, Feb. 2008.

[63] D. Liu, "Protecting Neighbor Discovery Against Node Compromises in Sensor Networks," in *IEEE ICDCS*, Jun. 2009, pp. 579–588.

[64] R. Leidenfrost and W. Elmenreich, "Establishing Wireless Time-Triggered Communication using a Firefly Clock Synchronization Approach," in *International Workshop on Intelligent Solutions in Embedded Systems*, Jul. 2008, pp. 1–18.

[65] Y. Zhang, G. Xu, and X. Geng, "Security Threats in Cognitive Radio Networks," in *IEEE International Conference on High Performance Computing and Communications*, Washington, DC, USA, 2008, pp. 1036–1041.

[66] R. Chen, J.-M. Park, and J. Reed, "Defense Against Primary User Emulation Attacks in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, Jan. 2008.

[67] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation Issues in Spectrum Sensing for Cognitive Radios," *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 772–776, Nov. 2004.

[68] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Mitigating Primary User Emulation Attacks in Dynamic Spectrum Access Networks Using Hypothesis Testing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 74–85, Apr. 2009.

[69] N. Baldo, A. Asterjadhi, and M. Zorzi, "Dynamic Spectrum Access Using a Network Coded Cognitive Control Channel," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2575–2587, Aug. 2010.

[70] ——, "Cooperative Detection and Spectrum Reuse using a Network Coded Cognitive Control Channel," in *Proc. of IEEE SECON Workshops (SDR)*, Jun. 2009.

[71] ——, "Multi-Channel Medium Access using a Virtual Network Coded Control Channel," in *ACM IWCMC*, 2009, pp. 1000–1005.

[72] S. L. Wu, C. Y. Lin, Y. C. Tseng, and J. P. Sheu, "A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks," *The Computer Journal*, vol. 45(1), pp. 101–110, 2002.

[73] H. Wing-Chung, K. L. Eddie Law, and A. Leon-Garcia, "A Dynamic Multi-Channel MAC for Ad-Hoc LAN," in *Proc. 21st Biennial Symposium on Communications*, 2002, pp. 31–35.

[74] A. Motamedi and A. Bahai, "MAC Protocol Design for Spectrum-agile Wireless Networks: Stochastic Control Approach," in *IEEE DySPAN*, Apr. 2007.

[75] C. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-channel Wireless Networks," in *IEEE DySPAN*, Apr. 2007.

[76] A. Tzamaloikas and J. J. Garcia-Luna-Aceves, "Channel Hopping Multiple Access," in *IEEE ICC*, New Orleans, LA, US, Jun. 2000, pp. 415–419.

[77] W. Hoi-Sheung So, J. Walrand, and J. Mo, "McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol," in *IEEE WCNC*, Mar. 2007.

[78] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks," in *ACM MobiCom*, Sep. 2004.

[79] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 589–600, Apr. 2007.

[80] S. M. Mishra, A. Sahai, and R. Brodersen, "Cooperative Sensing Among Cognitive Radios," in *IEEE ICC*, Jun. 2006.

[81] J. Unnikrishnan and V. V. Veeravalli, "Cooperative Sensing for Primary Detection in Cognitive Radio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 18–27, Feb. 2008.

[82] K. A. Woyach, A. Sahai, G. Atia, and V. Saligrama, "Crime and Punishment for Cognitive Radios," in *Forty-Sixth Annual Allerton Conference*, Sep. 2008.

[83] F. F. Digham, M. S. Alouini, and M. K. Simon, "On the Energy Detection of Unknown Signals Over Fading Channels," in *IEEE ICC*, vol. 5, 2003.

[84] J. Elson and K. Römer, "Wireless Sensor Networks: A New Regime for Time Synchronization," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 149–154, 2003.

[85] "IEEE 1609.4: Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-channel Operation," 2006.

[86] H. Urkowitz, "Energy Detection of Unknown Deterministic Signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, 1967.

[87] M. M. Rashid, J. Hossain, E. Hossain, and V. K. Bhargava, "Opportunistic Spectrum Access in Cognitive Radio Networks: A Queueing Analytic Model and Admission Controller Design," in *IEEE GlobeCom*, Nov. 2007.

[88] S. Misra, S. Geirhofer, and L. Tong, "Optimal Adaptive Transmission for a Cognitive Radio with Sensing," in *IEEE ICASSP*, Apr. 2008.

[89] J. Xie, I. Howitt, and A. Raja, "Cognitive Radio Resource Management Using Multi-Agent Systems," in *IEEE CCNC*, Jan. 2007.

[90] A. Asterjadhi, N. Baldo, and M. Zorzi, "A Distributed Network Coded Control Channel for Multi-Hop Cognitive Radio Networks," *IEEE Network*, vol. 23, no. 4, pp. 26–32, Aug. 2009.

[91] ——, "A Cluster Formation Protocol for Cognitive Radio Ad Hoc Networks," in *European Wireless Conference (EW)*, Apr. 2010, pp. 955–961.

[92] N. Baldo, A. Asterjadhi, L. Giupponi, and M. Zorzi, "A Scalable Dynamic Spectrum Access Solution for Large Wireless Networks," in *5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, May 2010, pp. 430–435.

[93] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," in *IEEE WoWMoM*, Jun. 2005, pp. 78–85.

[94] J. Mo, H.-S. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 50–65, 2008.

[95] Q. Wang and H. Zheng, "Route and Spectrum Selection in Dynamic Spectrum Networks," in *IEEE CNCC*, Jan. 2006, pp. 625–629.

[96] S. Merlin, N. Vaidya, and M. Zorzi, "Resource Allocation in Multi-Radio Multi-Channel Multi-Hop Wireless Networks," in *IEEE INFOCOM*, 2008, pp. 610–618.

[97] L. Cao and H. Zheng, "Distributed Spectrum Allocation via Local Bargaining," *Proc. of IEEE SECON*, Sept. 2005.

[98] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "CogMesh: A Cluster-based Cognitive Radio Network," in *Proc. of IEEE DySPAN*, Apr. 2007.

[99] L. Lazos, S. Liu, and M. Krunz, "Spectrum Opportunity-Based Control Channel Assignment in Cognitive Radio Networks," in *Proc. of IEEE SECON*, Jun. 2009.

[100] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[101] S. Basagni, "Distributed clustering for ad hoc networks," in *Proc. of I-SPAN*, Washington, DC, USA, June 1999.

[102] L. Narayanan, "Channel Assignment and Graph Multicoloring," *Handbook of wireless networks and mobile computing*, pp. 71–94, 2002.

[103] I. Caragiannis, C. Kaklamanis, and E. Papaioannou, "Efficient On-line Frequency Allocation and Call Control in Cellular Networks," *Theory of Computing Systems*, vol. 35, no. 5, pp. 521–543, Mar. 2008.

[104] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosén, "On-line Randomized Call Control Revisited," *SIAM Journal on Computing*, vol. 31, no. 1, pp. 86–112, Jan. 2002.

[105] G. Pantziou, G. Pentaris, and P. Spirakis, "Competitive Call Control in Mobile Networks," *Theory of Computing Systems*, vol. 35, no. 6, pp. 625–639, Mar. 2008.

[106] C. Peng, H. Zheng, and B. Zhao, "Utilization and Fairness in Spectrum Assignment for Opportunistic Spectrum Access," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 555–576, Aug. 2006.

[107] P. Kyasanur and N. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 515–527, Apr. 2009.

[108] H. M. Taylor and S. Karlin, *An introduction to Stochastic Modeling*. Academic Press.

[109] C. Cordeiro, K. Challapali, and D. Birru, "IEEE 802.22: the First Worldwide Wireless Standard Based on Cognitive Radios," *IEEE DySPAN*, pp. 328–337, 2005.

[110] D. Cabric, S. Mishra, D. Willkomm, R. Brodersen, and A. Wolisz, "CORVUS: A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum," *IST Mobile and Wireless Communications Summit*, Jun. 2005.

[111] M. Sherman, A. Mody, R. Martinez, C. Rodriguez, and R. Reddy, "IEEE Standards Supporting Cognitive Radio and Networks, Dynamic Spectrum Access, and Coexistence," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 72–79, 2008.

[112] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, "Network Coding Security: Attacks and Countermeasures," *Computing Research Repository*, 2008.

[113] L. Giupponi and C. Ibars, "Misbehaviour Detection in Cognitive and Cooperative Networks," in *Proc. of IEEE GLOBECOM*, Dec. 2009.

[114] A. Asterjadhi, V. Atanasovski, L. Gavrilovska, A. Zanella, and M. Zorzi, "Cross-Layer Optimization for Wireless Networks: an Overview," *IEEE Communications Surveys and Tutorials, submitted*, 2011.

[115] A. Ephremides and B. Hajek, "Information Theory and Communication Networks: an Unconsummated Union," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2416–2434, 1998.

[116] D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," in *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, Sep. 1990.

[117] F. Aune, "Cross-Layer Design Tutorial," *Norwegian University of Science and Technology, Published under Creative Commons License*, Nov. 2004.

[118] L. Gavrilovska and R. Prasad, *Ad Hoc Networking Towards Seamless Communications*. Springer, 2006.

[119] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-Driven Cross-Layer Optimization for Video Streaming over Wireless Networks," *IEEE Communications Magazine*, pp. 122–130, Jan. 2006.

[120] J. Brehmer and W. Utschick, "Modular Cross-Layer Optimization Based on Layer Descriptions," in *WPMC*, Aalborg, Denmark, 2005.

[121] F. Fu and M. Van Der Schaar, "A New Theoretic Foundation for Cross-Layer Optimization," *UCLA Technical Report*, Dec. 2007.

[122] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," *Proceedings of IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[123] "Cross-layer Optimization," *IST Integrated Project 26950: SATSIX, Deliverable 2000-4*, Dec. 2007. [Online]. Available: http://www.ist-satsix.org/DOC/D2000-4_v6.0.pdf

[124] Q. Wang and M. A. Abu, "Cross-Layer Signaling for Next-Generation Wireless Systems," *IEEE WCNC*, vol. 2, pp. 1084–1089, Mar. 2003.

[125] N. Baldo and M. Zorzi, "Fuzzy Logic for Cross-layer Optimization in Cognitive Radio Networks," *IEEE Communications Magazine*, Apr. 2008.

[126] L. U. Choi, W. Kellerer, and E. Steinbach, "Cross Layer Optimization for Wireless Multi-user Video Streaming," in *International Conference on Image Processing*, vol. 3, Oct. 2004, pp. 2047–2050.

[127] D. Triantafyllopoulou, N. Passas, A. K. Salkintzis, and A. Kaloxylos, "A Heuristic Cross-layer Mechanism for Real-time Traffic over IEEE 802.16 Networks," *Wireless Communications and Mobile Computing*, vol. 17, no. 5, pp. 347–361, Aug. 2007.

[128] N. Baldo, F. Maguolo, S. Merlin, A. Zanella, M. Zorzi, D. Melpignano, and D. Siorpaes, "APOS: Adaptive Parameters Optimization Scheme for Voice over IEEE 802.11g," in *IEEE ICC*, May 2008, pp. 2466–2472.

[129] C. Lochert, B. Scheuermann, and M. Mauve, "A Survey on Congestion Control for Mobile Ad-Hoc Networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, Jun. 2007.

[130] H. Zhai, X. Chen, and Y. Fang, "Improving Transport Layer Performance in Multihop Ad-hoc Networks by Exploiting MAC Layer Information," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1692–1701, May 2007.

[131] E. Hamadani and V. Rakocevic, "TCP Contention Control: A Crosslayer Approach to Improve TCP Performance in Multihop Ad hoc Networks," *Lecture Notes in Computer Science 4517, Springer*, vol. 4517/2007, pp. 1–16, Jun. 2007.

[132] L. Chen, S. H. Low, and J. C. Doyle, "Joint Congestion Control and Media Access Control Design for Ad-hoc Wireless Networks," in *IEEE INFOCOM*, vol. 3, Mar. 2005, pp. 2212–2222.

[133] D. Kliazovich, "Cross-layer Performance Optimization in Wireless Local Area Networks," Ph.D. dissertation, University of Trento, Trento, Italy, 2006.

[134] B. Scheuermann, *Reading Between the Packets - Implicit Feedback in Wireless Multihop Networks*. VDM Verlag Dr. Muller, 2008.

[135] H. K. Rath, A. Sahoo, and A. Karandikar, "A Cross-layer Congestion Control Algorithm in Wireless Networks for TCP Reno-2," in *National Conference on Communication*, Jan. 2006.

[136] L. Qin and T. Kunz, "Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design," Carleton University, Tech. Rep., Aug. 2004.

[137] S. Zhao, Z. Wu, A. Acharya, and D. Raychaudhuri, "PARMA: A PHY/MAC Aware Routing Metric for Ad-Hoc Wireless Networks with Multi-Rate Radios," in *IEEE WoWMoM*, Jun. 2005, pp. 286–292.

[138] R. Krishnan and T. C. Chiueh, "Link Characteristics Aware Wireless Protocol Design," *RPE Report, Experimental Computer Systems Lab*, Sep. 2006.

[139] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, "Cross-Layer Routing in Wireless Mesh Networks," in *1st International Symposium on Wireless Communication Systems*, Sep. 2004, pp. 319–323.

[140] L. Iannone, K. Kabassanov, and S. Fdida, "Evaluation of Cross-Layer Rate-Aware Routing in a Wireless Mesh Network Test-Bed," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, pp. 42–42, Jan. 2007.

[141] V. P. Mhatre, H. Lundgren, and C. Diot, "MAC-aware routing in wireless mesh networks," in *Annual Conference on Wireless on Demand Network Systems and Services*, Jan. 2007, pp. 46–49.

[142] H. Gossain, T. Joshi, C. M. Cordeiro, and D. P. Agrawal, "DRP: An Efficient Directional Routing Protocol for Mobile Ad Hoc Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 12, Dec. 2006.

[143] P. Skraba, H. Aghajan, and A. Bahai, "Cross-Layer Optimization for High Density Sensor Networks: Distributed Passive Routing Decisions," in *Ad-Hoc Now*, Jul. 2004, pp. 266–279.

[144] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Transactions on Mobile Computing*, vol. 2(4), pp. 349–365, 2003.

[145] R. C. Shah, "Distributed algorithms to maximize the lifetime of wireless sensor networks," Ph.D. dissertation, University of California, Berkeley, 2005.

[146] C. Julien and M. Venkataraman, "Cross-Layer Discovery and Routing in Reconfigurable Wireless Networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Oct. 2006, pp. 119–128.

[147] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, no. 3, 1998.

[148] R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 8, pp. 1452–1463, 2006.

[149] R. Thomas et al, "Cognitive Networks: Adaptation and Learning to Achieve End-to-End Performance Objectives," *IEEE Communications Magazine*, 2006.

[150] C. Julien and M. Venkataraman, "Some Fundamental Limits on Cognitive Radio," in *42nd Allerton Conf. on CCC*, Oct. 2004.

[151] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," in *IEEE INFOCOM*, 2005.

[152] M. Chiang, "Balancing Transport and Physical Layer in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control," *IEEE Journal on Selected Areas in Communication*, pp. 104–116, Jan. 2005.

[153] R. Srikant, *The mathematics of internet congestion control*. Birkhauser, 2006.

[154] D. P. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 8, pp. 1439–1451, 2006.

[155] D. Palomar and M. Chiang, "Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications," in *IEEE INFOCOM*, Apr. 2006, pp. 1–13.

[156] G. Sharma and N. Shroff, "Maximum Weighted Matching with Interference Constraints," in *PERCOM*, 2006.

[157] E. Modiano, G. Zussman, and A. Brzezinski, "Enabling Distributed Throughput Maximization in Wireless Mesh Networks via Local Pooling," in *Technical report, MIT*, 2006.

[158] X. Wu, R. Srikant, and J. R. Perkins, "Queue Length Stability of Maximal Greedy Schedules in Wireless Networks," in *ITA Workshop*, 2006.

[159] P. Chaporkar, K. Kar, and S. Sarkar, "Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks," in *ITA Workshop*, 2006.

[160] R. Srikant and X. Wu, "Regulated Maximal Matching: a Distributed Scheduling Algorithm for Multi-hop Wireless Networks with Node Exclusive Spectrum Sharing," in *IEEE CDC*, 2005.

[161] B. S. Manoj, R. R. Rao, and M. Zorzi, "CogNet: A Cognitive Complete Knowledge Network System," *IEEE Wireless Communications Magazine*, vol. 46, no. 6, 2008.

[162] T. Arildsen and F. H. P. Fitzek, "The C-Cube Concept - Combining Cross-Layer Protocol Design, Cognitive-, and Cooperative Network Concepts," *Cognitive Wireless Networks (eds. F. H. P. Fitzek and M. D. Katz), Springer*, pp. 423–433, 2007.

[163] J. Strassner, "Autonomic Networking - Theory and Practice," in *9th IFIP/IEEE International Symposium on Network Management (IM'2005)*, Nice, France, May 2005.

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

[165] "The ARAGORN Project (Adaptive reconfigurable access and generic interfaces in radio networks)," Jan. 2008. [Online]. Available: http://www.ict-aragorn.eu

[166] "State-of-the-art," *ICT STREP 216856: ARAGORN, Deliverable D2.1*, Apr. 2008. [Online]. Available: http://www.ict-aragorn.eu

[167] C. Barnhart, J. Wieselthier, and A. Ephremides, "A Neural Network Approach to Solving the Link Activation Problem in Multihop Radio Networks," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1277–1283, 1995.

[168] J. Wieselthier, C. Barnhart, and A. Ephremides, "Neural Network Approach to Routing Without Interference in Multihop Radio Networks," *IEEE Transactions on Communications*, vol. 42, no. 1, pp. 166–177, 1994.

[169] E. Like, V. Chakravarthy, and Z. Wu, "Reliable Modulation Classification at Low SNR Using Spectral Correlation," in *IEEE Consumer Communication and Networking Conference*, 2007.

[170] N. Baldo and M. Zorzi, "Learning and Adaptation in Cognitive Radios using Neural Networks," in *IEEE CCNC*, 2008.

[171] C. Rieser, T. Rondeau, C. Bostian, and T. Gallagher, "Cognitive Radio Testbed: Further Details and Testing of a Distributed Genetic Algorithm Based Cognitive Engine for Programmable Radios," in *IEEE Military Communications Conference (MILCOM)*, vol. 3, 2003.

[172] X. Xia and Q. Liang, "Bottom-Up Cross-Layer Optimization for Mobile Ad Hoc Networks," in *IEEE Military Communications Conference (MILCOM)*, 2005, pp. 1–7.

[173] B. Otal and L. Alonso, "A Cross-Layer Energy-saving Mechanism for an Enhancement of 802.11 WLAN Systems," in *IEEE 59th Vehicular Technology Conference (VTC 2004-Spring)*, vol. 3, 2004.

[174] V. Kawadia and P. R. Kumar, "A Cautionary Perspective on Cross-Layer Design," *IEEE Wireless Communication Magazine*, vol. 12, pp. 3–11, 2005.